UNIVERSIDAD DE OVIEDO

ESCUELA DE INGENIERÍA INFORMÁTICA

TRABAJO FIN DE MÁSTER

"DESARROLLO DIRIGIDO POR MODELOS Y BASADO EN DSL´s DE VIDEOJUEGOS DE ACCIÓN EN 2D PARA DISPOSITIVOS MÓVILES"

DIRECTORA: B. Cristina Pelayo García-Bustelo

CO-DIRECTOR: Oscar Sanjuán Martínez

AUTORA: Natalia García Menéndez

Vº Bº del Director del Proyecto

Agradecimientos

Para la realización de este trabajo han sido muchas las personas que me han ayudado y apoyado. No puedo citarlas aquí a todas, pero si me gustaría resaltar algunas de ellas.

En primer lugar, a B. Cristina Pelayo García-Bustelo y Oscar Sanjuán Martínez, mis directores de proyecto y quienes me dieron la oportunidad de poder desarrollarlo dentro de su grupo de investigación en la Universidad de Oviedo. Sus orientaciones, sus consejos y su generosidad para conmigo han sido fundamentales para llevar a término este trabajo.

Por otro lado la empresa GESTOR.EX S.C. me permitió contar con un respaldo laboral para colaborar en este proyecto, lo cual ha sido para mí motivo de orgullo y gran satisfacción personal y profesional.

Por supuesto no puedo dejar de mencionar a todos mis compañeros del proyecto GADE4All, con quienes he compartido largas horas de trabajo, a veces preocupaciones y otras muchas alegrías.

Finalmente, no puede faltar el agradecimiento a mi hermana, mis padres y a mis amigos por su apoyo incondicional y por haberme hecho mantener la ilusión, incluso en los momentos en que el trabajo se hizo más cuesta arriba.

A todos ellos, y a quienes no he podido citar pero que también han sido parte importante de mi vida en estos meses, les dirijo desde aquí mi más sincero agradecimiento.

Resumen

El objetivo de este trabajo de fin de máster es la definición, creación e implementación de una herramienta que permita a usuarios sin conocimientos previos de programación la creación de forma sencilla, de videojuegos de acción en 2D para dispositivos móviles con sistema operativo Android.

Mediante la utilización de una instancia de un DSL (Domain Specific Language) de creación de videojuegos, se especifican los elementos y reglas necesarios para la construcción de un juego de acción en 2D completo: elementos a utilizar, características y comportamientos. A través de plantillas se obtiene el código del videojuego que será completo e instalable en un dispositivo móvil Android.

Para garantizar que el uso de la herramienta sea sencillo, se contará con un Editor, que permitirá establecer todos los elementos del juego de forma gráfica y mediante sencillos pasos y acciones.

Este trabajo fin de máster está enmarcado dentro del proyecto "Gade4All: Plataforma Genérica para facilitar el desarrollo de videojuegos y software de entretenimiento multiplataforma", cofinanciado por el Ministerio de Industria, Energía y Turismo, dentro del Plan Nacional de Investigación Científica, Desarrollo e Innovación Tecnológica.

En el desarrollo de Gade4All participan varias empresas y universidades, entre la que se encuentra la Universidad de Oviedo, por medio del grupo de investigación "MDE Research Group".

Palabras Clave

LENGUAJE DE DOMINIO ESPECÍFICO (DSL), ANDROID, VIDEOJUEGOS, DISPOSITIVOS MÓVILES, MDE, MDA, MOTOR DE TRANSFORMACIÓN, PLANTILLA.

Abstract

The main objective of this master's thesis is to define, create and implement a tool which will allow users without programming knowledge to easily develop 2D action video games for mobile devices that run on an Android operative system.

By using a game development DSL (Domain Specific Language) request, we can set the parameters and rules that define a 2D action game: the in-game elements, characteristics and behaviors. The code for that videogame, which will be fully installable in Android mobile devices, is obtained through templates.

To make sure that the use of this tool is easy enough, there will be an editor, which will allow the establishment of all the game's elements in a very visual way and through straightforward steps and actions.

This master's thesis is a part of a project known as "Gade4all: Generic Platform for Making the Development of Video Games and Multiplatform Entertainment Software Easier", cofinanciated by the Ministry of Industry, Energy and Tourism, within the Scientific Research, Development and Technological Innovation National Plan.

Many companies and universities are participating in the development of Gade4all, like University of Oviedo, through its "MDE Research Group".

Keywords

DOMAIN SPECIFIC LANGUAGE (DSL), ANDROID, VIDEOGAMES, MOBILE DEVICES, MDE, MDA, TRANSFORM ENGINES, TEMPLATE.

Índice General

CAPÍTULO 1. MEMORIA DEL PROYECTO 23

1.1 Resumen de la Motivación, Objetivos y Alcance del Proyecto 23
1.1.1 Resumen de la motivación 23
1.1.2 Objetivos del proyecto 24
1.1.3 Alcance del proyecto 24
1.2 Resumen de Todos los Aspectos 25
1.3 Proyecto GADE4ALL 25
1.4 Metodología del proyecto 25
1.5 Estructura de la memoria 28

CAPÍTULO 2. INTRODUCCIÓN 31

2.1 Justificación del Proyecto 31
2.2 Objetivos del Proyecto 32
2.3 Estudio de la Situación Actual 33
2.3.1 Evaluación de alternativas 34

CAPÍTULO 3. ASPECTOS TEÓRICOS 55

3.1 Android 55
3.2 Domain Spedific Language 55
3.3 eXtensible Markup Language 56
3.4 eXtensible Application Markup Language 57
3.5 Model Driven Engineering 57
3.6 Model Driven Architecture 57
3.7 Transformación de modelos 58
3.8 Unified Modeling Language 58
3.9 Domain Specific Modeling 58
3.10 Windows Presentation Foundation 58
3.11 .NET 59

CAPÍTULO 4. PLANIFICACIÓN DEL PROYECTO Y RESUMEN DE PRESUPUESTOS 61

4.1 Planificación 61
4.1.1 Cronograma 62
4.1.2 Diagrama de Gantt 63
4.2 Resumen del Presupuesto 66

CAPÍTULO 5. ANÁLISIS 69

5.1 Definición del Sistema 69
5.1.1 Determinación del Alcance del Sistema 69
5.2 Requisitos del Sistema 70
5.2.1 Obtención de los Requisitos del Sistema 70
5.2.2 Identificación de Actores del Sistema 73
5.2.3 Especificación de Casos de Uso 74
5.3 Identificación de los Subsistemas en la Fase de Análisis 87
5.3.1 Descripción de los Subsistemas 87

5.3.2 Descripción de los Interfaces entre Subsistemas 88
5.4 Diagrama de Clases Preliminar del Análisis 89
5.4.1 Diagrama de Clases 89
5.4.2 Descripción de las Clases 92
5.5 Análisis de Casos de Uso y Escenarios 104
5.5.1 Caso de Uso: Crear Videojuego 104
5.5.2 Caso de Uso: Guardar Estado Videojuego 106
5.5.3 Caso de Uso: Cargar Estado Videojuego 107
5.5.4 Caso de Uso: Exportar Videojuego 108
5.5.5 Caso de Uso: Crear/Editar instancia DSL 109
5.5.6 Caso de Uso: Crear/Editar Plantilla 110
5.5.7 Caso de Uso: Jugar Videojuego 111
5.6 Análisis de Interfaces de Usuario 112
5.6.1 Descripción de la Interfaz 112
5.6.2 Descripción del Comportamiento de la Interfaz 118
5.6.3 Diagrama de Navegabilidad 119
5.7 Especificación del Plan de Pruebas 120
5.7.1 Pruebas Unitarias 120
5.7.2 Pruebas de Integración 120
5.7.3 Pruebas del Sistema 120
5.7.4 Pruebas de Usabilidad 120
5.7.5 Plan de pruebas 121

CAPÍTULO 6. DISEÑO DEL SISTEMA 145

6.1 Arquitectura del Sistema 145
6.1.1 Diagramas de Paquetes 145
6.2 Diseño de Clases 147
6.2.1 Diagrama de Clases 147
6.3 Diagramas de Interacción y Estados 155
6.3.1 Caso de Uso: Crear Videojuego 155
6.4 Diagramas de Actividades 156
6.5 Diseño de la Interfaz 159
6.5.1 Pantalla principal 159
6.5.2 Menú de la aplicación 160
6.5.3 Pantalla configuración propiedades globales 162
6.5.4 Pantalla configuración propiedades específicas 163
6.5.5 Pantalla menú principal 164
6.5.6 Pantalla menú opciones 165
6.5.7 Pantalla menú de selección de nivel 166
6.5.8 Pantalla carga 167
6.5.9 Pantalla de juego 168
6.5.10 Pantalla menú pausa 169
6.5.11 Pantalla menú victoria nivel 170
6.5.12 Pantalla menú derrota nivel 171
6.5.13 Pantalla menú fin juego 172
6.5.14 Pantalla personaje principal 173
6.5.15 Pantalla enemigos 174
6.5.16 Pantalla meta 175
6.5.17 Pantalla escenario 176

6.5.18 Pantalla selección ruta generación 179
6.5.19 Avisos y Mensajes 180
6.6 DISEÑO DE LA INSTANCIA DEL DSL 185
6.6.1 Descripción y funcionamiento 185
6.6.2 Elementos 186
6.7 DISEÑO DE LA PLANTILLA 207
6.7.1 Descripción y funcionamiento 207
6.7.2 Estructura de la plantilla 207
6.7.3 Valores variables 209
6.7.4 Vistas 210
6.7.5 Cargador de niveles 210
6.8 ESPECIFICACIÓN TÉCNICA DEL PLAN DE PRUEBAS 211
6.8.1 Pruebas Unitarias 211
6.8.2 Pruebas de Integración y del sistema 215
6.8.3 Pruebas de Usabilidad 215

CAPÍTULO 7. IMPLEMENTACIÓN DEL SISTEMA 223

7.1 LENGUAJES DE PROGRAMACIÓN 223
7.1.1 Java 223
7.1.2 C# 224
7.1.3 XML 224
7.1.4 XAML 225
7.2 HERRAMIENTAS Y PROGRAMAS USADOS PARA EL DESARROLLO 226
7.2.1 Visual Studio 2010 226
7.2.2 Eclipse 226
7.3 CREACIÓN DEL SISTEMA 227
7.3.1 Descripción Detallada de las Clases 227

CAPÍTULO 8. DESARROLLO DE LAS PRUEBAS 241

8.1 PRUEBAS UNITARIAS 241
8.1.1 Configuración Propiedades Globales 241
8.1.2 Configuración propiedades específicas 242
8.1.3 Pantallas menús 242
8.1.4 Personaje principal 243
8.1.5 Enemigos 243
8.1.6 Meta 244
8.1.7 Escenario 244
8.1.8 Utilidades 245
8.2 PRUEBAS DE INTEGRACIÓN Y DEL SISTEMA 245
8.3 PRUEBAS DE USABILIDAD 279
8.3.1 Resultados de las Pruebas de Usabilidad 279
8.3.2 Análisis de los resultados obtenidos 285

CAPÍTULO 9. MANUALES DEL SISTEMA 287

9.1 MANUAL DE INSTALACIÓN 287
9.1.1 Requisitos 287
9.1.2 Pasos a seguir 287
9.2 MANUAL DE EJECUCIÓN 291
9.3 MANUAL DE USUARIO 292

9.3.1 Requisitos ... 292
9.3.2 Tutorial de creación de un videojuego con la herramienta Gade4all ... 293
9.3.3 Compilar juego ... 349
9.4 MANUAL DEL PROGRAMADOR ... 352
9.4.1 Interfaz ... 352
9.4.2 Analizador ... 354
9.4.3 Instancia del DSL ... 354
9.4.4 Plantilla Android ... 354

CAPÍTULO 10. CONCLUSIONES Y AMPLIACIONES ... 357

10.1 CONCLUSIONES ... 357
10.2 AMPLIACIONES ... 359
10.2.1 Mejorar proceso generación videojuego ... 359
10.2.2 Nuevas funcionalidades ... 359
10.2.3 Biblioteca de juegos ... 359
10.2.4 Mejorar funcionalidad editor de niveles ... 359
10.2.5 Ampliación a videojuegos 3D ... 360

CAPÍTULO 11. APLICACIONES REALES DEL PROYECTO ... 361

11.1 VIDEOJUEGOS BASE ... 361
11.2 VIDEOJUEGOS DESARROLLADOS Y DISPONIBLES EN ANDROID ... 376
11.2.1 Regateo Extremo ... 376
11.2.2 Balonazo a la grada ... 378
11.2.3 Para los balones ... 379
11.2.4 Tiro Skeet ... 381
11.2.5 Ace Athlete ... 383

CAPÍTULO 12. REFERENCIAS BIBLIOGRÁFICAS ... 387

12.1 LIBROS Y ARTÍCULOS ... 387
12.2 REFERENCIAS EN INTERNET ... 388

CAPÍTULO 13. APÉNDICES ... 389

13.1 GLOSARIO Y DICCIONARIO DE DATOS ... 389
13.2 CONTENIDO ENTREGADO EN EL CD-ROM ... 390
13.2.1 Contenidos ... 390
13.3 ÍNDICE ALFABÉTICO ... 391
13.4 ANEXO I. INSTANCIA DEL DSL GENERADA CON LA HERRAMIENTA. ... 392

Índice de Figuras

Ilustración 1.1. Identificación de partes variables de los videojuegos 26
Ilustración 1.2. Construcción de la instancia del DSL con las partes variables....26
Ilustración 1.3. Construcción de la plantilla....27
Ilustración 1.4. Funcionamiento Analizador 27
Ilustración 2.1. Comparativa herramientas 34
Ilustración 2.1. Logo GameSalad 35
Ilustración 2.2 . GameSalad, actores y escenas....36
Ilustración 2.3.GameSalad, actores 37
Ilustración 2.4. GameSalad, comportamiento actores....37
Ilustración 2.5. GameSalad, visualización nivel 38
Ilustración 2.6. Logo GameMaker....39
Ilustración 2.7. Herramienta GameMaker 40
Ilustración 2.8. GameMaker, escenario....41
Ilustración 2.9. GameMaker, comportamientos elementos 42
Ilustración 2.10. GameMaker, creación nivel 43
Ilustración 2.11. GameMaker, nivel creado....43
Ilustración 2.12. Logo AppInvertor....45
Ilustración 2.13. Herramienta AppInventor....46
Ilustración 2.14. AppInventor, editor bloques....47
Ilustración 2.15. AppInventor, creación pantalla....48
Ilustración 2.16. AppInventor, barra estado....48
Ilustración 2.17. Logo Stencyl....50
Ilustración 2.18. Stencyl, nivel ejemplo....51
Ilustración 2.19. Stencyl, programación comportamientos 52
Ilustración 3.1 Transformación de modelos independientes de plataforma....56
Ilustración 4.1. Cronograma etapa desarrollo Videojuego base 62
Ilustración 4.2. Cronograma etapa desarrollo Plantilla....62
Ilustración 4.3. Cronograma etapa desarrollo Interfaz 62
Ilustración 4.4. Diagrama Gantt etapa desarrollo Videojuego Base....63
Ilustración 4.5. Diagrama Gantt etapa desarrollo Plantilla 64
Ilustración 4.6. Diagrama Gantt etapa desarrollo Interfaz....65
Ilustración 5.1. Caso de Uso: Crear Videojuego 74
Ilustración 5.2. Subcasos de Uso de CrearVideojuego 75
Ilustración 5.3. Subcaso de Uso: Configurar propiedades globales 76
Ilustración 5.4. Subcaso de Uso: Configurar propiedades específicas 76
Ilustración 5.5. Subcaso de Uso: Configurar pantalla principal 77
Ilustración 5.6. Subcaso de Uso: Configurar pantalla opciones....77
Ilustración 5.7. Subcaso de Uso: Configurar pantalla selección de nivel 78
Ilustración 5.8. Subcaso de Uso: Configuración pantalla carga 78
Ilustración 5.9. Subcaso de Uso: Configurar pantalla pausa 79
Ilustración 5.10. Subcaso de Uso: Configurar pantalla juego 79
Ilustración 5.11. Subcaso de Uso: Configurar pantalla derrota nivel....80
Ilustración 5.12. Subcaso de Uso: Configurar pantalla victoria nivel....80
Ilustración 5.13. Subcaso de Uso: Configurar pantalla fin juego 81
Ilustración 5.14. Subcaso de Uso: Crear personaje principal....81

Ilustración 5.15. Subcaso de Uso: Crear enemigos **82**
Ilustración 5.16. Subcaso de Uso: Crear meta **82**
Ilustración 5.17. Subcaso de Uso: Crear nivel **83**
Ilustración 5.18. Caso de Uso: Guardar videojuego **83**
Ilustración 5.19. Caso de Uso: Cargar Videojuego **84**
Ilustración 5.20. Caso de Uso: Exportar Videojuego **84**
Ilustración 5.21. Caso de Uso: Crear/Editar instancia DSL **85**
Ilustración 5.22. Caso de Uso: Crear/Editar Plantilla **85**
Ilustración 5.23. Caso de Uso: Jugar Videojuego **86**
Ilustración 5.24. Diagrama de Paquetes Análisis Global **89**
Ilustración 5.25. Diagrama de Clases Análisis Subsistema interfaz **90**
Ilustración 5.26. Diagrama de Clases Análisis Subsistema Analizador **90**
Ilustración 5.27. Diagrama de Clases Análisis Subsistema Plantilla **91**
Ilustración 5.28. Descripción actividades escenario: Crear videojuego **104**
Ilustración 5.29. Descripción actividades escenario: Guardar Estado Videojuego **106**
Ilustración 5.30. Descripción actividades escenario: Cargar Estado Videojuego **107**
Ilustración 5.31. Descripción actividades escenario: Exportar Videojuego **108**
Ilustración 5.32. Descripción actividades escenario: Crear/Editar instancia DSL **109**
Ilustración 5.33. Descripción actividades escenario: Crear/Editar Plantilla **110**
Ilustración 5.34. Descripción actividades escenario: Jugar Videojuego **111**
Ilustración 5.35. Ventana contenedora interfaz **112**
Ilustración 5.36. Análisis, pantalla principal **113**
Ilustración 5.37. Análisis, pantalla configuración globales **113**
Ilustración 5.38. Análisis, pantalla configuración específicas **114**
Ilustración 5.39. Análisis, pantalla configuración menús **114**
Ilustración 5.40. Análisis, pantalla creación personajes **115**
Ilustración 5.41. Análisis, pantalla creación nivel **116**
Ilustración 5.42. Análisis, pantalla generación juego **117**
Ilustración 6.1. Diagrama paquetes Global **145**
Ilustración 6.2. Diagrama de clases diseño: Subsistema Interfaz **148**
Ilustración 6.3. Diagrama de clases diseño: clase Scenery **149**
Ilustración 6.4. Diagrama de clases diseño: clase Store **150**
Ilustración 6.5. Diagrama de clases diseño: clase Almacen **150**
Ilustración 6.6. Diagrama de clases diseño: clase Serializer **151**
Ilustración 6.7. Diagrama de clases diseño Subsistema analizador **151**
Ilustración 6.8. Diagrama de clases diseño Subsistema Plantilla **152**
Ilustración 6.9. Diagrama de clases diseño: clase Model **153**
Ilustración 6.10. Diagrama de clases diseño: clase Screen **154**
Ilustración 6.11. Diagrama interacción Caso de Uso: Crear Videojuego **155**
Ilustración 6.12. Diagrama actividad: Crear Videojuego **157**
Ilustración 6.13. Pantalla principal **159**
Ilustración 6.14. Seleccionar ruta almacenar **160**
Ilustración 6.15. Menú de la aplicación **160**
Ilustración 6.16. Barra botones interfaz **161**
Ilustración 6.17. Menú opciones **161**
Ilustración 6.18. Botones navegación **161**
Ilustración 6.19. Pantalla configuración propiedades globales **162**
Ilustración 6.20. Pantalla configuración propiedades específicas **163**
Ilustración 6.21. Configuración menú principal **164**
Ilustración 6.22. Configurar menú opciones **165**

Ilustración 6.23. Configuración menú selección nivel 166
Ilustración 6.24. Configurar menú carga 167
Ilustración 6.25. Configuración menú juego 168
Ilustración 6.26. Configurar menú pausa 169
Ilustración 6.27. Configurar menú victoria nivel 170
Ilustración 6.28. Configurar menú derrota nivel 171
Ilustración 6.29. Configurar menú fin juego 172
Ilustración 6.30. Configurar personaje principal 173
Ilustración 6.31. Configurar enemigos 174
Ilustración 6.32. Configurar meta 175
Ilustración 6.33. Definir escenario 176
Ilustración 6.34. Configurar propiedades nivel 177
Ilustración 6.35. Configurar descripción nivel 177
Ilustración 6.36. Configurar delimitadores 178
Ilustración 6.37. Configurar imagen fondo 178
Ilustración 6.38. Generación juego 179
Ilustración 6.39. Mensaje guardar estado 180
Ilustración 6.40. Mensaje creación personaje principal 180
Ilustración 6.41. Mensaje creación enemigos 181
Ilustración 6.42. Mensaje nivel creado 181
Ilustración 6.43. Mensaje nivel guardado 181
Ilustración 6.44. Mensaje eliminar nivel 0 182
Ilustración 6.45. Mensaje eliminar personajes ya creados 182
Ilustración 6.46. Mensaje creado 183
Ilustración 6.47. Mensaje generando juego 183
Ilustración 6.48. Mensaje juego generado 184
Ilustración 6.49. Elemento game 186
Ilustración 6.50. Elemento global 186
Ilustración 6.51. Elemento screens 187
Ilustración 6.52. Subelementos button 188
Ilustración 6.53. Subelementos image 188
Ilustración 6.54. Elemento main_screen 189
Ilustración 6.55. Elemento options_screen 190
Ilustración 6.56. Elemento select_level_screen 191
Ilustración 6.57. Elemento loading_screen 192
Ilustración 6.58. Elemento pause_screen 192
Ilustración 6.59. Elemento end_level_screen 193
Ilustración 6.60. Elemento scoreboard 194
Ilustración 6.61. Elemento lose_level_screen 194
Ilustración 6.62. Elemento end_game_screen 195
Ilustración 6.63. Elemento game_screen 196
Ilustración 6.64. Elemento status_bar 196
Ilustración 6.65. Elemento game_properties 197
Ilustración 6.66. Elemento levels 198
Ilustración 6.67. Elemento stage 199
Ilustración 6.68. Elemento tops 201
Ilustración 6.69. Elemento character 201
Ilustración 6.70. Elemento npcs 204
Ilustración 6.71. Elemento npc 204
Ilustración 6.72. Elemento target 206

Ilustración 6.73. Estructura carpetas plantilla 207
Ilustración 6.74. Partes variables 209
Ilustración 7.1 . Ejemplo código Java 223
Ilustración 7.2 . Ejemplo código C# 224
Ilustración 7.3 . Ejemplo código XML 225
Ilustración 7.4 . Ejemplo código XAML 225
Ilustración 7.5 . Icono Visual Studio 2010 226
Ilustración 7.6 . Icono Eclipse 226
Ilustración 8.1. Preguntas carácter general perfil usuario 1 279
Ilustración 8.2. Preguntas carácter general perfil usuario 2 280
Ilustración 8.3. Preguntas carácter general perfil usuario 3 280
Ilustración 8.4. Preguntas carácter general perfil usuario 4 281
Ilustración 8.5. Preguntas cortas perfil usuario 1 282
Ilustración 8.6. Preguntas cortas perfil usuario 2 282
Ilustración 8.7. Preguntas cortas perfil usuario 3 283
Ilustración 8.8. Preguntas cortas perfil usuario 4 284
Ilustración 9.1. Archivos de instalación 287
Ilustración 9.2. Instalación .NETFramework 4.0 288
Ilustración 9.3. Instalando .NETFramework 4.0 288
Ilustración 9.4. Instalación .NETFramework 4.0 finalizada 288
Ilustración 9.5. Instalar herramienta 289
Ilustración 9.6. Seleccionar carpeta instalación 289
Ilustración 9.7. Carpeta Temp 290
Ilustración 9.8. Instalando 290
Ilustración 9.9. Icono aplicación 291
Ilustración 9.10. Ejecutar aplicación 291
Ilustración 9.11. Pantalla inicial 294
Ilustración 9.12. Selección Workspace 295
Ilustración 9.13. Menú de la aplicación 296
Ilustración 9.14. Barra de accesos directos 296
Ilustración 9.15. Menú Archivo 296
Ilustración 9.16. Menú navegación 297
Ilustración 9.17. Configuración propiedades globales 298
Ilustración 9.18. Ejemplo pantalla propiedades globales 299
Ilustración 9.19. Configuración propiedades específicas 300
Ilustración 9.20. Ejemplo pantalla propiedades específicas 302
Ilustración 9.21. Estructura pantallas menús 303
Ilustración 9.22. Nombre de la pantalla 303
Ilustración 9.23. Parámetros configurables de los elementos de los menús del juego 304
Ilustración 9.24. Ejemplo de parámetros configurables de los elementos de los menús del juego 305
Ilustración 9.25. Configuración pantalla principal del juego 306
Ilustración 9.26. Configuración botones pantalla menú principal 307
Ilustración 9.27. Ejemplo pantalla principal configurada 307
Ilustración 9.28. Configuración pantalla de opciones del juego 308
Ilustración 9.29. Pantalla opciones juego configurada 309
Ilustración 9.30. Configuración pantalla selección de nivel del juego 310
Ilustración 9.31. Pantalla selección de nivel juego configurada 311
Ilustración 9.32. Configuración pantalla de carga del juego 312
Ilustración 9.33. Pantalla Loading configurada 313
Ilustración 9.34. Configuración pantalla juego 314

Ilustración 9.35. Icono y marcador 314
Ilustración 9.36. Imagen barra de estado de pantalla de juego 315
Ilustración 9.37. Pantalla juego configurada 316
Ilustración 9.38. Configuración pantalla pausa del juego 317
Ilustración 9.39. Pantalla pausa configurada 318
Ilustración 9.40. Pantalla fin de nivel con éxito 319
Ilustración 9.41. Pantalla fin de nivel configurada 320
Ilustración 9.42. Configuración pantalla derrota 321
Ilustración 9.43. Pantalla derrota configurada 322
Ilustración 9.44. Configuración pantalla fin de juego 323
Ilustración 9.45. Pantalla fin de juego configurada 324
Ilustración 9.46. Instancia del DSL generada 324
Ilustración 9.47. Configuración personaje principal del juego 325
Ilustración 9.48. Configuración propiedades personaje principal 326
Ilustración 9.49. Área de colisión 327
Ilustración 9.50. Tamaños áreas de colisión 327
Ilustración 9.51. Frames 329
Ilustración 9.52. Pantalla personaje principal configurada 329
Ilustración 9.53. Opciones de la pantalla de creación de personaje principal 330
Ilustración 9.54. Personaje principal añadido a la lista 330
Ilustración 9.55. Configuración Enemigo 331
Ilustración 9.56. Pantalla enemigo configurada 332
Ilustración 9.57. Configuración meta 333
Ilustración 9.58. Pantalla meta configurada 334
Ilustración 9.59. Configuración niveles del juego 335
Ilustración 9.60. Dibujar elemento en escenario 336
Ilustración 9.61. Pasos creación nivel 337
Ilustración 9.62. Propiedades del nivel 338
Ilustración 9.63. Propiedades descripción del nivel 339
Ilustración 9.64. Descripción de nivel 339
Ilustración 9.65. Miniatura del nivel 340
Ilustración 9.66. Delimitadores del nivel 340
Ilustración 9.67. Propiedades del fondo 341
Ilustración 9.68. Velocidad del escenario 341
Ilustración 9.69. Avisos escenario. 342
Ilustración 9.70. Eliminación avisos escenario. 342
Ilustración 9.71. Coordenadas fuera pantalla. 343
Ilustración 9.72. Coordenadas negativas 344
Ilustración 9.73. Escenario vertical 344
Ilustración 9.74. Opciones niveles 345
Ilustración 9.75. Cargar niveles 345
Ilustración 9.76. Botones vaciar y restaurar escenario 346
Ilustración 9.77. Configuración pantalla plataforma videojuego 347
Ilustración 9.78. Imagen espera generación 348
Ilustración 9.79. Juego generado correctamente 348
Ilustración 9.80. Proyecto Eclipse generado para Android 349
Ilustración 9.81. Importar proyecto Eclipse 350
Ilustración 9.82. Ejecutar proyecto Eclipse 351
Ilustración 9.83. Juego desplegado en emulador Android 351
Ilustración 11.1 . Icono juego Bichos 361

Ilustración 11.2 . Explicación nivel uno juego Bichos **362**
Ilustración 11.3 . Nivel 1 juego Bichos **362**
Ilustración 11.4 . Explicación nivel dos juego Bichos **363**
Ilustración 11.5 . Nivel 2 juego Bichos **363**
Ilustración 11.6 . Explicación nivel tres juego Bichos **363**
Ilustración 11.7 . Nivel 3 juego Bichos **364**
Ilustración 11.8 .Enemigo con aspecto de vida extra **364**
Ilustración 11.9 . Enemigo con aspecto de hormiga **364**
Ilustración 11.10 . Enemigo con aspecto de abeja **364**
Ilustración 11.11 . Imagen caza-mariposas **365**
Ilustración 11.12 . Botón jugar juego Bichos **365**
Ilustración 11.13 .Botón salir juego Bichos **365**
Ilustración 11.14 . Selección de nivel juego Bichos **365**
Ilustración 11.15 . Iconos de la barra de estado **366**
Ilustración 11.16 . Pantalla de pausa juego Bichos **366**
Ilustración 11.17 . Pantalla de victoria juego Bichos **367**
Ilustración 11.18 . Icono juego Herocopter **367**
Ilustración 11.19 Explicación nivel uno juego Hercopter **368**
Ilustración 11.20 . Nivel 1 juego Herocopter **368**
Ilustración 11.21 . Explicación nivel dos juego Herocopter **368**
Ilustración 11.22 . Nivel 2 juego Herocopter **369**
Ilustración 11.23 . Explicación nivel tres juego Herocopter **369**
Ilustración 11.24 . Nivel 3 juego Herocopter **369**
Ilustración 11.25 . Character con aspecto de helicoptero **370**
Ilustración 11.26 .Enemigo con aspecto de bomba **370**
Ilustración 11.27 . Enemigo con aspecto de moneda **370**
Ilustración 11.28 . Botón jugar juego Herocopter **370**
Ilustración 11.29 . Botón salir juego Herocopter **370**
Ilustración 11.30 . Pantalla selección de nivel juego Herocopter **371**
Ilustración 11.31 . Iconos de barra de estado juego Herocopter **371**
Ilustración 11.32 . Pantalla de pausa juego de Herocopter **371**
Ilustración 11.33 . Pantalla de victoria juego Herocopter **372**
Ilustración 11.34. Regateo Extremo, pantalla principal **376**
Ilustración 11.35. Regateo Extremo, pantalla juego **377**
Ilustración 11.36. Regateo Extremo en Google Play **377**
Ilustración 11.37. Balonazo A La Grada, pantalla principal **378**
Ilustración 11.38. Balonazo A La Grada, pantalla juego **378**
Ilustración 11.39. Balonazo A La Grada en Google Play **379**
Ilustración 11.40. Para Los Balones, pantalla principal **380**
Ilustración 11.41. Para Los Balones, pantalla juego **380**
Ilustración 11.42. Para Los Balones en Google Play **381**
Ilustración 11.43. Tiro Skeet, pantalla principal **382**
Ilustración 11.44. Tiro Skeet, pantalla juego **382**
Ilustración 11.45. Tiro Skeet en Google Play **383**
Ilustración 11.46. Ace Athlete, pantalla principal **384**
Ilustración 11.47. Ace Athlete, pantalla juego **384**
Ilustración 11.48. Ace Athlete en Google Play **385**

Capítulo 1. Memoria del Proyecto

1.1 Resumen de la Motivación, Objetivos y Alcance del Proyecto

1.1.1 Resumen de la motivación

Un videojuego es un software cuya creación está destinada al entretenimiento y está basado en la interacción entre el jugador y el dispositivo en el que se ejecuta el videojuego. Desde sus inicios en la década de los 50 hasta hoy día, los videojuegos han pasado de ser un pasatiempo para jóvenes estudiantes de ingeniería a convertirse en la industria del ocio más poderosa.

En sus comienzos los videojuegos eran desarrollados por grupos muy reducidos de personas y solamente se podían ejecutar en un número muy limitado de dispositivos, algunos de los cuales tenían un elevado coste. Según ha ido creciendo la industria de los videojuegos también han aumentado los equipos de desarrollo y la variedad de los dispositivos destinados a la ejecución de los mismos, ya que con el furor de la telefonía móvil las empresas dedicadas a la fabricación del software de los móviles ampliaron la producción de videojuegos a estos dispositivos.

La reducción del tamaño de los dispositivos necesarios para jugar a los videojuegos aumentó notablemente la cantidad de usuarios de los mismos, ya que ahora no necesitaban dispositivos con cables para poder jugar.

Aparecieron nuevos sistemas operativos entre los cuales destacan Symbian, desarrollado por Nokia, Windows Mobile, la versión del móvil de Microsoft e IOS para iPhone, la versión del Mac para móvil. Pero Google irrumpió en el mundo de la telefonía móvil con la creación del sistema operativo Android. Por tanto, con la aparición del teléfono inteligente o *Smartphone* cambiaron las cosas, se disparó el desarrollo de aplicaciones y videojuegos para dispositivos móviles, lo cual ha desencadenado el desarrollo de diferentes aplicaciones software para que uno mismo pueda crear sus propios videojuegos. Pero el uso de todos esos programas requiere tener conocimientos básicos o avanzados de determinados lenguajes de programación.

Y aquí es donde surge la idea de este proyecto: desarrollo de videojuegos de acción en 2D basado en DSL para Android. Con esta aplicación el usuario podrá crear sus propios videojuegos para su dispositivo móvil sin tener ningún tipo de conocimiento sobre programación.

De esta manera los videojuegos no tendrán por qué ser creados solamente por desarrolladores, cualquiera que tenga una idea para un videojuego de acción podrá llevarla a cabo de manera sencilla con esta aplicación.

1.1.2 Objetivos del proyecto

El principal objetivo que se pretende alcanzar con este proyecto es el desarrollo de una herramienta que permita crear videojuegos desde cero.

Otros objetivos de este proyecto son los siguientes:

- Creación de una instancia del DSL para la definición de videojuegos de acción de tipo Touch.
- Construcción de una interfaz que permita el desarrollo de videojuegos de acción en 2D desde cero a usuarios sin conocimientos de programación.
- Minimizar los gastos y los recursos.
- Mejorar la rapidez y efectividad en los procesos de desarrollo de videojuegos.
- Creación de una instancia del DSL genérica independiente de la plataforma y reutilizable para la definición de juegos con independencia de la plataforma destino.
- Reducción del tiempo de desarrollo de videojuegos.

1.1.3 Alcance del proyecto

Este proyecto está integrado en otro más grande realizado para la creación de videojuegos para multiplataforma (Android, iPhone, WindowsPhone y HTML5) llamado **GADE4ALL**. Este prototipo se corresponde con la parte del proyecto GADE4ALL enfocada a la creación de videojuegos de acción de tipo Touch para Android.

Muchos videojuegos comparten una serie de patrones de comportamiento, de los cuales se pretende adoptar algunos de los más representativos para permitir la definición de varios juegos de forma rápida, simplemente modificando algunas de las propiedades de dichos comportamientos.

Para lograr esa adopción de los comportamientos es necesario definir los elementos variantes que describen un videojuego. Entre ellos se pueden mencionar aspectos gráficos, sonidos, personajes y elementos a utilizar, y sobre todo, definir el comportamiento de dichos elementos para establecer las reglas del juego.

Una herramienta que genere "cualquier" juego supone una complejidad muy elevada, por tanto, este proyecto se ha enfocado a una tipología concreta, para la que se han definido una serie de reglas, y así se limita el comportamiento de los elementos definidos en el juego.

El usuario diseñador del videojuego tendrá que definir las características del juego mediante una interfaz gráfica, precisando las propiedades configurables de los elementos del

juego, que serán las que formarán las reglas del mismo. Estas propiedades configurables estarán predefinidas para configurar juegos de la tipología elegida.

Por tanto la implementación de este proyecto fin de máster contará con un editor gráfico de videojuegos para juegos de acción de tipo Touch.

1.2 Resumen de Todos los Aspectos

El sistema desarrollado es una aplicación de escritorio diseñada para crear videojuegos para dispositivos móviles con sistema operativo Android, desde cero.

El usuario ejecutará la aplicación y mediante una interfaz sencilla e intuitiva podrá diseñar su propio videojuego teniendo únicamente que escoger los diferentes elementos que formarán parte del juego. Y se obtendrá un videojuego listo para instalar en un dispositivo móvil.

La novedad que diferencia este proyecto de otras aplicaciones para el diseño de videojuegos para móviles es que el usuario no necesita tener ningún conocimiento previo de programación. De esta manera se acerca el diseño de videojuegos a una amplia gama de usuarios.

1.3 Proyecto GADE4ALL

Como se ha mencionado anteriormente, el presente Proyecto Fin de Máster forma parte de un proyecto realizado para una empresa, cuyo nombre es GADE4ALL.

El objetivo de ese proyecto es la creación de una instancia del DSL única para el desarrollo de diferentes familias de juegos en distintas plataformas, así como el diseño y construcción de una herramienta que permita al usuario la creación de esos videojuegos a través de una interfaz gráfica, en la que se escogerán los elementos y las propiedades que formarán parte del lenguaje del videojuego.

1.4 Metodología del proyecto

EL desarrollo de este proyecto se ha dividido en varias fases, con diferentes tareas en cada una en cada una de ellas.

La metodología utilizada para la creación de este software es de tipo *Bottom-up*, que consiste en el desarrollo individual de cada parte con detalle, y posteriormente en su enlace para obtener un componente más grande.

Como primer paso de este proyecto se realizó un estudio de las herramientas similares utilizadas para la creación de videojuegos que se encuentran en el mercado. Se analizaron las características y funcionamiento de cada una de ellas para identificar las diferencias con la

herramienta que se iba a desarrollar en este proyecto. Una vez analizados los resultados del estudio se fijaron los objetivos del proyecto y se comenzó su desarrollo.

Inicialmente se crearon dos videojuegos de acción para la plataforma Android, que serían la base sobre la que se construiría posteriormente el resto del proyecto. Ambos juegos se diseñaron de forma similar, de modo que, si se variaban los valores de ciertos atributos de uno de los juegos, se obtendría el otro y viceversa.

Una vez realizadas las pruebas de los videojuegos en diferentes dispositivos móviles con sistema operativo Android, se identificaron las partes variables de cada uno de ellos, es decir, aquellas partes del código de los juegos que podrían variar para dar lugar a otros videojuegos, como por ejemplo: las dimensiones de la pantalla, las imágenes, etc.

```
/**** VARIABLE PART *****/
public int screenWidth = 480;
public int screenHeight = 320;

public int screenOrientation = SCREEN_ORIENTATION_LANDSCAPE;

/*****************************************/
```

Ilustración 1.1. Identificación de partes variables de los videojuegos

Con esas partes variables identificadas se empezó la construcción del lenguaje basado en DSL (Domain Specific Language) que contendría todos los elementos variables y sus valores. Con este lenguaje se podrán generar diferentes juegos modificando los valores de sus elementos.

```
<global>
    <global_name>Herocopter</global_name>
    <global_name_timespan>8:54</global_name_timespan>
    <global_package_name>com.Touch</global_package_name>
    <global_screen_orientation>SCREEN_ORIENTATION_LANDSCAPE</global_screen_orientation>
    <global_icon_source>ic_launcher</global_icon_source>
    <global_window_width>480</global_window_width>
    <global_window_height>320</global_window_height>
    <global_window_tile_width>480</global_window_tile_width>
    <global_window_tile_height>320</global_window_tile_height>
    <global_window_type_game>0</global_window_type_game>
    <global_window_has_advertising>NO</global_window_has_advertising>
    <global_window_advertising_fixed_position>15</global_window_advertising_fixed_position>
</global>
```

Ilustración 1.2. Construcción de la instancia del DSL con las partes variables

Utilizando el lenguaje construido y los videojuegos creados en la primera fase del proyecto, se construyó una plantilla. Esta plantilla era el resultado de sustituir los valores de los atributos de las partes variables de los videojuegos por las etiquetas de cada uno de los elementos del lenguaje.

```
public int screenWidth = //%global_window_width% ;
public int screenHeight = //%global_window_height% ;

public int screenOrientation = //%global_screen_orientation% ;
```

Ilustración 1.3. Construcción de la plantilla

Ya construida la plantilla, mediante un analizador se sustituirían en ella los valores obtenidos del lenguaje del videojuego.

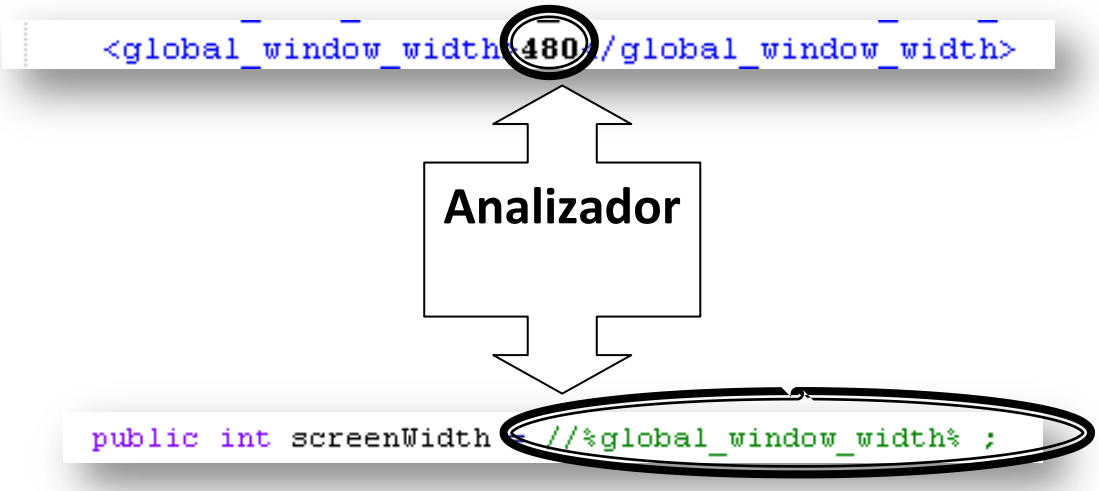
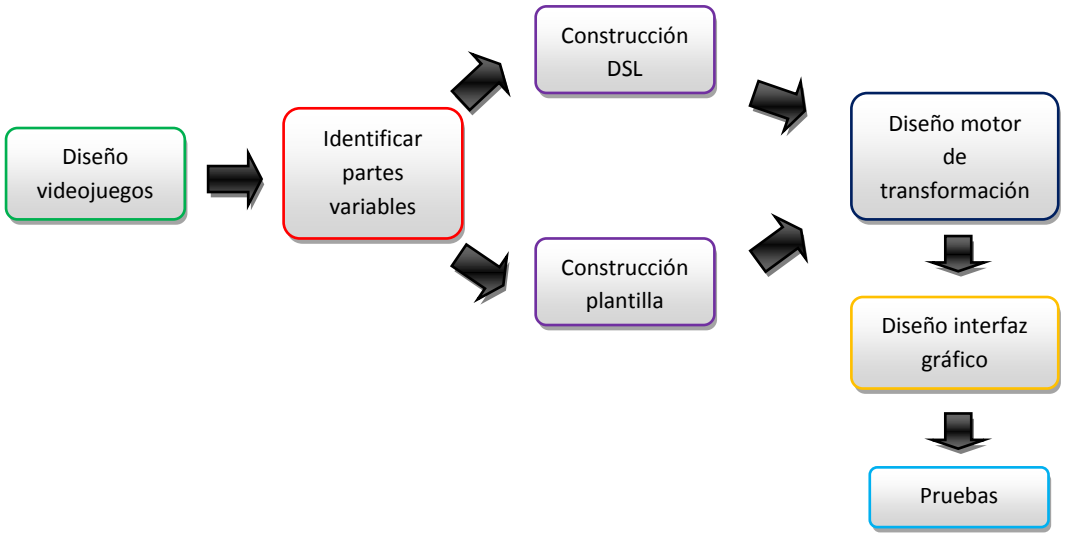


Ilustración 1.4. Funcionamiento Analizador

Para que el usuario no tenga que completar la instancia del DSL directamente, se diseñó una interfaz de usuario, que permitirá a este diseñar de forma sencilla e intuitiva un videojuego sin tener ningún conocimiento de programación. Escogerá las imágenes de los objetos, configurará sus propiedades y podrá diseñar cada nivel del juego a su gusto. Desde este interfaz se obtendrá un proyecto de *Eclipse*, que bastará con importarlo y ejecutarlo para obtener la *apk* del videojuego creado y así poder instalarlo en el móvil.

Como último paso de este proyecto se han realizado las pruebas de cada módulo por separado y de la herramienta completa.



1.5 Estructura de la memoria

A continuación se explica brevemente el contenido de cada capítulo incluido en este documento.

En el capítulo 1, *Memoria del proyecto*, se expone su motivación del presente proyecto, los objetivos que persigue y el alcance del mismo. Además se incluye una breve descripción de la metodología utilizada para su desarrollo y una visión general de los contenidos de la memoria.

El capítulo 2, *Introducción*, describe el motivo por el que se desarrolla este proyecto. Se describe de forma no técnica en qué consiste el proyecto, incluyendo las necesidades que va a cubrir. También se ofrece una explicación general de algunos aspectos relacionados con el tema, como la evolución y situación actual de Android y otras herramientas existentes en el mercado que también sirven para la creación de videojuegos, realizándose un estudio comparativo para obtener las diferencias y similitudes con este proyecto.

En el capítulo 3, *Aspectos teóricos*, se describen brevemente aquellos conceptos, herramientas y tecnologías utilizados: DSL, XML, desarrollo por modelos, etc.

El capítulo 4, *Planificación del proyecto y resumen de presupuestos*, muestra la evolución del desarrollo del proyecto en el tiempo y un presupuesto de la realización del mismo de forma justificada.

El capítulo 5, *Análisis*, contiene toda la especificación de requisitos y la documentación del análisis de la aplicación, a partir de la cual se elaborará posteriormente el diseño. Contendrá unos diagramas de casos de uso (DCU) que especificarán la funcionalidad del sistema. Se realizará un análisis de la interfaz de usuario mediante capturas de pantalla y una breve explicación de cada una de ellas.

En el capítulo 6, *Diseño del sistema*, se describirán: la arquitectura del sistema y del entorno tecnológico que le va a dar soporte; el funcionamiento del interfaz gráfico mediante capturas de pantalla de cada una de las ventanas de la misma; y una explicación técnica del plan de pruebas.

El capítulo 7, *Implementación del sistema*, recoge la información referente a los lenguajes de programación utilizados en la construcción de la herramienta, así como el software empleado.

En el capítulo 8, *Desarrollo de las pruebas*, se analiza el resultado obtenido al realizar las pruebas recogidas en el plan de pruebas diseñado anteriormente.

El capítulo 9, *Manuales del sistema*, se recogen los manuales de instalación, ejecución, usuario y programador sobre el sistema desarrollado para este proyecto.

En el capítulo 10, *Conclusiones y ampliaciones*, se analiza si los resultados obtenidos estaban dentro de lo esperado, y si se han cumplido los objetivos y las expectativas. Se mencionarán las futuras ampliaciones que podrá tener la herramienta, analizando sus ventajas y justificándose también los motivos por los que no se han abordado las ampliaciones en este proyecto.

En el capítulo 11, *Aplicaciones reales del proyecto*, incluye una descripción de los juegos diseñados como base de este proyecto y los desarrollados con la herramienta para su comercialización, su funcionamiento y sus objetivos.

En el capítulo 12, *Referencias bibliográficas*, se indican los libros y artículos manejados durante el desarrollo del proyecto y su documentación, así como las páginas web consultadas.

El último capítulo, *Apéndices*, contiene un glosario y diccionario de datos de todos los términos que se consideren importantes en la aplicación, con una descripción breve de su significado dentro de la misma. Se describe el contenido de los CD's entregados junto con la documentación del proyecto.

Capítulo 2. Introducción

2.1 Justificación del Proyecto

El proyecto que se describe a lo largo de este documento forma parte del proyecto Gade4all, como ya se comentó en el apartado 1.3. La versión completa de Gade4all permite al usuario crear juegos de tipo trivial, estrategia, puzle y plataformas para dispositivos iPhone, Windows Mobile y navegadores web con soporte de HTML5 y para Android.

Debido a la amplitud del proyecto, se ha optado por tomar una parte del mismo para el presente trabajo, la cual se corresponde con la categoría de juegos de acción para dispositivos móviles con sistema operativo Android.

La herramienta desarrollada para este trabajo fin de máster, permitirá al usuario crear videojuegos para Android sin necesidad de poseer conocimientos de programación o especificaciones propias de las aplicaciones para Android. Para lograr este objetivo, se definió un lenguaje de alto nivel que describa los elementos variables de un videojuego, como son los aspectos gráficos, sonidos, personajes y enemigos a utilizar en el juego, y sobre todo, el comportamiento de dichos elementos para establecer las reglas del juego.

La dificultad que presenta este tipo de abstracción aparece a la hora de predefinir los tipos de comportamientos que el usuario de la herramienta asignará a los elementos que está configurando para establecer las reglas de su juego, dado que se pretende que la creación de los videojuegos no requiera conocimientos de programación. Para ello se establecieron y programaron todos esos comportamientos en la plantilla creada en los primeros pasos del desarrollo de este proyecto.

Por otra parte, una de las ventajas que presentan los videojuegos generados con esta herramienta es que el resultado final será un proyecto de aplicación de *Eclipse* que podrá ser modificado desde el entorno de desarrollo de Android como si se tratara de un proyecto realizado desde cero. Esto permitirá la creación de prototipos de videojuegos completamente funcionales que podrían actuar como base para la creación de juegos más complejos, en el caso de que el usuario de la herramienta tuviera los conocimientos necesarios de programación para añadir nuevas funcionalidades a los videojuegos creados.

Desde esta herramienta, utilizando el editor gráfico, se podrán definir y configurar los siguientes elementos:

- Gráficos: imágenes de fondo, botones, iconos e imágenes de los enemigos y del personaje principal.
- Sonidos: música de fondo de los menús, del juego y efectos de sonido asociados a los personajes y enemigos.

- Pantallas de los menús del juego: pantalla principal, opciones, selección de nivel, pausa, barra de estado del juego, fin de nivel y fin del juego.
- Personaje principal y enemigos: serán sobre los que se realizará la interacción en el juego y a los que se les aplicarán comportamientos específicos, como ejemplo tipo movimiento y velocidad, que finalmente definirán las reglas del juego.
- Meta del juego: será el elemento con el que tendrá que colisionar el personaje principal al final de cada nivel para superar el mismo.
- Reglas del juego: son elementos configurables como el tipo de movimiento, de colisión, la gravedad, condiciones de fin de juego (para ganar y perder), etc.
- Niveles del juego: se podrán colocar los elementos dentro del escenario que poseerá cada nivel.

Del editor se obtendrán los siguientes elementos:

- Una carpeta imágenes, que contendrá todos los recursos gráficos utilizados en el diseño del videojuego.
- Una carpeta sonidos, donde se almacenan los recursos de audio escogidos por el usuario para ser utilizados en el juego.
- Una carpeta niveles, que contendrá los niveles diseñados.
- Un fichero formato XML que incluirá todos los elementos definidos correspondientes del videojuego.

La herramienta se encargará, mediante un analizador, de asignar estos elementos obtenidos a la plantilla existente, y así obtener el proyecto de *Eclipse*, que puede ser ejecutado con el simulador de Android o bien en un dispositivo móvil.

2.2 Objetivos del Proyecto

El sistema desarrollado es una aplicación de escritorio que permite al usuario la creación de sus propios videojuegos para un dispositivo móvil. Y los principales objetivos que se pretende lograr con ello son:

- Definición de una instancia del DSL para juegos de acción. Se pretende crear un lenguaje de alto nivel definido en XML que describa las características de un videojuego en base a características predefinidas.
- Diseñar los diferentes menús que contendrá el juego. Se podrá configurar el aspecto de dichos menús.
- Elegir los personajes y elementos del futuro videojuego. El usuario podrá escoger el personaje principal de su juego, así como la cantidad o tipos de enemigos que interactuarán con dicho personaje principal. También podrá elegir la meta que estará situada al final de cada nivel.

- Escoger la apariencia y comportamiento de los elementos. Se podrán elegir las imágenes de los elementos y configurar las propiedades asignadas a cada elemento, que determinarán el comportamiento de dicho elemento en el videojuego.
- Guardar los cambios y avances realizados.
- Cargar el estado del desarrollo, así como un juego ya creado.
- Obtener una App para instalar en el dispositivo móvil y comenzar a jugar.
- Definición de una instancia del DSL que pueda ser aprovechado para generar juegos en diferentes plataformas. Aunque este trabajo fin de máster se enfoque en dispositivos Android, la instancia del DSL generada permitirá su implementación en generadores de juegos para otras plataformas.

2.3 Estudio de la Situación Actual

En el estado actual de las aplicaciones informáticas, la creación de videojuegos supone una tarea compleja, que requiere, en el caso de los comerciales, de la participación de un amplio equipo de profesionales especializados y del soporte de una serie de medios económicos y técnicos, y en el caso de juegos desarrollados individualmente por un usuario por medio de programas específicos, de unos conocimientos previos, ya sean básicos o avanzados, de programación.

Así pues, pese al gran desarrollo de esta industria del ocio, su creciente competitividad, y a la aparición reciente de los Smartphones y sus varios software para diseñar aplicaciones y juegos para móviles con sistema operativo Android, no ha habido un interés en el sector en generar herramientas que permitan al usuario inexperto, sin formación específica, crear este tipo de aplicaciones, pues las existentes acaban demandando en algún momento unos determinados conocimientos técnicos, como en los ejemplos de las herramientas *GameSalad, GameMaker Studio, App Inventor* o *Stencyl* que se utilizarán como comparativa con el presente proyecto.

Por este motivo el proyecto Gade4all se ha planteado avanzar en ese campo apenas explorado, y sin embargo demandado por una sociedad en la que el acceso a las nuevas tecnologías se ha democratizado y popularizado hasta el punto de que el usuario demanda herramientas para construir sus propias aplicaciones, principalmente de ocio.

En este sentido, por tanto, nuestro proyecto se presenta como una novedad, ya que la principal diferencia de la herramienta desarrollada en él reside en que está destinada a cualquier usuario, tenga o no conocimientos de programación o experiencia previa en el diseño de videojuegos. A través de una interfaz de usuario sencilla podrá construir su propio videojuego en muy poco tiempo, situando los elementos en la pantalla y eligiendo los eventos que puedan tener sin necesidad de escribir una sola línea de código.

Además ninguna de las herramientas existentes en el mercado permite la creación de un videojuego a partir de una instancia del DSL, generándolo desde la interfaz de usuario.

2.3.1 Evaluación de alternativas

En este apartado se analizan una serie de herramientas que en la actualidad son las más usadas para la creación de juegos para Android, señalando las similitudes y diferencias que presentan respecto a este proyecto.

Para ello se muestra, en primer lugar, una tabla comparativa con las diferentes herramientas evaluadas y las características que se han analizado de las mismas. A continuación se incluye una explicación detallada del análisis realizado de cada herramienta.

	GameSalad Windows PC	**GameMaker Studio**	**App Inventor**	**Stencyl**	**Gade4all**
Conocimientos previos de programación	Si	Si	Si	Si	No
Editor gráfico de menús	No	No	Si	No	Si
Editor gráfico de personajes y niveles	Si	Si	Si	Si	Si
Definición de sonidos para eventos	Si	Si	Si	Si	Si
Generación de juegos para Android	Si	Si	Si	Si	Si
Generación de juego completo, no solo archivo ejecutable.	No	No	No	No	Si
Salvar estado del juego	Si	Si	Si	Si	Si
Cargar juego ya creado	Si	Si	Si	Si	Si
Editar solución fuera de la herramienta	No	No	No	No	Si
Añadir comportamientos nuevos	Si	Si	Si	Si	No
Modificar niveles desde fuera del editor	No	No	No	No	Si
Visualizar funcionamiento del nivel creado	Si	Si	Si	Si	No

Ilustración 2.1. Comparativa herramientas

2.3.1.1 GameSalad

GameSalad es una herramienta que permite desarrollar videojuegos y exportarlos para su distribución en diferentes dispositivos y entornos: Android, HTML5, iPhone, iPad y Mac.

Ilustración 2.1. Logo GameSalad

Esta herramienta posee una versión gratuita para su evaluación. Por otra parte, se encuentra la versión comercial que finalmente proporciona la distribución de juegos para todas las plataformas soportadas.

La versión estable de GameSalad Creator está disponible para su ejecución en entornos Mac y existe una versión Beta para sistemas operativos Windows. Para la evaluación de esta herramienta se utilizó la versión Beta de escritorio para Windows.

Los juegos creados con esta herramienta pueden ser publicados directamente a los canales de distribución de las diferentes plataformas, excepto en el caso de aplicaciones para dispositivos con sistema operativo iOS, que requiere disponer de una cuenta del programa de desarrolladores de iOS.

La creación de videojuegos a través de GameSalad Creator se realiza en base a escenas y actores, a los cuales se les define una serie de atributos para establecer su comportamiento, apariencia física, sonidos y reglas.

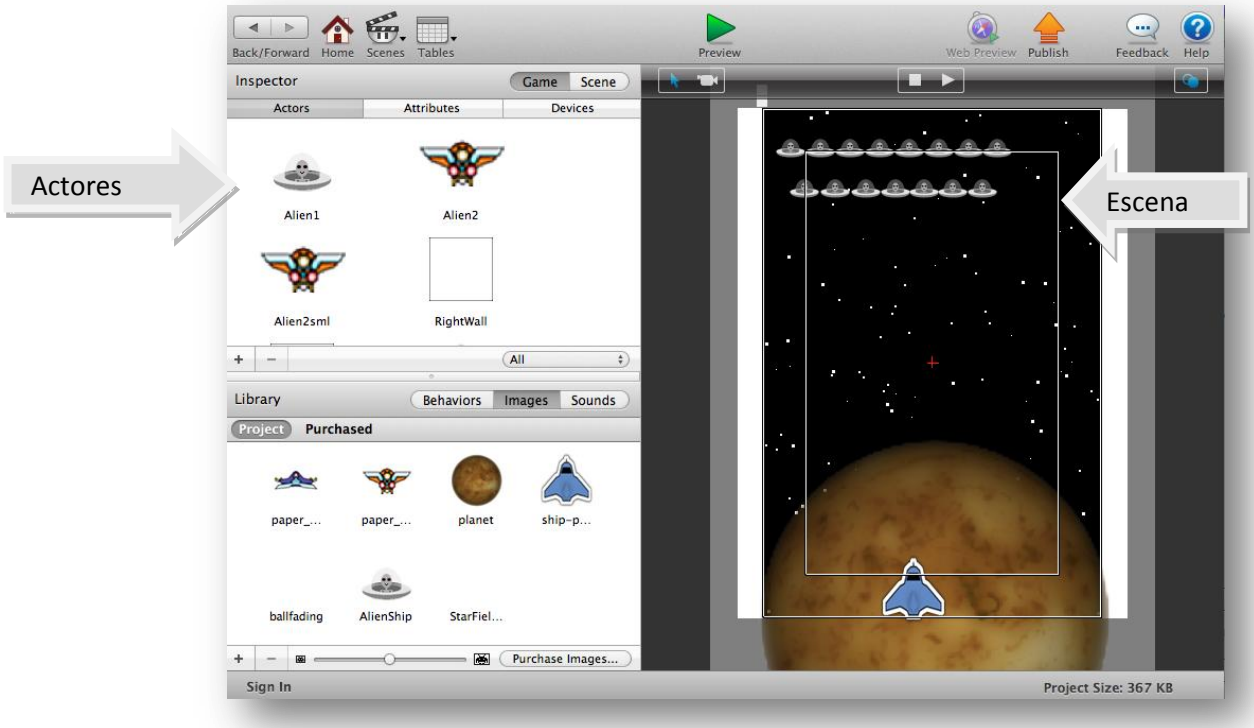


Ilustración 2.2 . GameSalad, actores y escenas

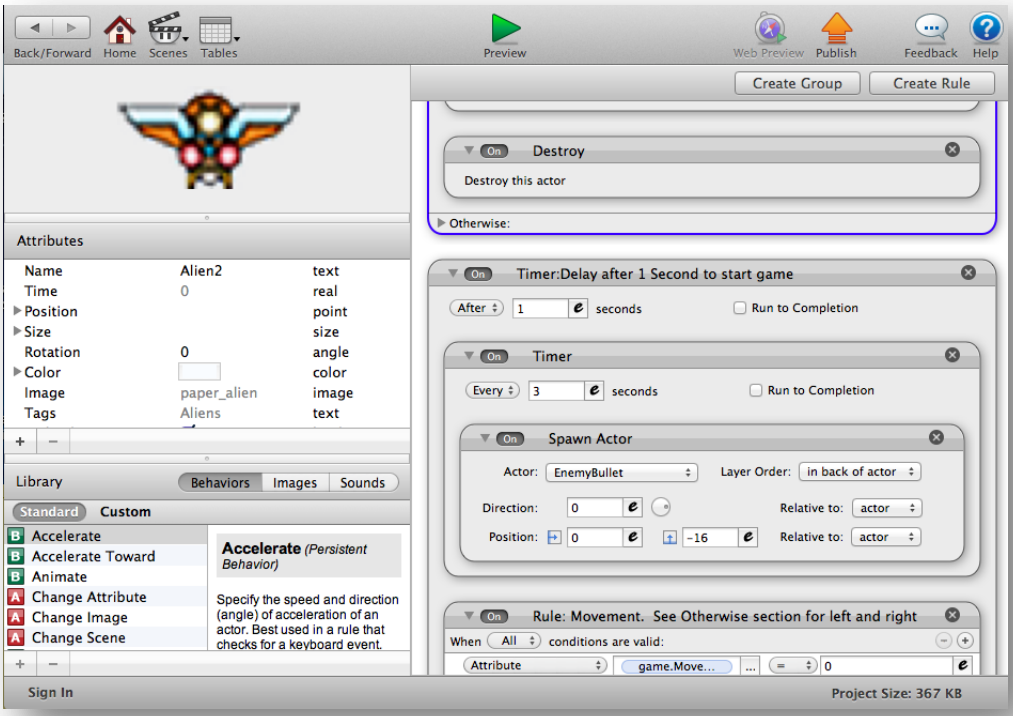


Ilustración 2.3.GameSalad, actores

En esta imagen se muestra el panel de control de los actores con algunos elementos de configuración predefinidos.

Una vez definidos los actores se puede configurar las reglas. La imagen 2.4 muestra un ejemplo de las reglas configurables para un actor de juego. La herramienta cuenta con reglas predefinidas que cubren muchas de las características comunes de un juego, además existe la posibilidad de crear nuevas reglas y modificar las existentes.

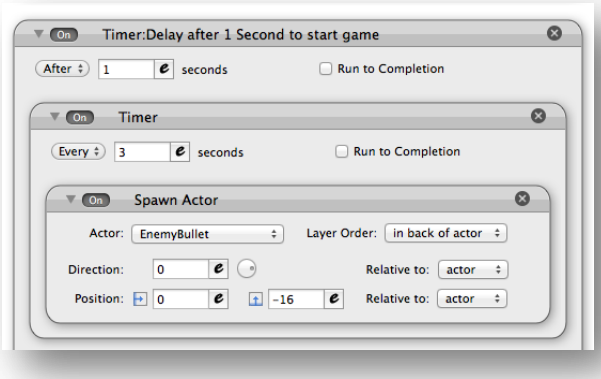


Ilustración 2.4. GameSalad, comportamiento actores

Finalmente se procede a la creación de escenas o niveles del juego, para esta tarea se presenta un panel de control bastante intuitivo que permite visualizar los actores disponibles y arrastrarlos directamente sobre el área destinada a la escena. La imagen 2.5 muestra un ejemplo de creación de escenas.

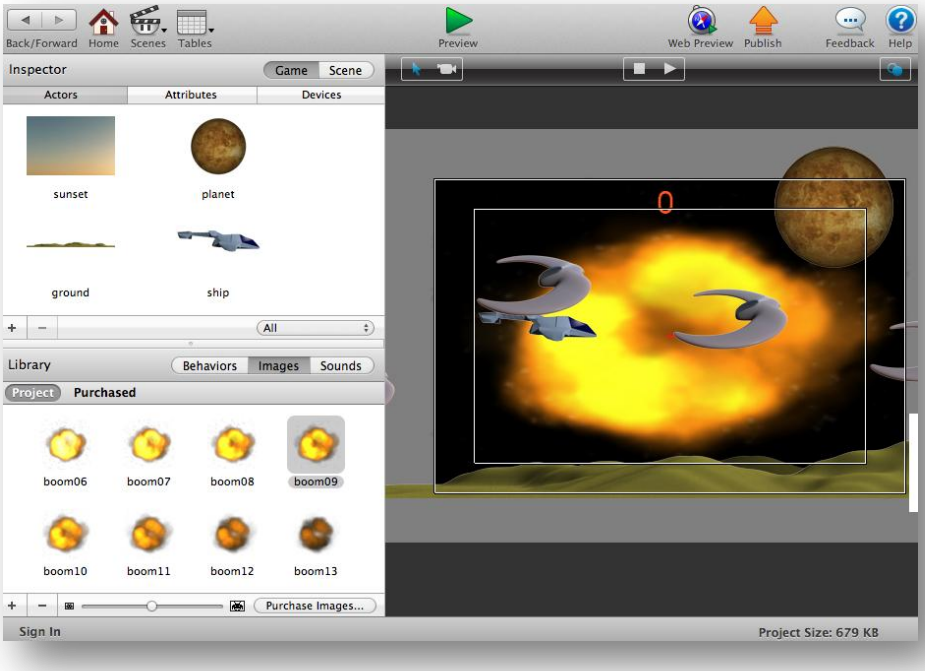


Ilustración 2.5. GameSalad, visualización nivel

GameSalad posee un emulador para ejecutar la vista previa del juego en tiempo real, esta característica proporciona muchas ventajas durante la creación de los juegos.

2.3.1.1.1 Similitudes

Se han identificado las siguientes similitudes entre GameSalad y la herramienta desarrollada en este proyecto:

- Capacidad de creación de actores y personajes de un juego.
- Definición de sonidos para eventos.
- Editor gráfico para la creación y edición de niveles.
- Generación de juegos para dispositivos Android.
- Almacenamiento de las características de un juego para su posterior modificación.

2.3.1.1.2 Diferencias

En esta sección se enumeran los elementos que diferencian la herramienta desarrollada para este trabajo fin de máster de GameSalad:

- La primera diferencia radica en que utilizando Gade4all se puede configurar los diferentes menús del juego, mientras que con GameSalad solamente se puede configurar la pantalla de juego y sus elementos.
- El resultado final de un juego creado con GameSalad puede ser editado solamente desde la misma herramienta, no existe manera de utilizar los entornos nativos de desarrollo. Sin embargo, Gade4all proporciona los códigos fuente de los juegos que pueden ser modificados directamente desde el entorno de desarrollo original, Eclipse.
- GameSalad permite incluir nuevas reglas de comportamiento desde el mismo editor, lo que en Gade4all no está contemplado, pero existe la posibilidad de modificar directamente el proyecto generado. Para configurar estas nuevas reglas o modificar las existentes es necesario tener conocimientos de programación.
- En Gade4all, además de contar un editor gráfico, se puede acceder al fichero XML que describe la instancia del DSL completa.
- Los juegos desarrollados con Gade4all cuentan con un cargador de niveles basado en XML, esto permite que los niveles puedan ser añadidos o modificados sin necesidad del editor.
- GameSalad cuenta con un emulador de juegos que permite ver el producto resultante incluso antes de la finalización del diseño del mismo. Esta funcionalidad no ha sido implementada en Gade4all.

2.3.1.2 GameMaker Studio

GameMaker Studio es una herramienta para crear videojuegos, dirigida tanto a principiantes como a experimentados profesionales en el desarrollo de videojuegos. Está basada en un lenguaje interpretado propio, basado en scripts destinado a los usuarios más expertos llamado GML (Game Maker Language). Permite desplegar para iOS, Android, PC y HTML5, pero con la versión gratuita solamente es posible desplegar para ordenador, Mac o Windows.

Ilustración 2.6. Logo GameMaker

El programa se rige por la filosofía de *Drag and Drop* (arrastrar y soltar) a la hora de realizar las interacciones y relaciones de objetos, de forma que en un principio no se tiene que escribir ni una sola línea de código. Un ejemplo práctico: si se desea que un objeto se desplace en cierta dirección cuando se pulse una tecla concreta se creará mediante una interfaz dicha interacción. Se cogerá un botón que dirá "si se pulsa X" y se le asociará "Muévete hacia la derecha".

Como se puede observar en la imagen siguiente, esta herramienta está divida en dos zonas, una paleta en la parte izquierda y la zona de configuración de los elementos y de los niveles, situada en la parte derecha.

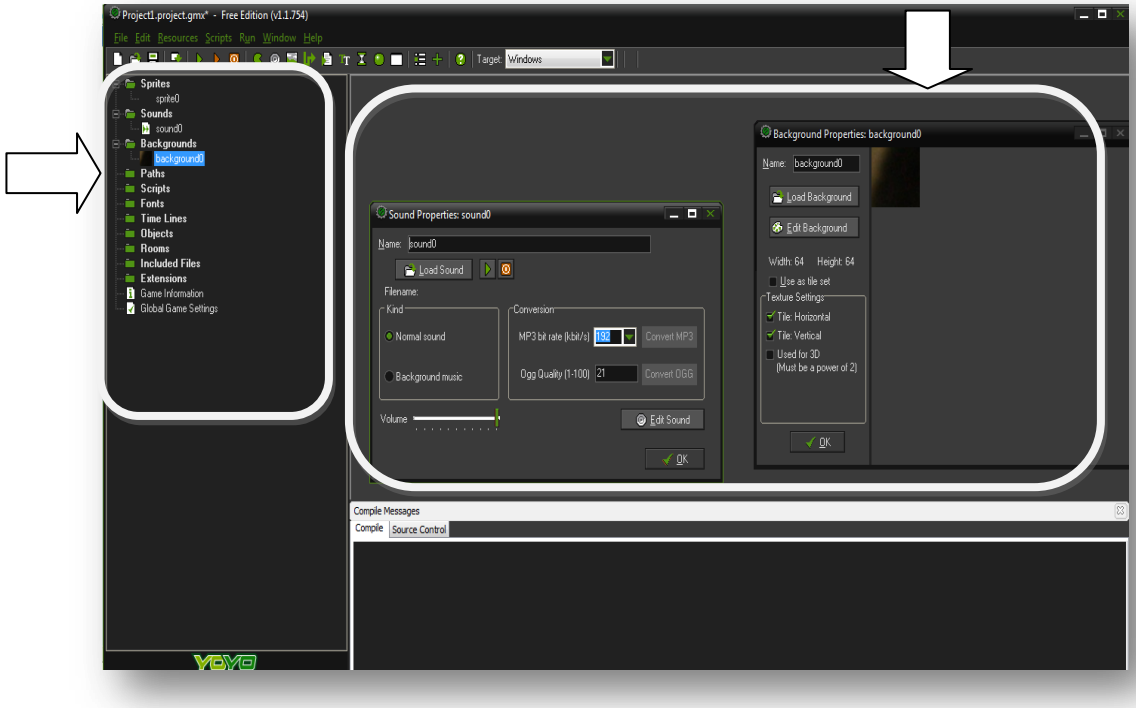


Ilustración 2.7. Herramienta GameMaker

En el editor de niveles se muestra la escena donde se añadirán los diferentes elementos de cada nivel, y la zona que está fuera de la pantalla en la que también se pueden colocar elementos. El tamaño de esta región fuera de pantalla se puede modificar. Los elementos que ya han sido creados, como fondos, sprites, tiles, aparecen al lado de la escena para arrastrarlos directamente.

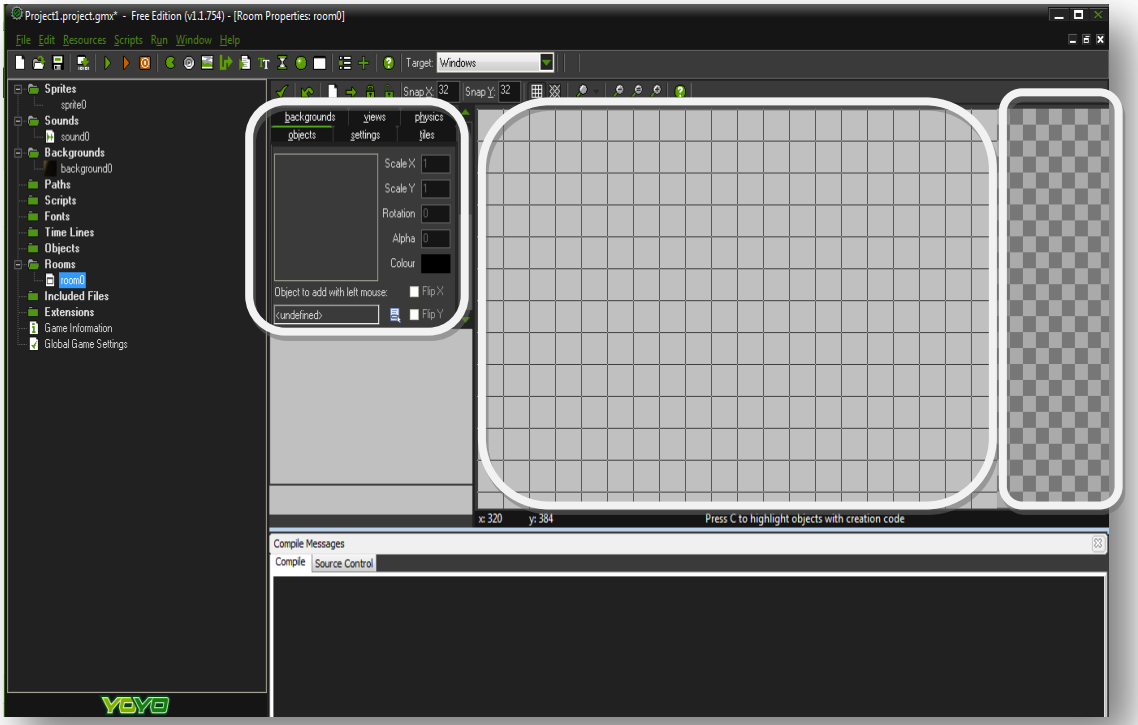


Ilustración 2.8. GameMaker, escenario

A cada elemento se le puede asignar un evento, como una colisión con un enemigo, y a cada evento se le pueden encadenar varias acciones, como por ejemplo que cuando colisione salte o varíe su velocidad. Dispone de una biblioteca de eventos y de acciones. En caso de crear alguno nuevo o de querer modificar alguno de los existentes, sería necesario tener conocimientos de programación en lenguaje propio GML.

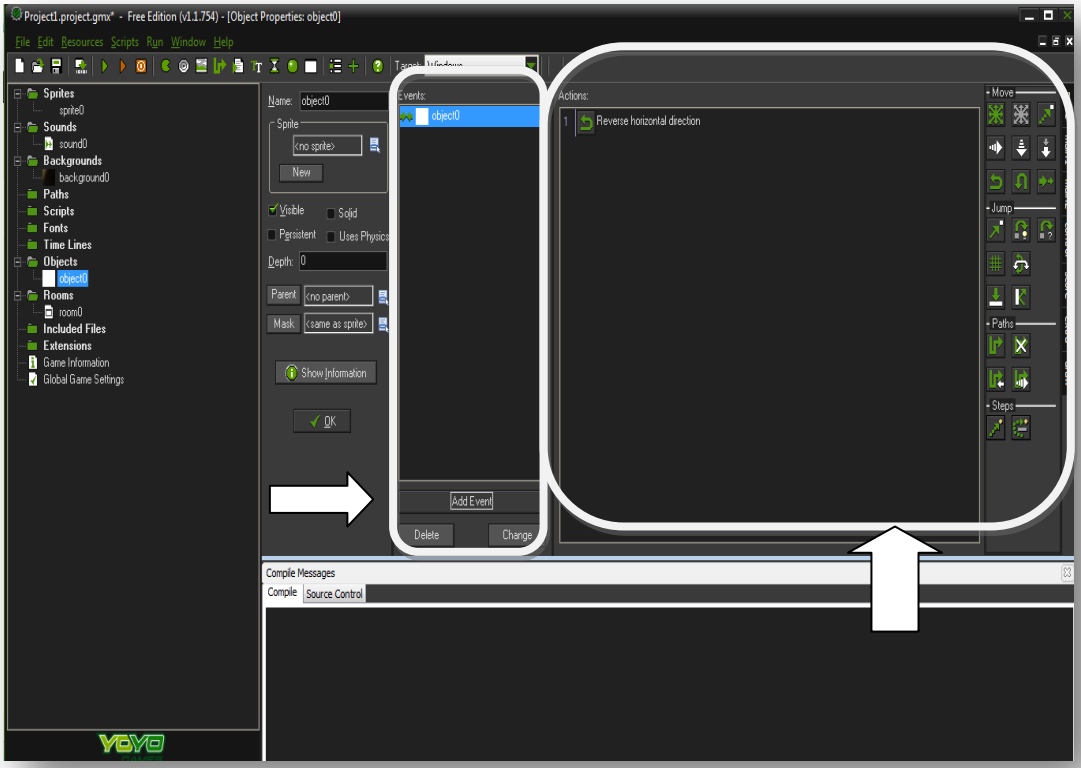


Ilustración 2.9. GameMaker, comportamientos elementos

A continuación se muestra una imagen de cómo sería una escena de un juego creada con esta herramienta. Se ha añadido un fondo, unos tiles y unos sprites, a los cuales se les asignó alguno de los eventos existentes en la biblioteca. También se puede ver una miniatura del nivel mientras se está diseñando este.

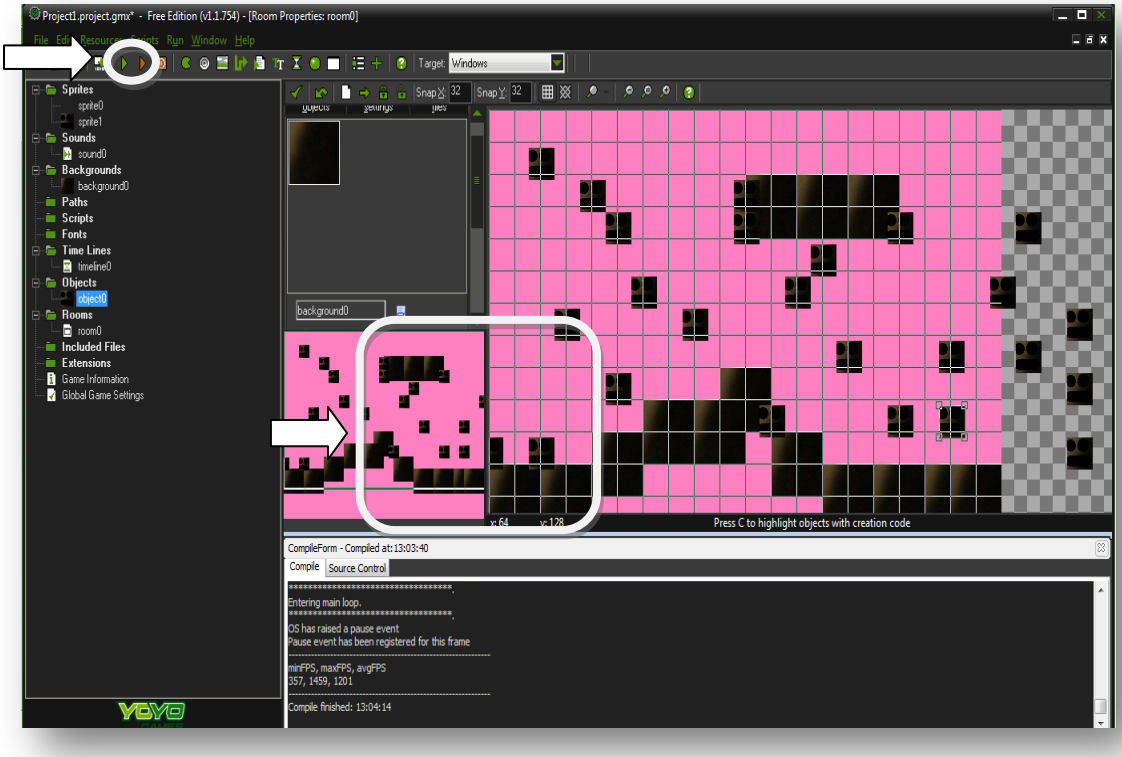


Ilustración 2.10. GameMaker, creación nivel

Para poder probar el funcionamiento del nivel creado, bastaría con darle al botón Play y se abriría una ventana emergente en la que se mostraría la ejecución del nivel.

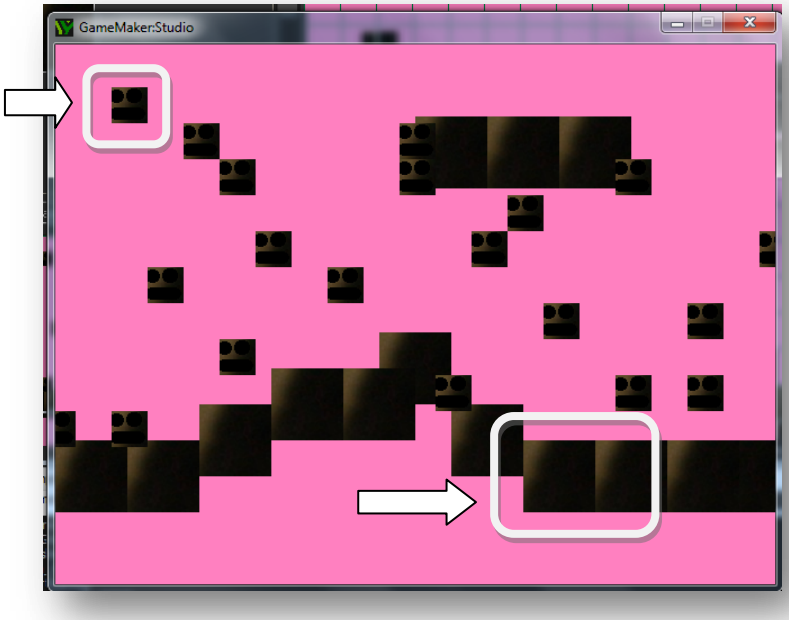


Ilustración 2.11. GameMaker, nivel creado

2.3.1.2.1 Similitudes

A continuación se describen algunas de las similitudes identificadas entre GameMaker Studio y la herramienta desarrollada en este proyecto:

- Capacidad de creación de actores y personajes de un juego.
- Definición de sonidos para eventos.
- Editor gráfico para la creación y edición de niveles.
- Generación de juegos para dispositivos Android.
- Almacenamiento de las características de un juego para su posterior modificación.

2.3.1.2.2 Diferencias

En esta sección se enumeran los elementos que diferencian GameMaker Studio de la herramienta desarrollada para este trabajo fin de máster:

- La primera diferencia radica en que utilizando Gade4all se puede configurar los diferentes menús del juego, mientras que con GameMaker Studio solamente se puede configurar la pantalla de juego y sus elementos.
- El resultado final de un juego creado con GameMaker Studio puede ser editado solamente desde la misma herramienta, no existe manera de utilizar los entornos nativos de desarrollo. Sin embargo, Gade4all proporciona los códigos fuente de los juegos que pueden ser modificados directamente desde el entorno de desarrollo original, Eclipse.
- GameMaker Studio permite incluir nuevas reglas de comportamiento desde el mismo editor, lo que en Gade4all no está contemplado, pero existe la posibilidad de modificar directamente el proyecto generado.
- GameMaker no solo requiere conocimientos de programación, sino que además tiene la dificultad añadida de que es necesario aprender su propio lenguaje GML.
- En Gade4all, además de contar un editor gráfico, se puede acceder al fichero XML que describe la instancia del DSL completa.
- Los juegos desarrollados con Gade4all cuentan con un cargador de niveles basado en XML, esto permite que los niveles puedan ser añadidos o modificados sin necesidad del editor.
- GameMaker Studio cuenta con un emulador de juegos que permite ver el producto resultante incluso antes de la finalización del diseño del mismo. Esta funcionalidad no ha sido implementada en Gade4all.

2.3.1.3 App Inventor Beta

App Inventor es un servicio web de Google Labs, ahora propiedad de **MIT** (Massachussets Institute of Technology), que permite crear sencillas aplicaciones para Android. Con esta herramienta no se programa código pero si hace falta tener algún conocimiento de programación para configurar los eventos que se producen al pulsar sobre los botones o configurar los comportamientos de los personajes del juego. Permite crear las aplicaciones eligiendo un diseño y seleccionando las acciones que se van a realizar en el juego.

Para crear una aplicación con *App Inventor* hay que realizar dos pasos:

- El diseño de la aplicación, en la que se seleccionan los componentes para su aplicación.
- El editor de bloques, donde se escogen los bloques necesarios según la aplicación que se quiera hacer.

La aplicación aparecerá paso a paso en la pantalla del teléfono a medida que se añadan piezas a la misma, para que se pueda probar el trabajo. Cuando se haya terminado, se puede empaquetar la aplicación y producir de manera independiente para instalar.

Si no se dispone de un teléfono Android, se pueden construir las aplicaciones utilizando el emulador de Android, con el que el software se ejecuta en el ordenador, comportándose como en un dispositivo móvil.

El entorno de desarrollo de *App Inventor* aplicación es compatible con Mac OS X, GNU / Linux y sistemas operativos de Windows, y varios modelos de teléfonos Android populares.

Antes de poder utilizar *App Inventor,* es necesario configurar el ordenador e instalar los archivos de instalación de *App Inventor* en el equipo.

Ilustración 2.12. Logo AppInvertor

La creación de juegos con *App Invertor Beta* se basa en la configuración de cada uno de los elementos que van a tener todas las pantallas del juego y de los eventos que se desencadenarán a lo largo del juego.

Se podrán crear varias pantallas, dentro de cada una de las cuáles se configurarán todos los elementos que se desee que contengan. De esta forma, además de poder diseñar los niveles del juego se podrán diseñar también los diferentes menús. Además también se podrán crear aplicaciones que no sean juegos.

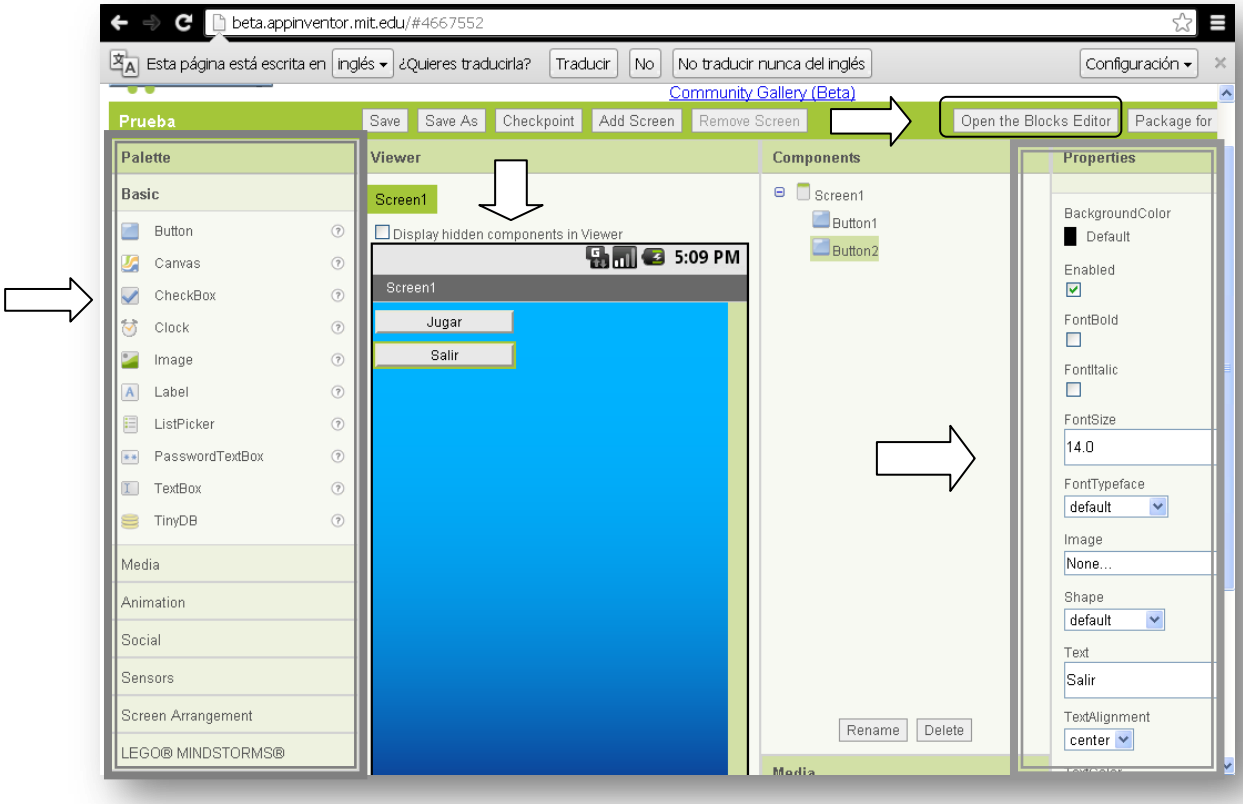


Ilustración 2.13. Herramienta AppInventor

En la parte izquierda, como se puede ver en la figura anterior, se encuentra la paleta que contiene todos los elementos que se pueden arrastrar a la pantalla. En este caso se creó un ejemplo para la pantalla principal del juego, con dos botones y un fondo. Y en la parte derecha se muestra una columna con las propiedades del elemento seleccionado, en este caso el botón Salir.

Para dar comportamiento a esos dos botones, se tiene que utilizar el editor de bloques, en el que se programará el evento que se desencadenará al pulsar el botón.

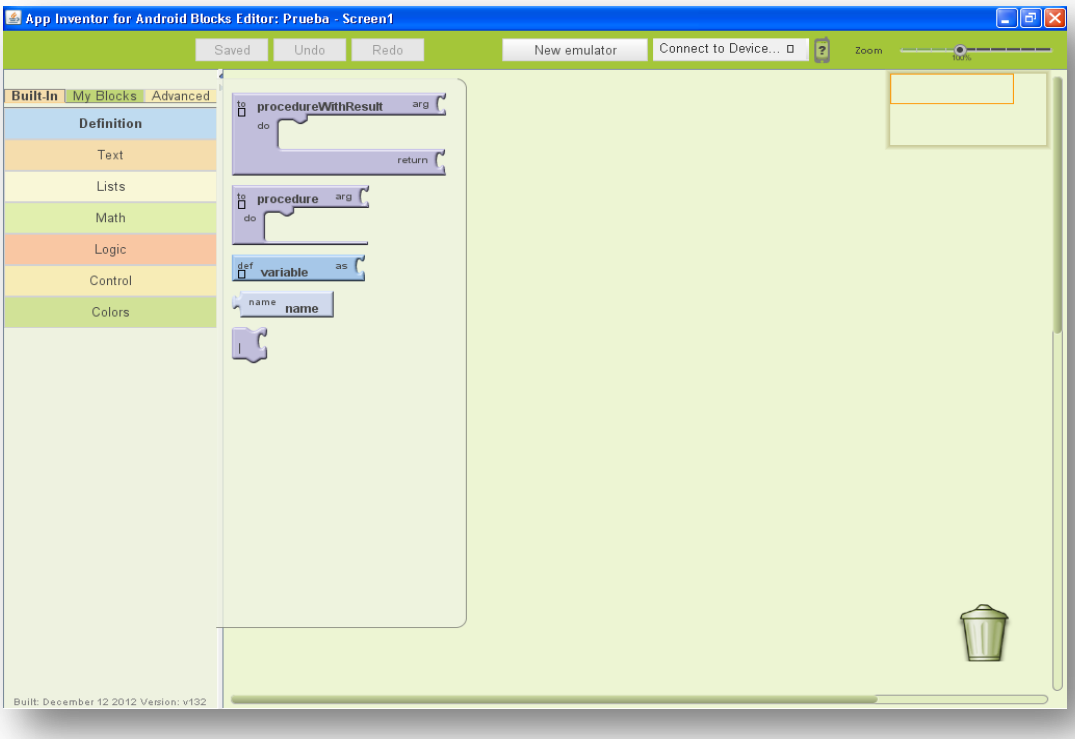


Ilustración 2.14. AppInventor, editor bloques

Para poder darle comportamiento al botón, o a otro elemento del juego, es necesario tener conocimientos de programación, ya que los bloques de comportamiento contienen elementos de lógica algorítmica: condicionales, bucles, entre otros, del lenguaje de programación Java.

Para diseñar un nivel de un juego la herramienta ofrece dos tipos de elementos "Ball" y "Animation Sprite". De cada uno de ellos se puede configurar su imagen, tamaño, posición, velocidad y otras propiedades.

Como se muestra en la imagen siguiente, solamente se dispone de una pantalla del tamaño de un móvil para diseñar los niveles, no se puede ampliar el tamaño, por lo que a la hora de diseñar los niveles de un juego no se podrían incluir elementos que estuvieran fuera de la pantalla y que con una determinada velocidad fueran avanzando y aparecieran en la pantalla del juego transcurrido un cierto tiempo.

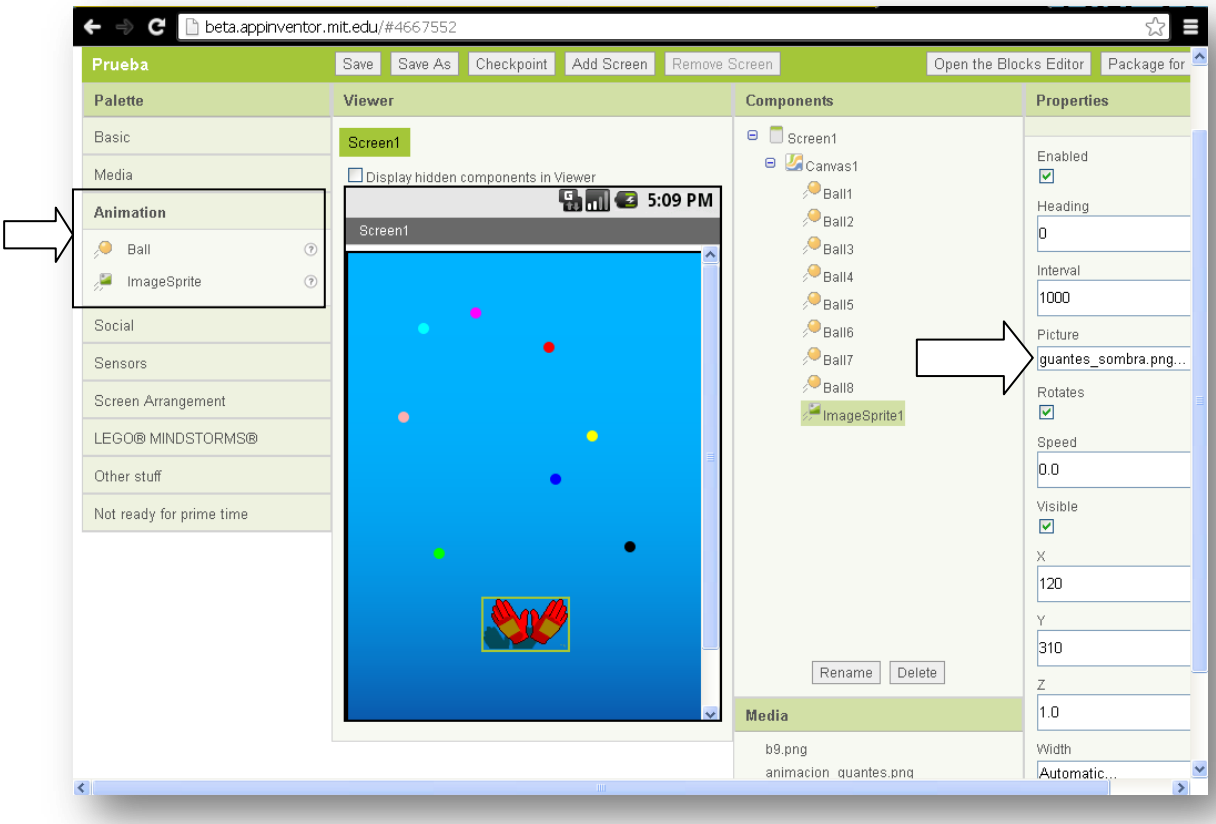


Ilustración 2.15. AppInventor, creación pantalla

Una vez diseñado el juego y programados los comportamientos de cada uno de los elementos de cada pantalla, se empaqueta la aplicación para obtener la aplicación lista para instalar en el móvil. Esta se podrá instalar directamente en el móvil si este se encuentra conectado al ordenador o bien se podrá almacenar en el PC y posteriormente instalar en el dispositivo móvil.

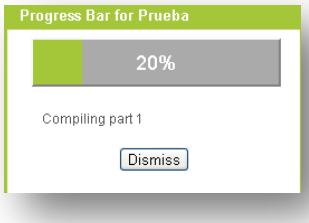


Ilustración 2.16. AppInventor, barra estado

2.3.1.3.1 Similitudes

A continuación se describen algunas de las similitudes identificadas entre *App Inventor* y la herramienta desarrollada en este proyecto:

- Capacidad de creación de actores y personajes de un juego.
- Editor gráfico para la creación y edición de los menús del juego.
- Editor gráfico para la creación y edición de niveles.
- Generación de juegos para dispositivos Android.
- Almacenamiento de las características de un juego para su posterior modificación.

2.3.1.3.2 Diferencias

En esta sección se enumeran los elementos que diferencian la herramienta desarrollada para este trabajo fin de máster de *App Inventor:*

- El resultado final de un juego creado con *App Inventor* puede ser editado solamente desde la misma herramienta, no existe manera de utilizar los entornos nativos de desarrollo. Sin embargo, Gade4all proporciona los códigos fuente de los juegos que pueden ser modificados directamente desde el entorno de desarrollo original, Eclipse.
- *App Inventor* permite incluir nuevas reglas de comportamiento desde el editor bloques, lo que en Gade4all no está contemplado, pero existe la posibilidad de modificar directamente el proyecto generado.
- En Gade4all, además de contar un editor gráfico, se puede acceder al fichero XML que describe la instancia del DSL completa.
- Los juegos desarrollados con Gade4all cuentan con un cargador de niveles basado en XML, esto permite que los niveles puedan ser añadidos o modificados sin necesidad del editor.
- *App Inventor* no cuenta con un emulador de juegos, pero se puede desplegar la aplicación directamente en el móvil o en el emulador de Android que viene incluido en la herramienta, y así se puede visualizar y probar el juego mientras se esté diseñando. Esta funcionalidad no ha sido implementada en Gade4all.
- Para la utilización de esta herramienta de creación de juegos no se programa código directamente pero sí son necesarios unos conocimientos básicos de programación, ya que para configurar los bloques asociados a cada elemento es necesario conocer elementos de lógica algorítmica: condicionales, bucles, entre otros, del lenguaje de programación Java.

2.3.1.4 Stencyl

Stencyl es una herramienta para la creación de videojuegos desarrollada por StencylWorks. Proporciona a los diseñadores de videojuegos un editor gráfico que permite la creación de juegos y posteriormente su exportación a formatos iPhone, iPad, Mac, Exe, Flash y Android.

Ilustración 2.17. Logo Stencyl

Los juegos desarrollados pueden ser publicados en Google Play de Android. Esta herramienta, en el mercado desde mediados de 2011, ha tenido bastante éxito entre los desarrolladores de videojuegos.

Ofrece dos tipos de licencias, una versión gratuita que permite conocer el funcionamiento completo de la herramienta y realizar exportaciones a Flash, incluyendo publicidad de la herramienta, y por otra parte está la versión profesional que permite crear juegos para Android, iOS, Flash y ejecutables para Windows y Linux sin incluir ninguna publicidad.

La creación de juegos en *Stencyl* se basa en la definición de escenas y actores. Las escenas se corresponden a cada uno de los niveles del juego, sobre las cuales se añaden los diferentes actores. Un actor será cualquier elemento que se muestre en un nivel, desde los elementos gráficos sin funcionalidad ni lógica añadida, hasta los propios personajes del juego, a los que se les establecen sus propias reglas y comportamientos.

En la siguiente figura se observa una escena ya construida que forma parte de un juego de las plantillas de juegos de Stencyl. Se cuenta con el panel de actores que han sido creados específicamente para este juego y los cuales pueden ser arrastrados para posicionarlos en el lugar deseado. La mayor parte de los actores poseen reglas de lógica y comportamiento relacionadas.

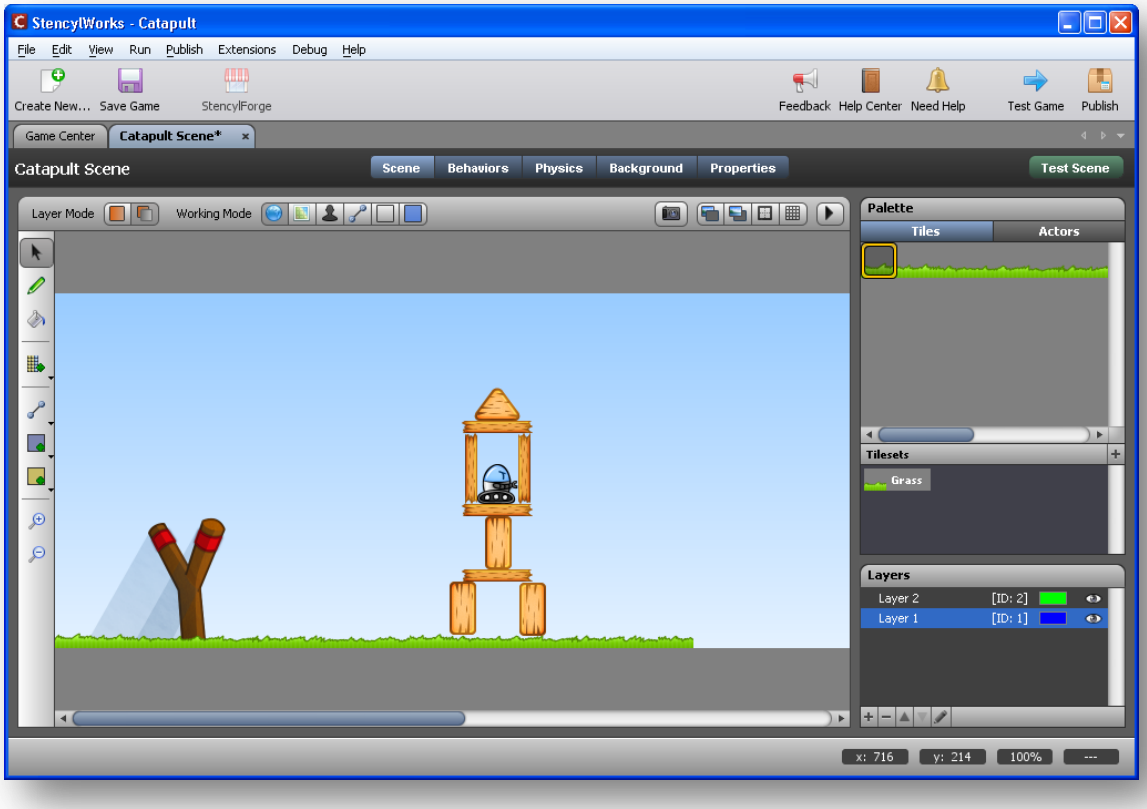


Ilustración 2.18. Stencyl, nivel ejemplo

Entre las propiedades configurables del personaje, se pueden enumerar algunas de las más representativas.

- Contenido gráfico y animaciones. Permite definir las imágenes y animaciones que representarán al actor dentro del juego.
- Aplicación de leyes físicas. Determinan si el objeto es afectado por la gravedad, si pueden ser detectados en colisiones e incluso los niveles de resistencia que pueden presentar en caso de colisión o roces con otros actores.
- Posición y movimiento. Determina las posiciones y velocidad/aceleración inicial de los actores.
- Eventos. Define los eventos que afectan de alguna forma al personaje, de esta forma, se puede decidir si al interactuar con el actor se debe destruir, iniciar una animación o modificar alguna de sus propiedades.

El entorno de desarrollo de *Stencyl* es suficientemente intuitivo para personas que tengan experiencia con programas de diseño, y aunque cuenta con algunas plantillas predefinidas, siempre es necesario poseer conocimientos de programación y lógica algorítmica, ya que muchas de las modificaciones del juego requieren crear diagramas de flujo de algoritmos.

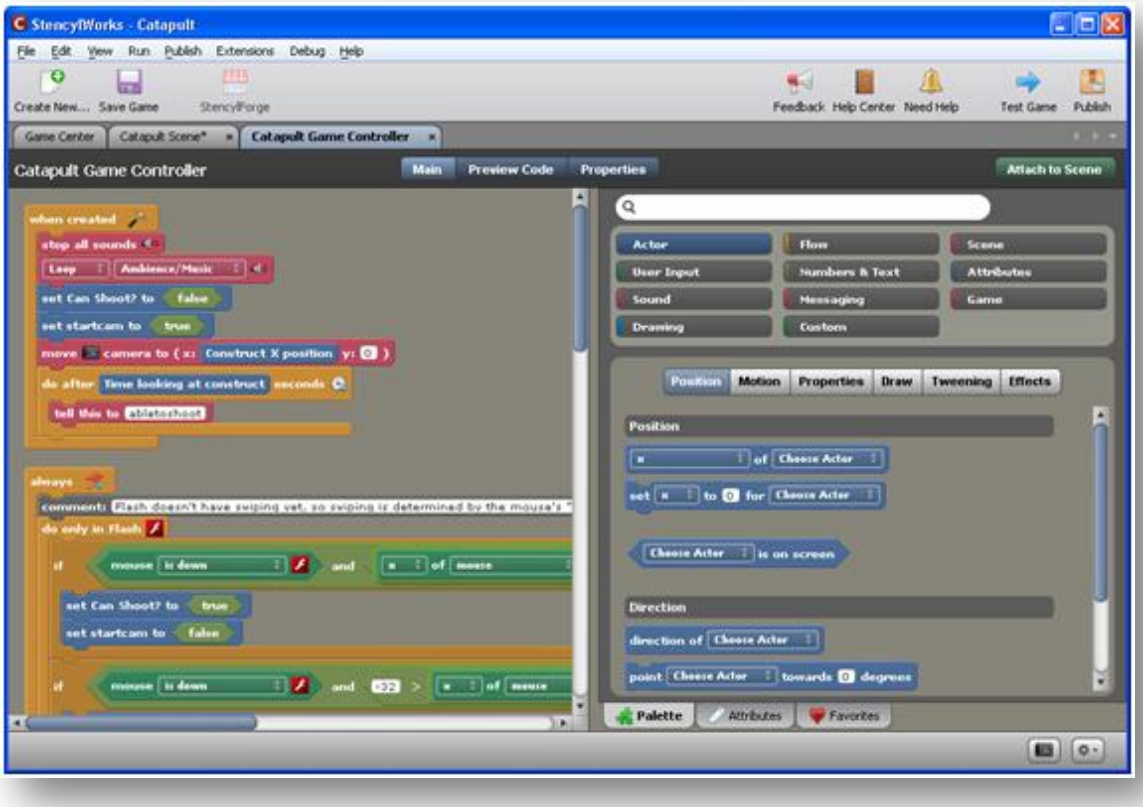


Ilustración 2.19. Stencyl, programación comportamientos

En esta imagen se muestra un ejemplo de código de *Stencyl* del comportamiento de una catapulta y se puede observar claramente que cuenta con todos los elementos utilizados en la lógica algorítmica: condicionales, bucles, entre otros. Esta utilidad es de gran ayuda para agregar funcionalidad al juego, sin embargo, limita de cierta forma el uso de la misma a personas que no posean conocimientos específicos.

2.3.1.4.1 Similitudes

La herramienta *Stencyl* posee algunas semejanzas con la herramienta desarrollada para este trabajo fin de máster. A continuación se enumeran las más representativas:

- Editor gráfico que permite crear diferentes personajes y diseñar su apariencia.
- Editor gráfico para la creación y edición de niveles.
- Definición de sonidos ante diferentes eventos.
- Generación de juegos para dispositivos Android.
- Almacenamiento de las características de un juego para su posterior modificación.

2.3.1.4.2 Diferencias

Las diferencias más significativas entre *Stencyl* y la herramienta desarrollada para este proyecto son las siguientes:

- Posiblemente la mayor ventaja de Gade4All radica en el resultado final del juego, ya que *Stencyl* solamente genera un fichero ejecutable que depende del tipo de exportación que se esté realizando, el cual se considera la implementación definitiva y no puede ser editado, mientras que Gade4All genera un proyecto completamente editable en Eclipse, lo que implica que si el diseñador de videojuegos desea modificar alguna de las funcionalidades no definidas en la instancia del DSL lo puede hacer directamente sobre el proyecto generado.
- La herramienta de creación de juegos de este proyecto no requiere escribir ni una sola línea de código, todo el comportamiento y reglas de los juegos están predefinidas, lo que implica que cualquier persona, incluso sin conocimientos de programación, podrá crear juegos sin problemas. Si en un dado caso se necesitara modificar a nivel de código se puede acudir directamente al proyecto generado, aunque esto requerirá conocimientos de desarrollo en Android.
- En Gade4All, además de contar con un editor gráfico, se podrá acceder al fichero XML que describe la instancia del DSL completa. Stencyl por su parte, maneja toda la configuración de forma que solamente pueda ser manejada y visualizada desde el editor.
- Los juegos desarrollados en Gade4All contarán con un módulo software cargador de niveles basado en XML, esto permite que los niveles puedan ser añadidos o modificados sin necesidad de utilizar editor, aunque se debe mencionar que dicha tarea requiere el conocimiento de la estructura de los ficheros XML que definen los niveles.
- Los juegos desarrollados con Stencyl carecen de una sección de configuración, pantalla de bienvenida y menú de niveles, al ejecutar el juego desarrollado se accede directamente al primer nivel y cada vez que se supera se carga automáticamente el siguiente nivel, no otorgando por defecto opciones de regresar o de navegar entre los diferentes niveles. Gade4All por su parte, cuenta con pantallas por defecto que serán diseñadas desde el editor.

Capítulo 3. Aspectos Teóricos

3.1 Android

Android es un sistema operativo móvil basado en Linux, destinado a una utilización en dispositivos móviles, como teléfonos inteligentes, tabletas, Google TV y otros. Este sistema maneja, generalmente, aplicaciones como Market (Marketing) o su actualización, PlayStore. La estructura de Android está formada por aplicaciones que se ejecutan en un framework Java de aplicaciones orientadas a objetos sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik con compilación en tiempo de ejecución.

Android, a diferencia de otros sistemas operativos para dispositivos móviles como iOS o Windows Phone, se desarrolla de forma abierta, pudiendo accederse a su código fuente y al listado de incidencias donde se pueden ver problemas aún no resueltos e informar de otros nuevos.

Pese al acceso libre al código fuente, no es posible tener siempre la última versión de Android en un determinado móvil, ya que el código para soportar el hardware (controladores) de cada fabricante de terminales móviles normalmente no es público, por lo que faltaría una parte básica del firmware para poder hacerlo funcionar en dicho terminal. Asimismo, las nuevas versiones de Android suelen requerir más recursos, por lo que los modelos más antiguos de móviles quedan descartados por razones de memoria (RAM), velocidad de procesador, etc.

Las aplicaciones se desarrollan habitualmente en el lenguaje Java con Android Software Development Kit (Android SDK).

3.2 Domain Specific Language

DEFINICIÓN. DSL, o "Domain Specific Language" (Lenguaje Específico de Dominio), es un lenguaje de programación dedicado a un problema de un dominio en particular, o una técnica de representación o resolución de problemas específica. [DESETA09].

Los DSL se crean específicamente para resolver problemas de un determinado dominio, con lo cual no están pensados para resolver problemas fuera de este. Son lenguajes con objetivos muy específicos, tanto en diseño como en implementación. Puede ser un lenguaje de diagramación visual así como textual. Son por lo general pequeños, ofreciendo un conjunto limitado de notaciones y abstracciones. Son, por lo general, lenguajes declarativos, por lo que pueden verse como lenguajes de especificación, así como lenguajes de programación. [FURTADO06].

Estos DSL no tienen funciones de bajo nivel (acceso al sistema de ficheros, control de procesos internos, etc.) y además no se compilan a código ejecutable. El uso de DSL trae

consigo una serie de ventajas, como permitir expresar soluciones usando los términos y el nivel de abstracción apropiado para el dominio de un problema concreto, así como una mejora de la calidad, productividad, mantenibilidad, portabilidad y reutilización de las aplicaciones.

Tal como lo describe (Miotto, 2009), los modelos pueden ser claramente expresados utilizando un DSL. En este sentido, el DSL contiene conceptos específicos del dominio que permiten un mejor entendimiento del mismo, obviando así elementos que se alejan de la realidad del dominio.

Una de las características principales del DSL es que permite implementar una definición independiente de la plataforma, la cual una vez finalizada pasará por un proceso de transformación proporcionando como resultado final la construcción de un proyecto de una plataforma específica.

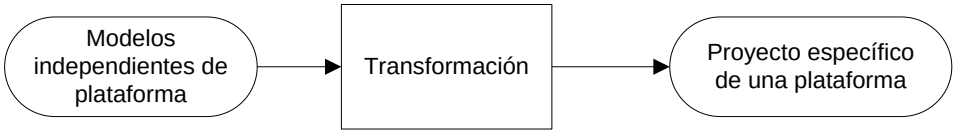


Ilustración 3.1 Transformación de modelos independientes de plataforma

El DSL es crucial para la ingeniería dirigida por modelos, y por lo tanto para este proyecto. La instancia del DSL utilizada para la definición de los videojuegos deberá contener, con un nivel de abstracción lo más alto posible, la descripción de todos los elementos necesarios para la generación del videojuego, así como también el establecimiento de los valores requeridos para asignar comportamiento a los actores. Dicha tarea será realizada por medio del Editor de Videojuegos, por lo tanto, para el usuario final no será necesario conocer los detalles del mismo.

3.3 eXtensible Markup Language

XML, o "eXtensible Markup Language" (Lenguaje Extensible de Marcado), es un metalenguaje extensible de etiquetas desarrollado por el consorcio W3C (World Wide Web Consortium). La palabra metalenguaje indica que XML no supone por sí solo un lenguaje particular, sino una manera de definir lenguajes para diferentes necesidades.

XML es una tecnología sencilla, que se puede complementar con otras para ampliar sus posibilidades. Tiene una gran proyección en la actualidad, pues permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

Tiene un formato abierto, independiente de las plataformas. Permite, además, utilizar vocabularios específicos dentro de una aplicación, además de la auto-descripción de éstos. [SANJUAN09].

3.4 eXtensible Application Markup Language

XAML, o "eXtensible Application Markup Language" (Lenguaje Extensible de Marcado para Aplicaciones), es un lenguaje de marcado para la programación declarativa de aplicaciones, basado en XML, optimizado para describir gráficamente interfaces de usuario visuales ricas desde el punto de vista gráfico. [MSDN07]

3.5 Model Driven Engineering

MDE, o "Model-Driven Engineering" (Ingeniería dirigida por modelos), es una metodología de desarrollo de aplicaciones centrada en la creación de modelos. Estos modelos son las abstracciones que anteriormente se comentaban en los DSL. Como ya se dijo, estas abstracciones van focalizadas en desarrollos de aplicaciones cerradas, es decir, específicas de un problema.

Todo esto se hace con el propósito de incrementar la productividad y la compatibilidad entre sistemas, simplificando la etapa de diseño para facilitar y ampliar la comunicación entre miembros de un equipo.

En este sentido, se considera efectivo un modelado del paradigma para MDE si este modelo tiene sentido desde el punto de vista del usuario y sirve como base para una implementación de un sistema. Los modelos se desarrollan a partir de una comunicación fluida y extensa entre los miembros de un equipo de desarrollo (analistas, diseñadores…). En definitiva, deben permitir el desarrollo de software y de sistemas. [SCHMIDT06].

3.6 Model Driven Architecture

MDA, o "Model Driven Architecture", (Arquitectura dirigida por modelos), ha sido concebido para dar soporte a la ingeniería dirigida a modelos de los sistemas de software. Proporciona un conjunto de guías para estructurar especificaciones expresadas como modelos.

MDA nació con la idea de separar la especificación de la lógica operacional de un sistema, de los detalles que definen cómo el sistema usa las capacidades de la plataforma tecnológica donde es implementado.

Promovido por la OMG (Object Management Group), se propone basar el desarrollo software en modelos especificados utilizando UML ("Unified Modeling Language"), para que, a partir de estos modelos, se realicen transformaciones que generen código y otro modelo, con características de una tecnología particular (o con menor nivel de abstracción). MDA define un framework para procesar y relacionar modelos. [ANACLETO06].

La finalidad del sistema será definida en primer lugar como un modelo independiente de la plataforma (Platform-Independent Model o PIM) a través de un lenguaje específico para el dominio del que se trate. Dado un modelo de definición de la plataforma (Platform

Definition Model o PDM) correspondiente a CORBA, .NET, web, etc., el modelo PIM puede traducirse entonces a uno o más modelos específicos de la plataforma (Platform-specific models o PSM) para la implementación correspondiente, usando diferentes lenguajes específicos del dominio, o lenguajes de propósito general como Java, C#, Python, etc.

3.7 Transformación de modelos

Una transformación de modelo se define como la generación automática de un modelo destino a partir de un modelo origen, de acuerdo con una definición de transformación. Una definición de transformación es un conjunto de reglas de transformación que unidas describen cómo un modelo en el idioma de origen puede transformarse en un modelo en el idioma de destino. Una regla de transformación es una descripción de cómo una o más construcciones en el idioma origen se pueden transformar en una o más construcciones en el idioma destino [KLEPPE03].

3.8 Unified Modeling Language

UML, o "Unified Modeling Language", (Lenguaje Unificado de Modelado), es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. Ofrece un estándar para describir un plano del sistema (modelo), incluyendo aspectos conceptuales, como procesos de negocio y funciones del sistema, además de aspectos concretos, como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. Es una de las especificaciones de OMG (Object Management Group) más usadas en el mundo del desarrollo de modelos. [FOWLER99].

3.9 Domain Specific Modeling

DSM, o "Domain Specific Modeling" (Modelado Específico de Dominio), es un conjunto de especificaciones para dar solución a un problema con un nivel de abstracción más allá de la programación, especificando la solución mediante la utilización directa de los conceptos de dominio. El producto final es generado a partir de estas especificaciones de alto nivel. [FURTADO06].

3.10 Windows Presentation Foundation

WPF, o "Windows Presentation Foundation" es un subsistema gráfico para la representación de interfaces de usuario en aplicaciones basadas en Windows. WPF, anteriormente conocido como "Avalon", fue publicado inicialmente como parte de .NET Framework 3.0.

Diseñado para eliminar las dependencias en el subsistema GDI, WPF se basa en DirectX, lo cual ofrece aceleración de hardware y permite soportar características de la interfaz de usuario tales como la transparencia, gradientes y transformaciones.

WPF proporciona un modelo de programación coherente para la creación de aplicaciones y una clara separación entre la interfaz de usuario y la lógica de negocio. Ofrece además un lenguaje de marcas, conocido como XAML, que es un medio alternativo para la definición de elementos de la interfaz y las relaciones con otros elementos de ésta. Una aplicación WPF puede, además de ser desplegada en el escritorio, estar alojada en un navegador web. Permite también el control, diseño y desarrollo de los aspectos visuales de los programas de Windows.

Su principal objetivo es unificar una serie de servicios: la interfaz de usuario, gráficos 2D y 3D, tipografía avanzada, gráficos vectoriales, gráficos rasterizados, animación, enlaces de datos, audio y video, entre otros.

3.11 .NET

.NET es un proyecto de Microsoft diseñado para crear una nueva plataforma de desarrollo de software. Está concebido para que haya una independencia total de plataformas hardware, así como para permitir el desarrollo rápido de aplicaciones.

Provisto de un extenso conjunto de soluciones ya instanciadas para las necesidades generales de programación de aplicaciones, administra la ejecución de los programas escritos específicamente con la plataforma.

Es preciso destacar sus principales componentes dentro del marco de trabajo: el conjunto de lenguajes de programación (como C#, lenguaje seleccionado para este proyecto), la Biblioteca de Clases Base o BCL, y el CLR o Entorno Común de Ejecución.

3.12 Patrón de diseño Modelo-Vista-Controlador

MVC, o "Model, View, Controller", es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control, del programa en tres componentes distintos, tal y como define su creador Trygve M. H. Reenskaug. [REENSKAUG78].

Como su nombre indica, consta de tres componentes:

- **Modelo**: dominio de la representación específica de la información con la que la aplicación funciona.
- **Vista**: hace que el modelo aparezca de una forma adecuada para su interacción, generalmente una interfaz de usuario.
- **Controlador**: es quien recibe la información para realizar, a continuación, las llamadas adecuadas al sistema.

Capítulo 4. Planificación del Proyecto y Resumen de Presupuestos

4.1 Planificación

En esta sección se detalla el cronograma de actividades del proyecto, el cual está contemplado para iniciarse del 2 de Mayo de 2012 y finalizar el 24 de Enero de 2013, con una dedicación de seis horas diarias (30 horas semanales).

El proyecto se dividirá en tres etapas de desarrollo, cada una de las cuales se dividirá de la siguiente manera:

- Desarrollo del videojuego base
    - Análisis
    - Diseño
    - Implementación
    - Pruebas
    - Documentación
- Desarrollo de la plantilla
    - Análisis
    - Diseño
    - Implementación
    - Pruebas
    - Documentación
- Desarrollo de la interfaz
    - Análisis
    - Diseño
    - Implementación
    - Pruebas
    - Pruebas Integración
    - Documentación

4.1.1 Cronograma

Para representar la duración, en semanas, de cada etapa del proyecto, se ha utilizado un cronograma, en el que se muestran las fases de cada etapa de desarrollo en la que se va a dividir el proyecto.

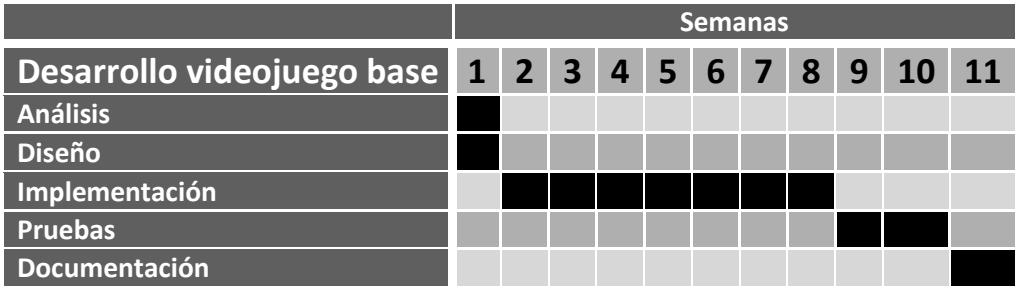


Ilustración 4.1. Cronograma etapa desarrollo Videojuego base

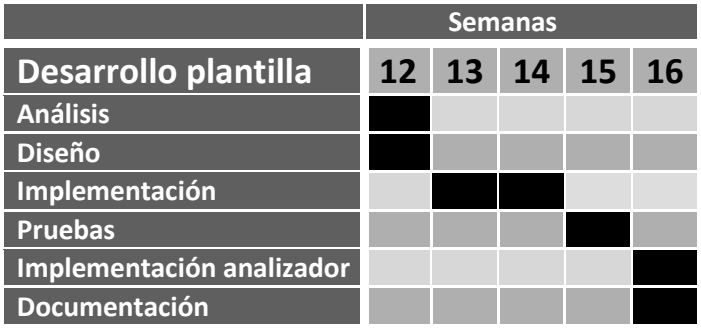


Ilustración 4.2. Cronograma etapa desarrollo Plantilla

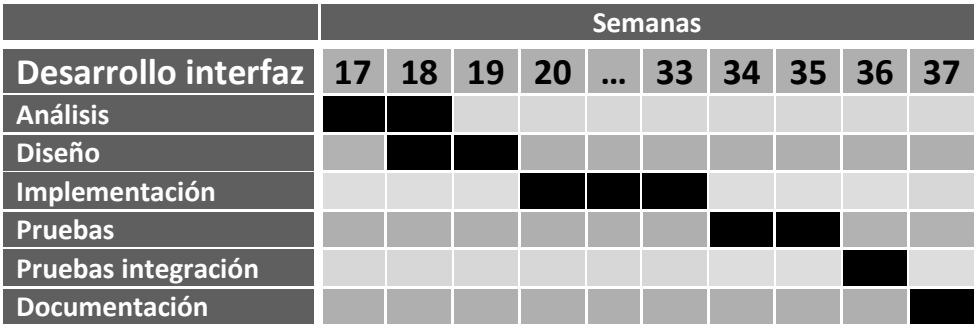


Ilustración 4.3. Cronograma etapa desarrollo Interfaz

4.1.2 Diagrama de Gantt

En la siguiente figura se muestra el diagrama de Gantt correspondiente a la fase de desarrollo del videojuego base, a partir del cual se desarrollará la plantilla.

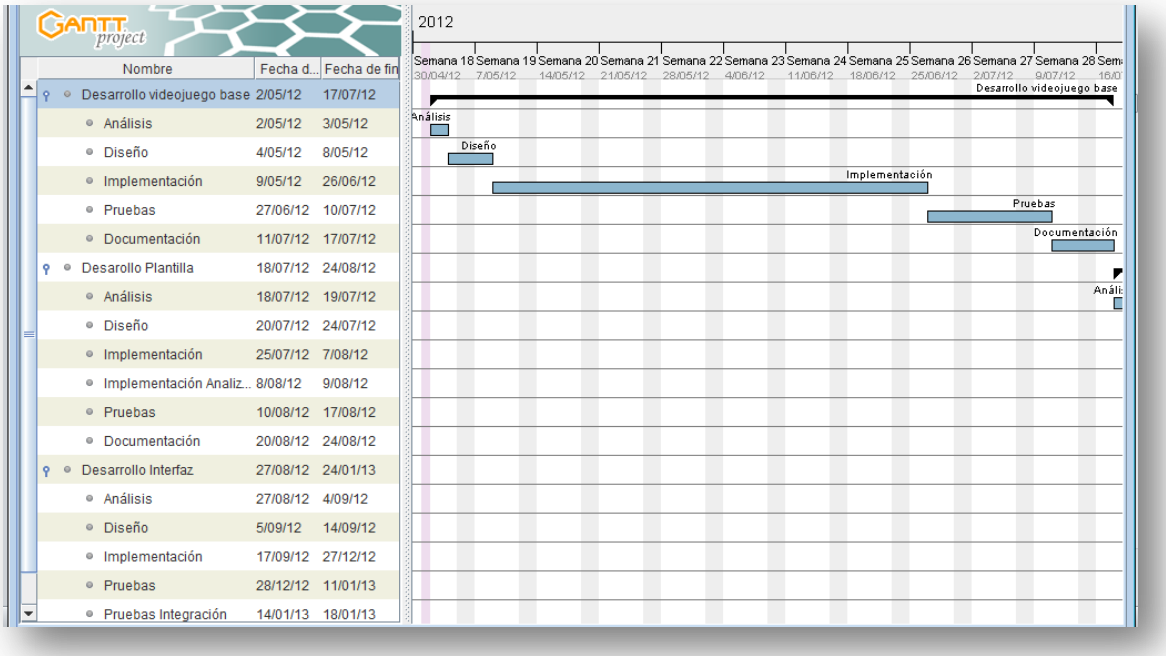


Ilustración 4.4. Diagrama Gantt etapa desarrollo Videojuego Base

La siguiente fase es el desarrollo de la plantilla, a partir del videojuego realizado en la etapa anterior.

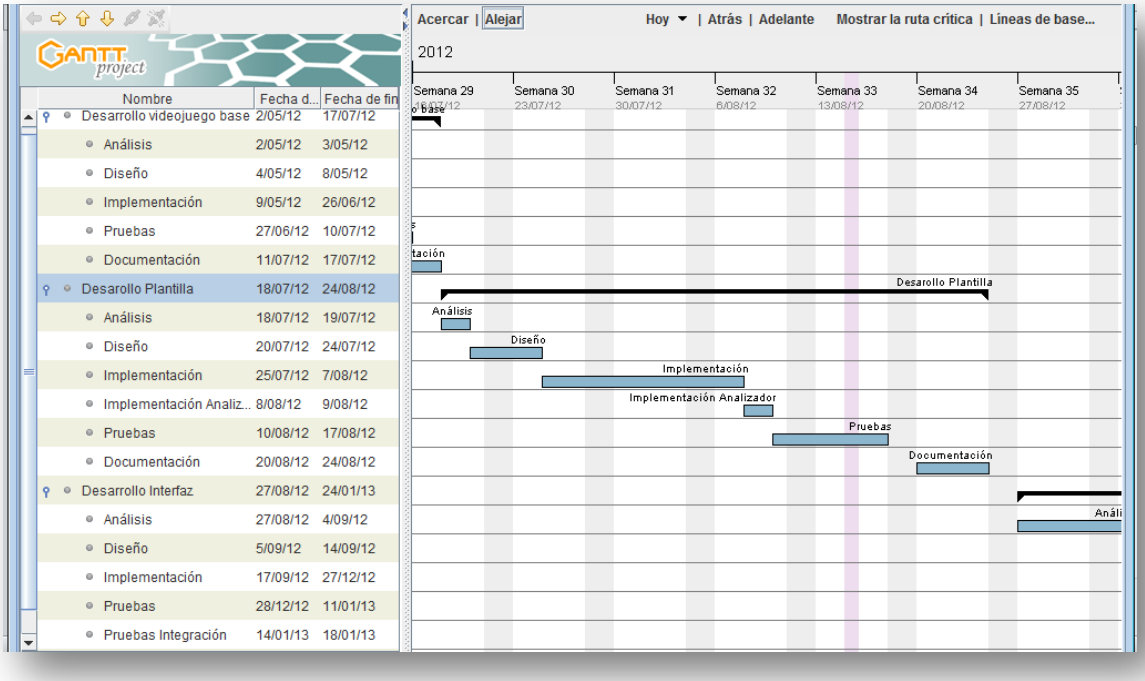


Ilustración 4.5. Diagrama Gantt etapa desarrollo Plantilla

Una vez concluida la fase de creación de la plantilla, el siguiente paso es el desarrollo de la interfaz, que permitirá crear los videojuegos mediante una interfaz de usuario.

Finalizado el desarrollo de la misma, se realizarán las pruebas de integración para asegurarse del correcto funcionamiento de todo el sistema.

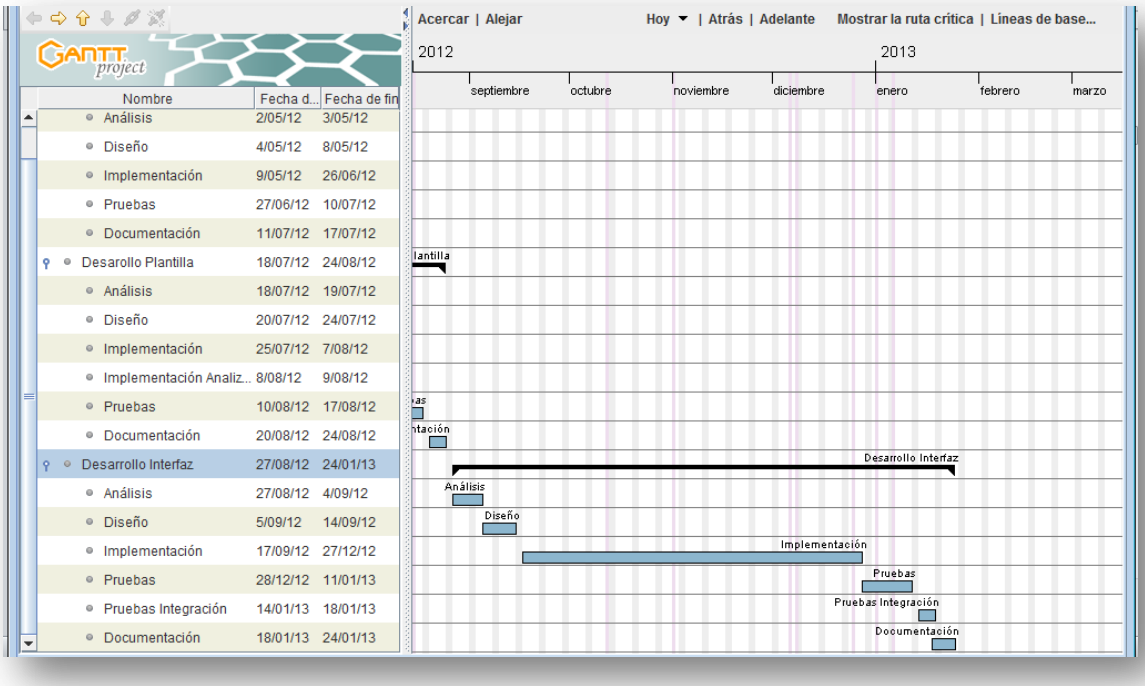


Ilustración 4.6. Diagrama Gantt etapa desarrollo Interfaz

4.2 Resumen del Presupuesto

El resumen que se muestra a continuación contiene el coste estimado para la implementación de este proyecto fin de máster, con una dedicación de seis horas diarias (30 horas semanales).

Se ha dividido en las tres fases en las que se va a dividir este proyecto:

- Desarrollo de los videojuegos base
- Desarrollo de la plantilla
- Desarrollo de la interfaz gráfica

Ítem	SubItem	Concepto	Horas	Precio Unitario €	TOTAL €
1		Desarrollo videojuegos base			7050
	1	Análisis	15	30	450
	2	Diseño	15	30	450
	3	Implementación	210	20	4200
	4	Pruebas	60	25	1500
	5	Documentación	30	15	450

Ítem	SubItem	Concepto	Horas	Precio Unitario €	TOTAL €
2		Desarrollo Plantilla			3375
	1	Análisis	15	30	450
	2	Diseño	15	30	450
	3	Implementación	60	20	1200
	4	Pruebas	30	25	750
	5	Documentación	15	15	225
	6	Implementación Analizador	15	20	300

Ítem	SubItem	Concepto	Horas	Precio Unitario €	TOTAL €
3		Desarrollo Interfaz gráfica			13800
	1	Análisis	45	30	1350
	2	Diseño	45	30	1350
	3	Implementación	420	20	8400
	4	Pruebas	60	25	1500
	5	Pruebas Integración	30	25	750
	6	Documentación	30	15	450

Item	SubItem	Concepto	Unidades	Precio Unitario €	TOTAL €
4		Software (Licencias)			675
	1	Visual Studio 2010 Professional	1	675	675

Item	SubItem	Concepto	Unidades	Precio Unitario €		TOTAL €
5		Hardware		Amortización		231,6
	1	Móvil con Android	1	0.40	79	31,6
	2	Ordenador Portátil	1	0.40	500	200

Subtotal €	25131,6
IVA (21%)	5277,636
TOTAL €	30.409,24

Capítulo 5. Análisis

En este apartado se aborda la descripción detallada y rigurosa del sistema: el Análisis del Sistema de Información (ASI), para este proyecto.

Al no haber realizado la etapa del análisis estudio de viabilidad del sistema (EVS), se incluirá un catálogo en el que se recogerán todos los requisitos del software.

Mediante DIAGRAMAS DE CASOS DE USO (DCU) se especificará la funcionalidad del sistema, representando así gráficamente la relación entre los actores y el sistema. Se incluirá también una descripción pormenorizada de los escenarios y de las interacciones existentes entre los subsistemas.

Las interfaces del usuario con el sistema se definirán mediante una serie de capturas de pantalla acompañadas de una explicación de las mismas. Estas pantallas servirán para visualizar las diferentes funcionalidades del sistema y hacerse una idea de cómo será el proyecto final.

5.1 Definición del Sistema

5.1.1 Determinación del Alcance del Sistema

El objetivo de este proyecto es la creación de videojuegos de acción en 2D para dispositivos móviles con sistema operativo Android. Se pretende que el usuario del sistema sea capaz de crear los videojuegos desde cero sin tener conocimientos de programación, simplemente seleccionando los diferentes elementos que compondrán el juego. Posteriormente se aplicará una plantilla, identificando los diferentes elementos y sustituyendo sus valores por los elegidos por el usuario, y se generará el juego que se podrá instalar directamente en un dispositivo móvil. Previamente a la aplicación de la plantilla se utilizará un lenguaje basado en DSL, creado para este proyecto, para identificar los diferentes elementos que compondrán el juego y sus propiedades.

Por tanto, en este proyecto se pretende construir un generador de videojuegos de acción en 2D para Android de sencillo de manejar por el usuario. Se realizará mediante una instancia del DSL que definirá las reglas que deberá aplicar a la plantilla los datos introducidos por el usuario del sistema.

5.2 Requisitos del Sistema

5.2.1 Obtención de los Requisitos del Sistema

En este apartado se reseñan los requisitos del sistema, los cuales han sido extraídos de los objetivos del proyecto. Se han dividido en requisitos funcionales y no funcionales.

5.2.1.1 Requisitos Funcionales

5.2.1.1.1 Requisitos de Persistencia

Son los que atañen a la información que debe permanecer en el sistema, que se debe guardar entre diferentes ejecuciones del sistema.

Código	Nombre Requisito	Descripción del Requisito
R1.1.1	Crear juego	El sistema debe incluir la opción de crear un nuevo juego.
R1.1.2	Guardar juego	El usuario podrá guardar el estado del juego en cualquier momento.
R1.1.3	Abrir juego	Se podrá abrir un juego previamente almacenado.
R1.1.4	Exportar juego	El sistema debe incluir la opción de generar el juego para la plataforma Android.

5.2.1.1.2 Requisitos Multimedia

Referentes a los recursos multimedia (sonidos e imágenes) que deben ser aceptados por el sistema para su posterior almacenamiento en el mismo.

Código	Nombre Requisito	Descripción del Requisito
R1.2.1	Carga de imágenes	El sistema debe permitir la asignación de gráficos a los diferentes elementos configurables del juego. El formato admitido es .png, ya que la plantilla solo admite imágenes en ese formato.
R1.2.2	Carga de sonidos	El usuario deberá poder asociar una melodía a los menús, otra al juego y a los eventos o acciones llevadas a cabo por los personajes del juego. El formato soportado es mp3 u ogg.

5.2.1.1.3 Requisitos de Ejecución

Hacen referencia a las necesidades de ejecución de la aplicación: control de errores que se puedan producir e indicación de en qué estado se encuentra la ejecución cuando se realizan procesos de larga duración.

Código	Nombre Requisito	Descripción del Requisito
R1.3.1	Detección de errores de ejecución	Se debe detectar todos los errores producidos en la ejecución de la aplicación.
R1.3.2	Estado de la ejecución.	Deberá mostrarse en todo momento el estado de la ejecución de la aplicación (listo o procesando).

5.2.1.1.4 Requisitos de Interacción

Son aquellos que describen la forma en que el usuario interactuará con el sistema.

Código	Nombre Requisito	Descripción del Requisito
R1.4.1	Interacción con la aplicación	El usuario deberá interactuar con el sistema mediante una interfaz.
R1.4.2	Producto obtenido en la generación	El sistema deberá generar un proyecto completo y funcional de un juego en entorno Eclipse, que pueda ser ejecutado en el emulador de Android del Eclipse o bien en un dispositivo Android.

5.2.1.1.5 Requisitos de Usabilidad

Estos requisitos determinan la dificultad de uso de la herramienta, dependiendo de los conocimientos del usuario y la interfaz de usuario.

Código	Nombre Requisito	Descripción del Requisito
R1.5.1	Conocimiento de desarrollo de videojuegos	La herramienta deberá poder usarla por cualquier usuario, independientemente de sus conocimientos en el desarrollo de videojuegos.
R1.5.2	Conocimiento de entorno de programación de Android	El usuario deberá poder utilizar la herramienta aunque carezca de conocimientos sobre el entorno de programación de Android, Eclipse.
R1.5.3	Interfaz intuitiva	La herramienta deberá poseer una interfaz sencilla e intuitiva, que permita que cualquier usuario pueda manejarla sin dificultades.

5.2.1.1.6 Requisitos de la instancia del DSL

En esta tabla se recogen los requisitos que deberá cumplir la instancia del DSL que se obtendrá al final de la generación de la herramienta, tanto en su formato como en su contenido.

Código	Nombre Requisito	Descripción del Requisito
R1.6.1	Formato de la instancia del DSL	La instancia resultado de la generación de videojuegos con la herramienta deberá estar en formato XML.
R1.6.2	Instancia del DSL intuitiva	La instancia del DSL debe contener elementos que posean una lógica intuitiva, que haga que sea fácil de entender.
R1.6.3	Edición manual sin interfaz	La instancia del DSL generado con la herramienta deberá poder ser modificado por el usuario utilizando cualquier editor de textos convencional.

5.2.1.1.7 Requisitos del Juego Generado

A continuación se muestran los requisitos que deberá cumplir el juego generado con la herramienta.

Código	Nombre Requisito	Descripción del Requisito
R1.7.1	Formato del juego	Se deberá obtener un proyecto de Eclipse con la estructura de un proyecto de una aplicación para Android.
R1.7.2	Errores	El juego no deberá contener errores, ya que el usuario solamente deberá importarlo al Eclipse y ejecutarlo.
R1.6.3	Funcionamiento	El juego deberá tener el funcionamiento esperado por el usuario en función del diseño que haya hecho del mismo. Mismo aspecto de los menús, personajes y niveles creados.

5.2.1.2 Requisitos No Funcionales

5.2.1.2.1 Requisitos Tecnológicos

Código	Nombre Requisito	Descripción del Requisito
R2.1.1	Sistema operativo del Editor.	Es necesario el uso del Sistema Operativo Windows XP (o superior) para la creación y exportación de juegos.
R2.1.2	Ejecución de los videojuegos en dispositivo real.	La ejecución real de los juegos requiere la disponibilidad de un dispositivo móvil con Android.
R2.1.3	Framework.	Debe utilizarse el Framework .NET 4.0 o superior.
R2.1.4	IDE para el Editor.	Se necesita el IDE Visual Studio 2010 o una versión

		posterior.
R2.1.5	IDE para la plantilla.	Se necesita el entorno de desarrollo Eclipse y Android Software Development Kit (Android SDK) instalado en Eclipse para el desarrollo de la plantilla y para realizar pruebas con el emulador en caso de no disponer de un dispositivo con Android.

5.2.1.2.2 Requisitos del Sistema

Código	Nombre Requisito	Descripción del Requisito
R2.2.1	Procesador	Debe utilizarse una CPU con procesador Pentium IV a 3 GHz o superior.
R2.2.2	Memoria RAM	Se requiere al menos 1GB de RAM en equipos de 32 bits o 2GB en uno de 64 bits.
R.2.2.3	Disco duro	Es recomendable disponer de un disco duro de 100GB con al menos 3GB libres.
R.2.2.4	Tarjeta Gráfica	Es necesario poseer una tarjeta gráfica compatible con DirectX 10.

5.2.2 Identificación de Actores del Sistema

Aunque la herramienta está desarrollada de tal forma que cualquier usuario pueda utilizarla, se pueden identificar tres tipos de usuarios:

- Usuario Diseñador
- Usuario Jugador
- Usuario Administrador

5.2.2.1 Usuario Diseñador

Este usuario es el que utilizará la herramienta para crear, editar y exportar videojuegos a través de la interfaz de la misma. No es necesario que este usuario posea conocimientos o experiencia relacionados con los lenguajes de programación, ni el desarrollo de aplicaciones en Android o de videojuegos.

5.2.2.2 Usuario Jugador

El usuario Jugador es quien prueba la labor realizada por el usuario Diseñador, en un dispositivo móvil con sistema operativo Android o bien con el emulador de Android disponible en el entorno de desarrollo de Eclipse. Si se prueba con el emulador tampoco se requieren conocimientos de programación ya que en el manual de usuario se explica de forma clara y sencilla como realizar este proceso.

5.2.2.3 Usuario Administrador

El usuario Administrador es el que tendrá acceso a la instancia del DSL y a la plantilla. La principal función de este usuario es crear una nueva instancia de DSL o modificar una existente, en caso de no utilizar la interfaz de la que dispone la herramienta. También podrá modificar o crear una plantilla, en caso de querer cambiar el comportamiento de alguno de los elementos de la misma o bien añadir uno nuevo. Sí es necesario que este usuario tenga conocimientos de programación, para intervenir en la instancia del DSL y/o la plantilla. Este usuario podrá realizar todas las acciones que engloba la herramienta.

5.2.3 Especificación de Casos de Uso

A través de los siguientes apartados, se especificarán los diferentes casos de uso identificados en este proyecto, señalando para cada uno que actor le corresponde.

5.2.3.1 Crear nuevo Videojuego

Nombre del Caso de Uso
Crear nuevo juego
Descripción
El usuario diseñador podrá crear un nuevo juego mediante la interfaz de la que dispone la herramienta. Debe ser este usuario el que se encargue de editar fuera de la aplicación las imágenes y sonidos que se van a utilizar en el diseño del videojuego. Dentro de este caso de uso se puede hacer una clasificación de las posibilidades que tiene este usuario. Al final del proceso de creación, el programa preguntará al usuario dónde quiere almacenar el juego creado.

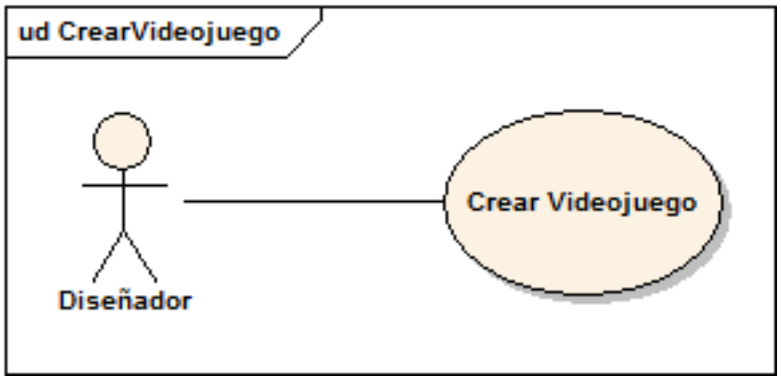


Ilustración 5.1. Caso de Uso: Crear Videojuego

Dentro del caso de uso "Crear nuevo Videojuego", se puede hacer una subdivisión de los siguientes casos de uso:

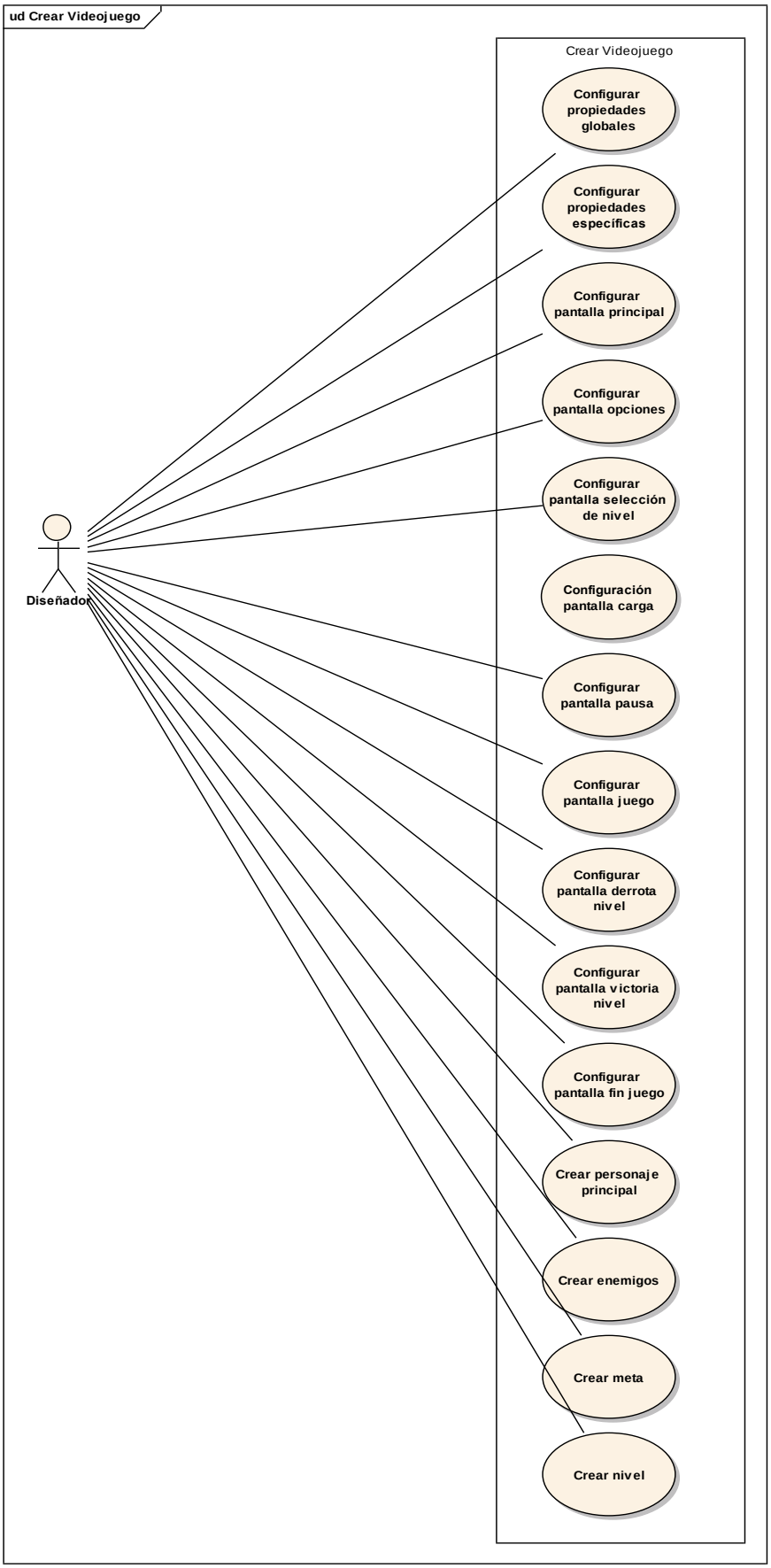


Ilustración 5.2. Subcasos de Uso de CrearVideojuego

5.2.3.1.1 Configurar propiedades globales

Nombre del Caso de Uso
Configuración propiedades globales
Descripción
El usuario diseñador podrá configurar las propiedades globales del juego que va a crear; escoger el nombre del juego así como el icono; y configurar otros parámetros que son opcionales.

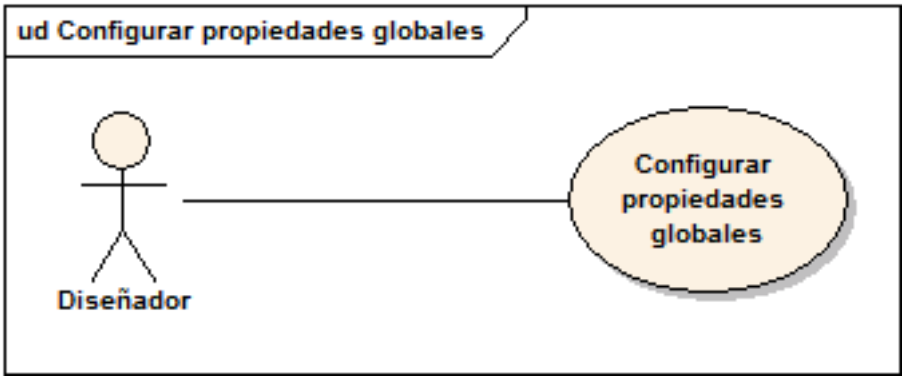


Ilustración 5.3. Subcaso de Uso: Configurar propiedades globales

5.2.3.1.2 Configurar propiedades específicas

Nombre del Caso de Uso
Configuración propiedades específicas
Descripción
El usuario diseñador podrá configurar las propiedades específicas del juego, escoger el tipo de juego que va a crear dentro de la tipología en la que se encuentra, y también podrá configurar otras propiedades más específicas de cada tipo de juego disponible.

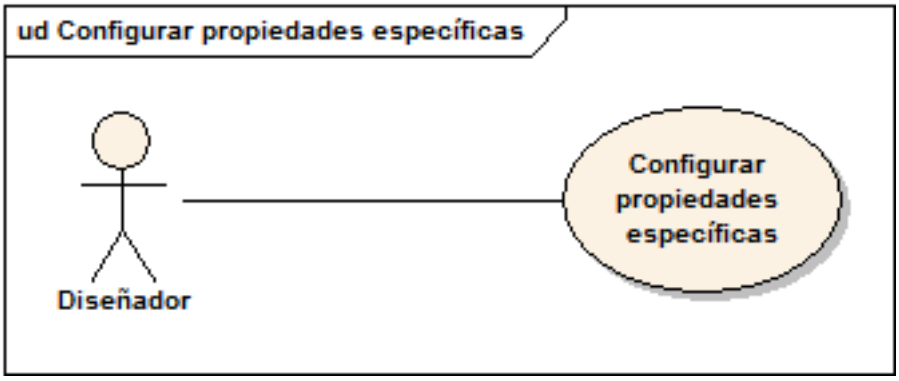


Ilustración 5.4. Subcaso de Uso: Configurar propiedades específicas

5.2.3.1.3 Configurar pantalla principal

Nombre del Caso de Uso
Configuración pantalla principal
Descripción
El usuario diseñador podrá configurar la pantalla del menú principal del juego, la apariencia y ubicación de los botones, y la imagen de fondo.

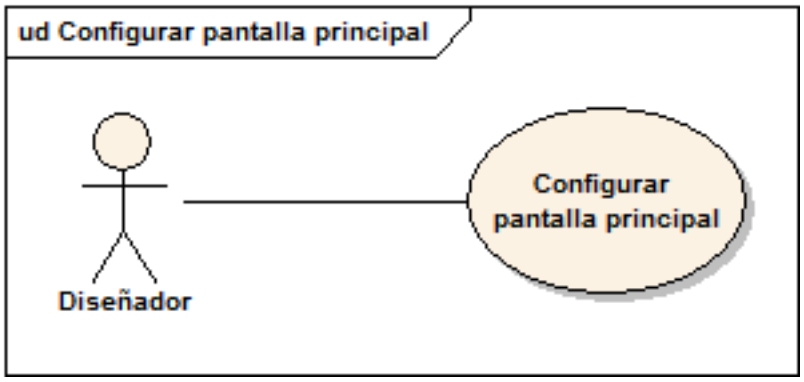


Ilustración 5.5. Subcaso de Uso: Configurar pantalla principal

5.2.3.1.4 Configurar pantalla de opciones

Nombre del Caso de Uso
Configuración pantalla opciones
Descripción
El usuario diseñador podrá configurar la pantalla del menú opciones del juego, la apariencia y ubicación de los botones y la imagen de fondo.

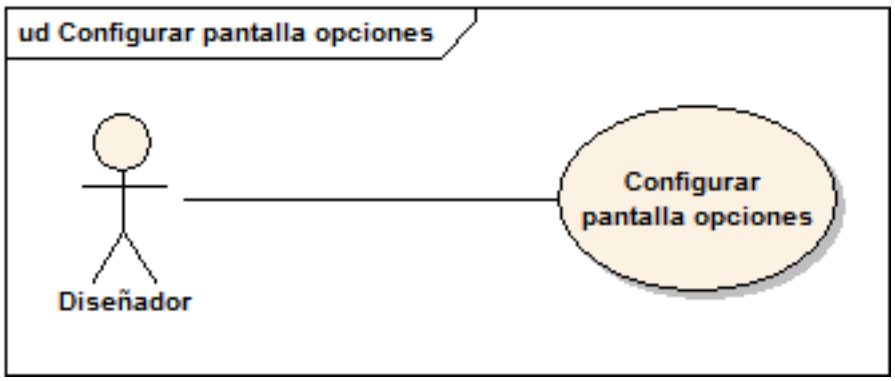


Ilustración 5.6. Subcaso de Uso: Configurar pantalla opciones

5.2.3.1.5 Configurar pantalla selección de nivel

Nombre del Caso de Uso
Configuración pantalla selección de nivel
Descripción
El usuario diseñador podrá configurar la pantalla del menú de selección de nivel del juego, la apariencia y ubicación de los botones y la imagen de fondo de la pantalla.

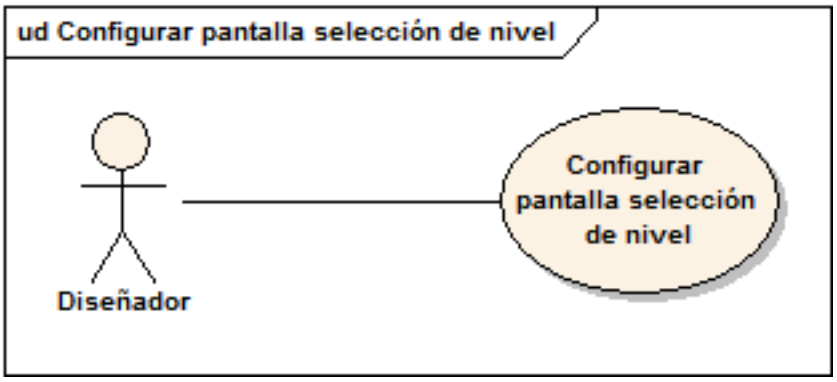


Ilustración 5.7. Subcaso de Uso: Configurar pantalla selección de nivel

5.2.3.1.6 Configurar pantalla de carga

Nombre del Caso de Uso
Configuración pantalla carga
Descripción
El usuario diseñador podrá configurar la pantalla carga del juego. Podrá configurar únicamente la imagen de fondo de la pantalla.

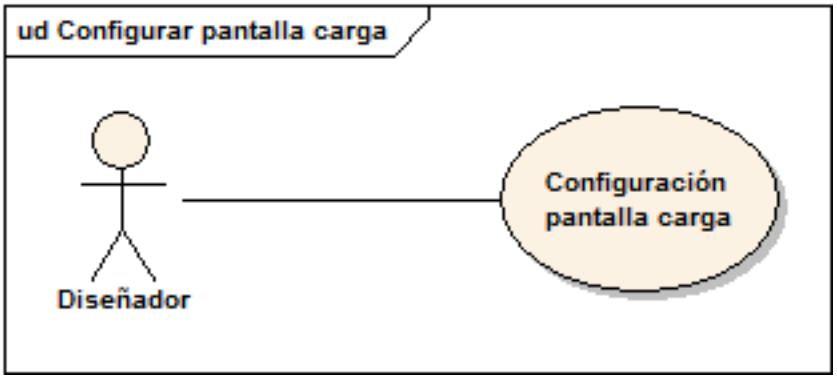


Ilustración 5.8. Subcaso de Uso: Configuración pantalla carga

5.2.3.1.7 Configurar pantalla pausa juego

Nombre del Caso de Uso
Configuración pantalla pausa
Descripción
El usuario diseñador podrá configurar la pantalla de pausa del juego, la apariencia y ubicación de los botones y la imagen de fondo.

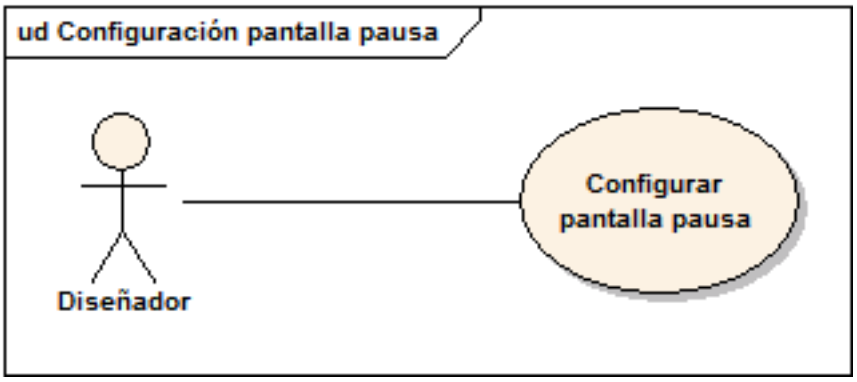


Ilustración 5.9. Subcaso de Uso: Configurar pantalla pausa

5.2.3.1.8 Configurar pantalla juego

Nombre del Caso de Uso
Configuración pantalla juego
Descripción
El usuario diseñador podrá configurar los elementos de la barra de estado de la pantalla de juego, la apariencia y ubicación de los iconos y marcadores y la imagen de fondo.

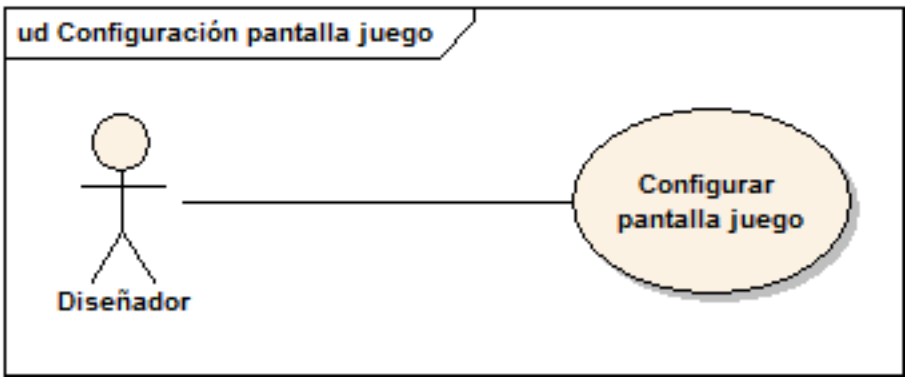


Ilustración 5.10. Subcaso de Uso: Configurar pantalla juego

5.2.3.1.9 Configurar pantalla derrota nivel

Nombre del Caso de Uso
Configuración pantalla derrota
Descripción
El usuario diseñador podrá configurar la pantalla del menú de derrota de nivel del juego, la apariencia y ubicación de los botones y la imagen de fondo.

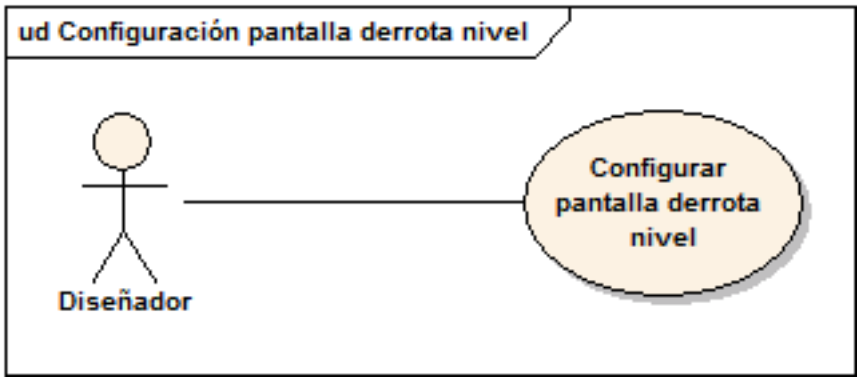


Ilustración 5.11. Subcaso de Uso: Configurar pantalla derrota nivel

5.2.3.1.10 Configurar pantalla victoria nivel

Nombre del Caso de Uso
Configuración pantalla victoria
Descripción
El usuario diseñador podrá configurar la pantalla del menú de victoria de nivel del juego, la apariencia y ubicación de los botones y la imagen de fondo.

Ilustración 5.12. Subcaso de Uso: Configurar pantalla victoria nivel

5.2.3.1.11 Configurar pantalla fin de juego

Nombre del Caso de Uso
Configuración pantalla fin
Descripción
El usuario diseñador podrá configurar la pantalla del menú de fin de juego, la apariencia y ubicación de los botones y la imagen de fondo.

Ilustración 5.13. Subcaso de Uso: Configurar pantalla fin juego

5.2.3.1.12 Crear personaje principal

Nombre del Caso de Uso
Creación personaje principal
Descripción
El usuario diseñador podrá crear el personaje principal del juego y configurarlo. Podrá configurar las imágenes de las animaciones y las propiedades del personaje principal, que determinarán su comportamiento dentro del juego. Solamente se podrá crear un único personaje principal. Se le podrá asociar sonidos dependiendo de su comportamiento.

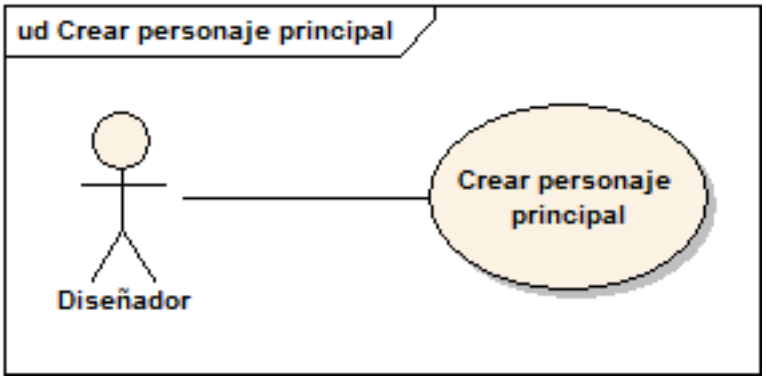


Ilustración 5.14. Subcaso de Uso: Crear personaje principal

5.2.3.1.13 Crear enemigos

Nombre del Caso de Uso
Creación enemigos
Descripción
El usuario diseñador podrá crear y configurar los enemigos que estará presentes en el juego. Podrá crear tantos como desee y podrá configurar las imágenes de las animaciones y las propiedades de los enemigos, que determinarán su comportamiento dentro del juego. También se les podrá asociar sonidos.

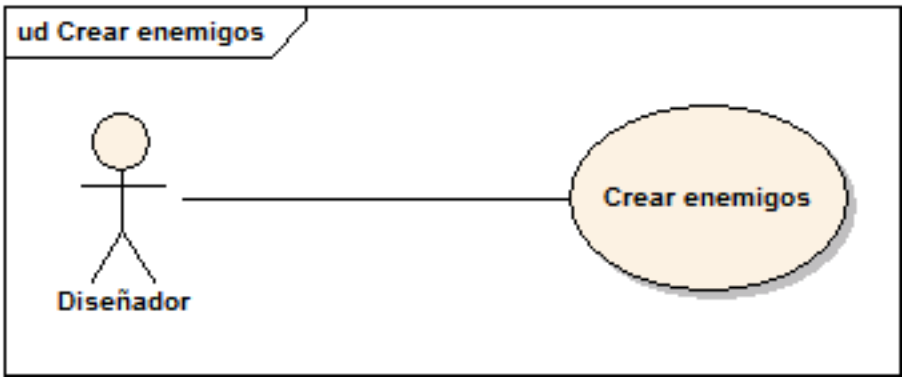


Ilustración 5.15. Subcaso de Uso: Crear enemigos

5.2.3.1.14 Crear meta

Nombre del Caso de Uso
Creación meta
Descripción
El usuario diseñador podrá crear y modificar la meta del juego. Podrá escoger la imagen que la representará dentro del juego, así como el sonido que se reproducirá cuando el personaje principal la alcance.

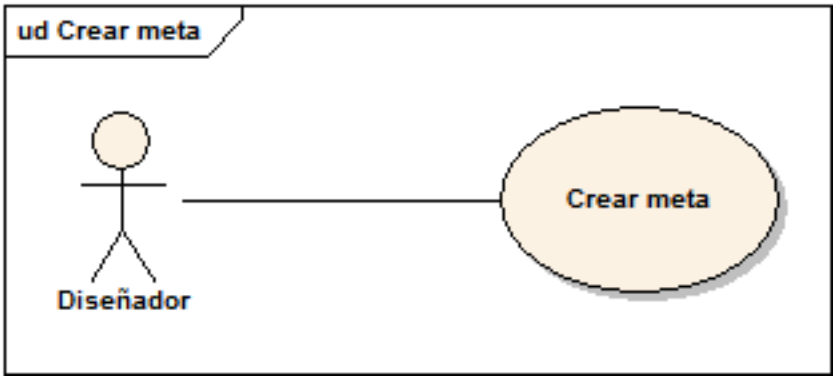


Ilustración 5.16. Subcaso de Uso: Crear meta

5.2.3.1.15 Crear nivel

Nombre del Caso de Uso
Creación nivel
Descripción
El usuario diseñador podrá crear tantos niveles para el juego como desee. Se podrá configurar las propiedades del nivel, escoger la imagen de descripción del mismo, la de la miniatura del nivel que aparecerá en el menú de selección de nivel, y la del fondo. Podrá colocar los personajes creados en el escenario para dar al nivel la apariencia que desee.

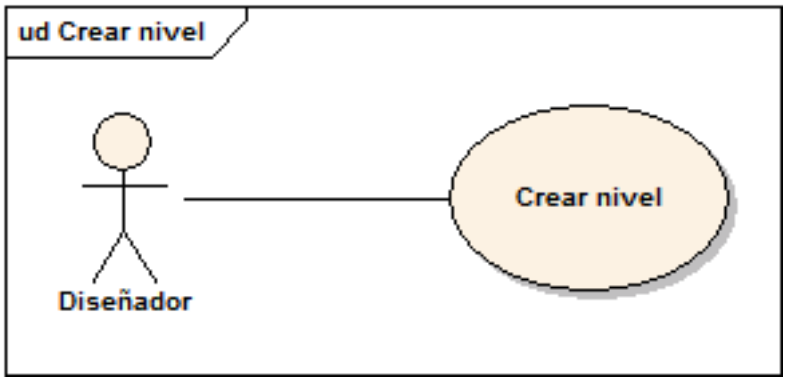


Ilustración 5.17. Subcaso de Uso: Crear nivel

5.2.3.2 Guardar/Cargar estado del videojuego

5.2.3.2.1 Guardar estado videojuego

Nombre del Caso de Uso
Guardar juego
Descripción
El usuario diseñador podrá guardar el estado de un juego nuevo o bien de uno ya existente y que está modificando. Este proceso se llevará a cabo cada vez que el usuario pase de una pantalla de la aplicación a otra o bien desde el menú de navegación, usando la opción "Guardar". El estado se almacenará en ficheros binarios .dat, uno por cada pantalla.

Ilustración 5.18. Caso de Uso: Guardar videojuego

5.2.3.2.2 Cargar estado videojuego

Nombre del Caso de Uso
Cargar juego
Descripción
El usuario diseñador podrá cargar el estado de desarrollo de un juego guardado desde la herramienta y posteriormente almacenado. Se mostrará al usuario el estado en el que quedó la aplicación en el momento en que se salvó. Se cargarán los ficheros .dat ubicados en la carpeta del juego.

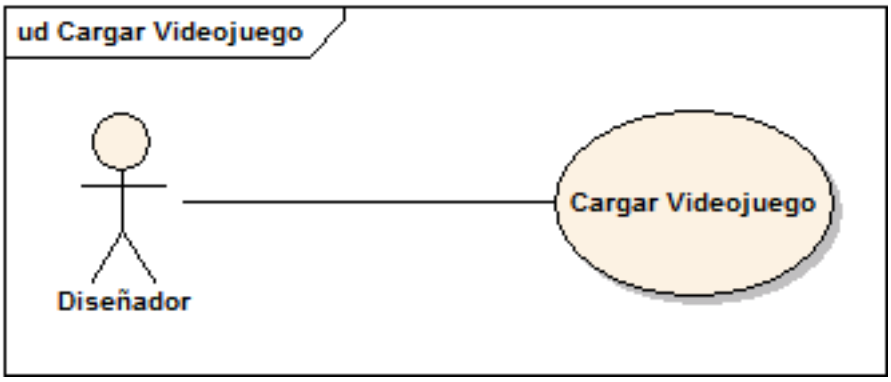


Ilustración 5.19. Caso de Uso: Cargar Videojuego

5.2.3.3 Exportar Videojuego

Nombre del Caso de Uso
Exportar Videojuego
Descripción
El usuario diseñador podrá exportar el juego creado con el editor. Deberá seleccionar la ruta donde desea almacenar el juego que se va a generar. En esa ruta se almacenará el proyecto de Eclipse generado. Deberá ser importado al entorno de desarrollo y compilarlo. A continuación se obtendrá el proyecto listo para instalar en un dispositivo móvil.

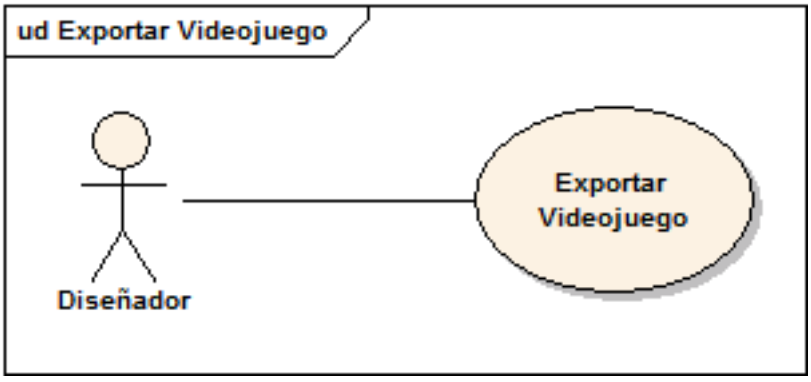


Ilustración 5.20. Caso de Uso: Exportar Videojuego

5.2.3.4 Crear/Editar instancia DSL

Nombre del Caso de Uso
Crear/Editor instancia DSL
Descripción
El usuario administrador podrá crear o editar la instancia del DSL manualmente antes de pasar por el analizador, para la posterior generación del juego. De esta manera podrá crear un juego nuevo o modificar uno sin falta de utilizar la interfaz de la herramienta.

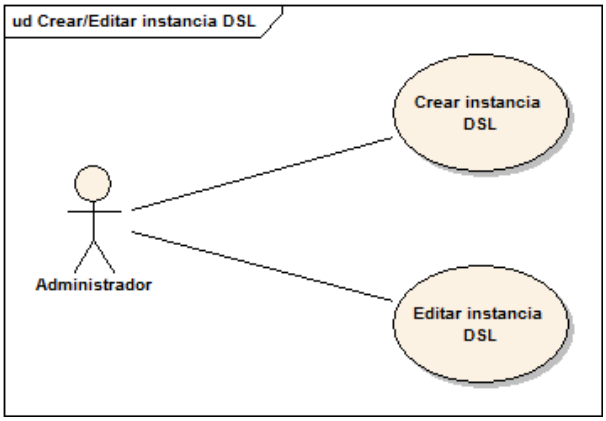


Ilustración 5.21. Caso de Uso: Crear/Editar instancia DSL

5.2.3.5 Crear/Editar Plantilla

Nombre del Caso de Uso
Crear/Editar plantilla
Descripción
El usuario administrador podrá crear una plantilla nueva o bien modificar una existente, modificando los valores extraídos de la instancia del DSL, los archivos de los niveles o el comportamiento del juego.

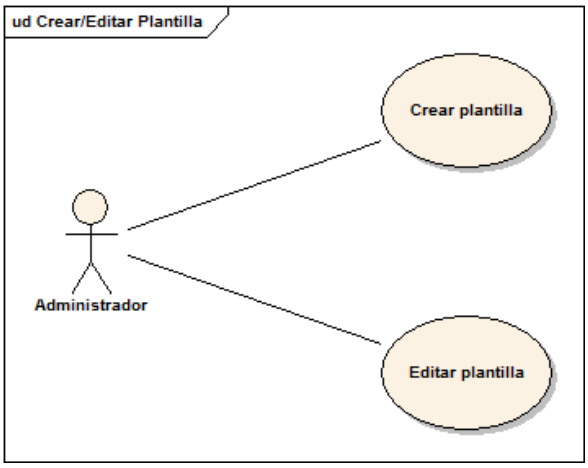


Ilustración 5.22. Caso de Uso: Crear/Editar Plantilla

5.2.3.6 Jugar videojuego

Nombre del Caso de Uso
Jugar videojuego
Descripción
El usuario jugador podrá jugar al videojuego previamente creado por el usuario diseñador, o bien en un dispositivo móvil o en un emulador de dispositivos Android.

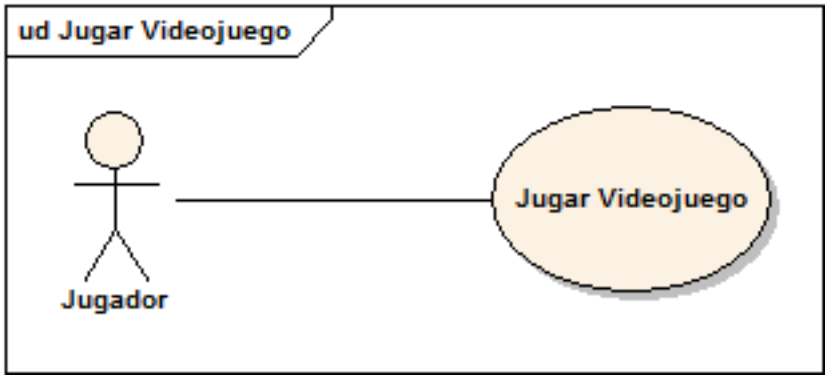


Ilustración 5.23. Caso de Uso: Jugar Videojuego

5.3 Identificación de los Subsistemas en la Fase de Análisis

Para facilitar el análisis del sistema, en este epígrafe se describirán y explicarán los subsistemas que lo componen.

5.3.1 Descripción de los Subsistemas

Los subsistemas que componen este proyecto son los siguientes:

- Subsistema interfaz (Interfaz de usuario).
- Subsistema Analizador (Motor de transformación).
- Subsistema Plantilla de Juego

5.3.1.1 Subsistema interfaz (Interfaz de usuario)

Se encarga de proveer al usuario de una interfaz sencilla e intuitiva con la cual pueda interactuar para crear videojuegos.

5.3.1.1.1 Configuración menús

Hace referencia a la configuración de todas las pantallas de los menús de los juegos.

5.3.1.1.2 Creación personajes

Corresponde a la creación de personajes del juego: personaje principal, enemigos y meta.

5.3.1.1.3 Creación niveles

Permite crear los niveles que formarán el juego con los personajes creados en el subsistema anterior.

5.3.1.1.4 Generación instancia DSL

Construye la instancia del DSL correspondiente al juego creado, con los datos introducidos por el usuario.

5.3.1.2 Subsistema Analizador (Motor de transformación)

Está relacionado con la gestión de los datos procedentes de la interfaz de usuario. Este motor de transformación recoge los datos almacenados en la instancia del DSL y los sustituye en las partes variables de la plantilla.

5.3.1.3 Subsistema Plantilla de Juego

Es el subsistema que contiene la plantilla y mediante el motor de transformación, permitirá construir el juego.

5.3.2 Descripción de los Interfaces entre Subsistemas

Todos estos subsistemas forman parte del mismo proyecto, por tanto se conectan de forma local y no mediante interfaces.

5.4 Diagrama de Clases Preliminar del Análisis

Como complemento gráfico de la descripción de los subsistemas identificados en el punto 5.3 se muestran a continuación varios diagramas de clases, referentes al sistema en su conjunto y a los subsistemas Interfaz, Analizador y Plantilla, y un diagrama de paquetes de todo el sistema.

5.4.1 Diagrama de Paquetes

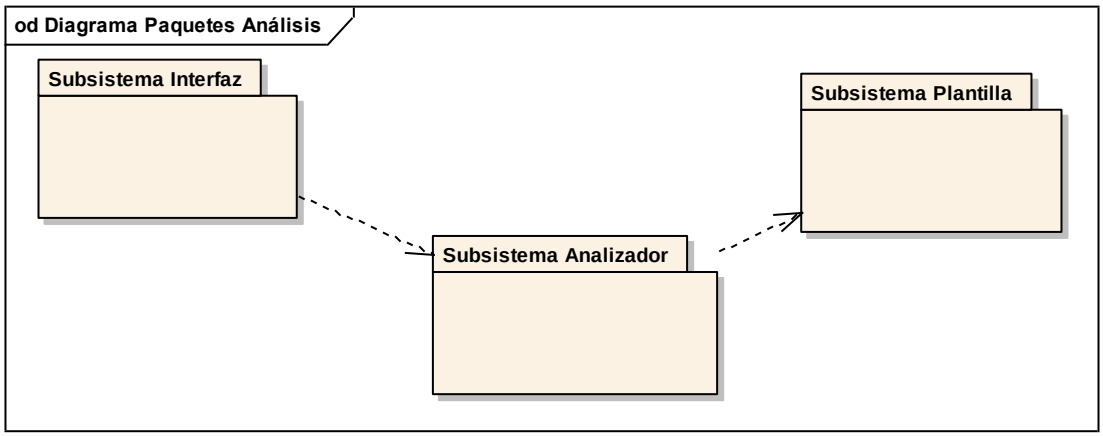


Ilustración 5.24. Diagrama de Paquetes Análisis Global

Tal y como se observa en el diagrama anterior, el sistema sigue el patrón de diseño Modelo-Vista-Controlador (MVC), donde el subsistema Interfaz representa la vista, el subsistema Analizador el controlador y el subsistema Plantilla el modelo.

5.4.1.1 Diagrama de Clases Subsistema Interfaz

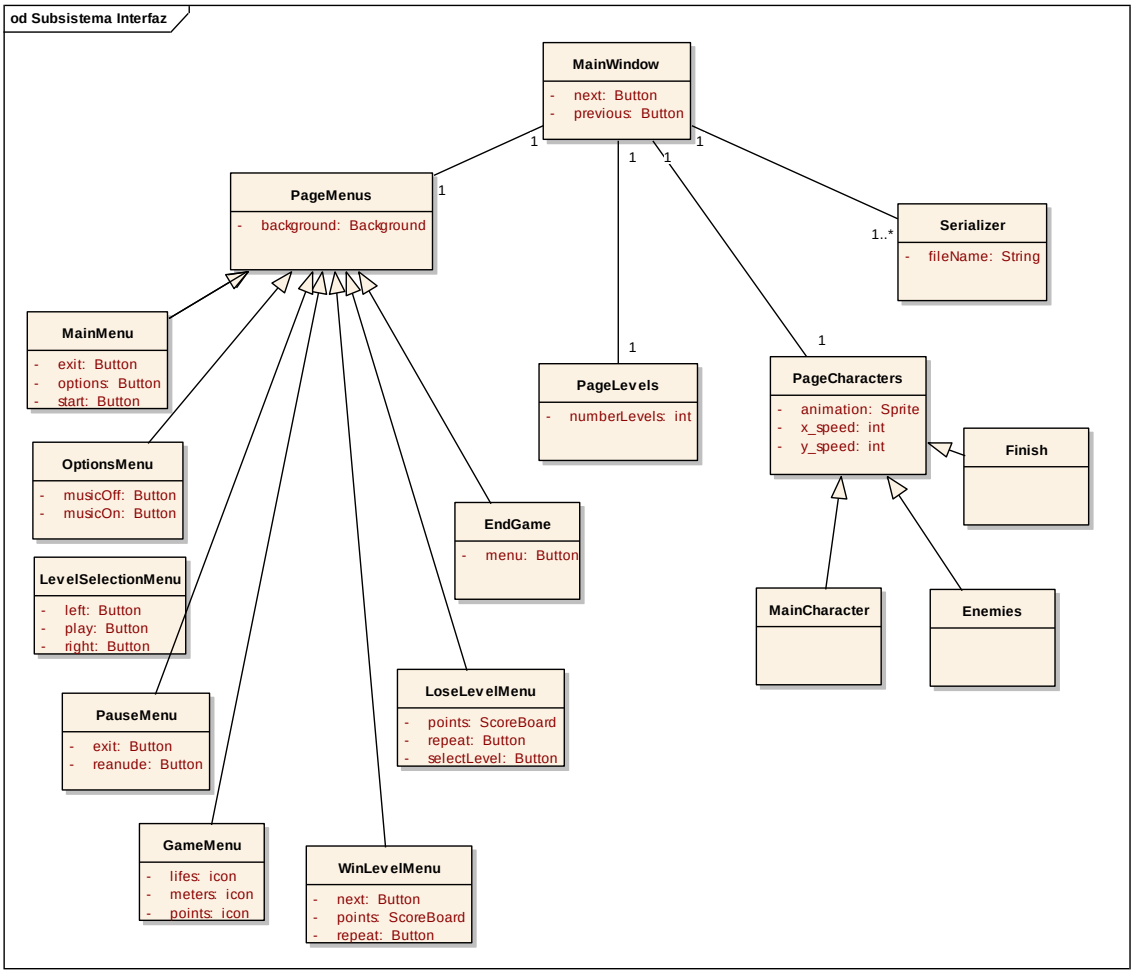


Ilustración 5.25. Diagrama de Clases Análisis Subsistema interfaz

5.4.1.2 Diagrama de Clases Subsistema Analizador

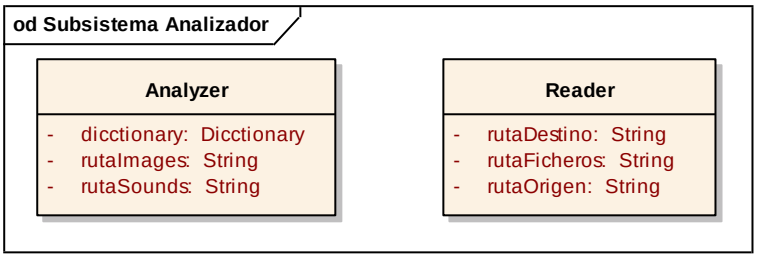


Ilustración 5.26. Diagrama de Clases Análisis Subsistema Analizador

5.4.1.3 Diagrama de Clases Subsistema Plantilla

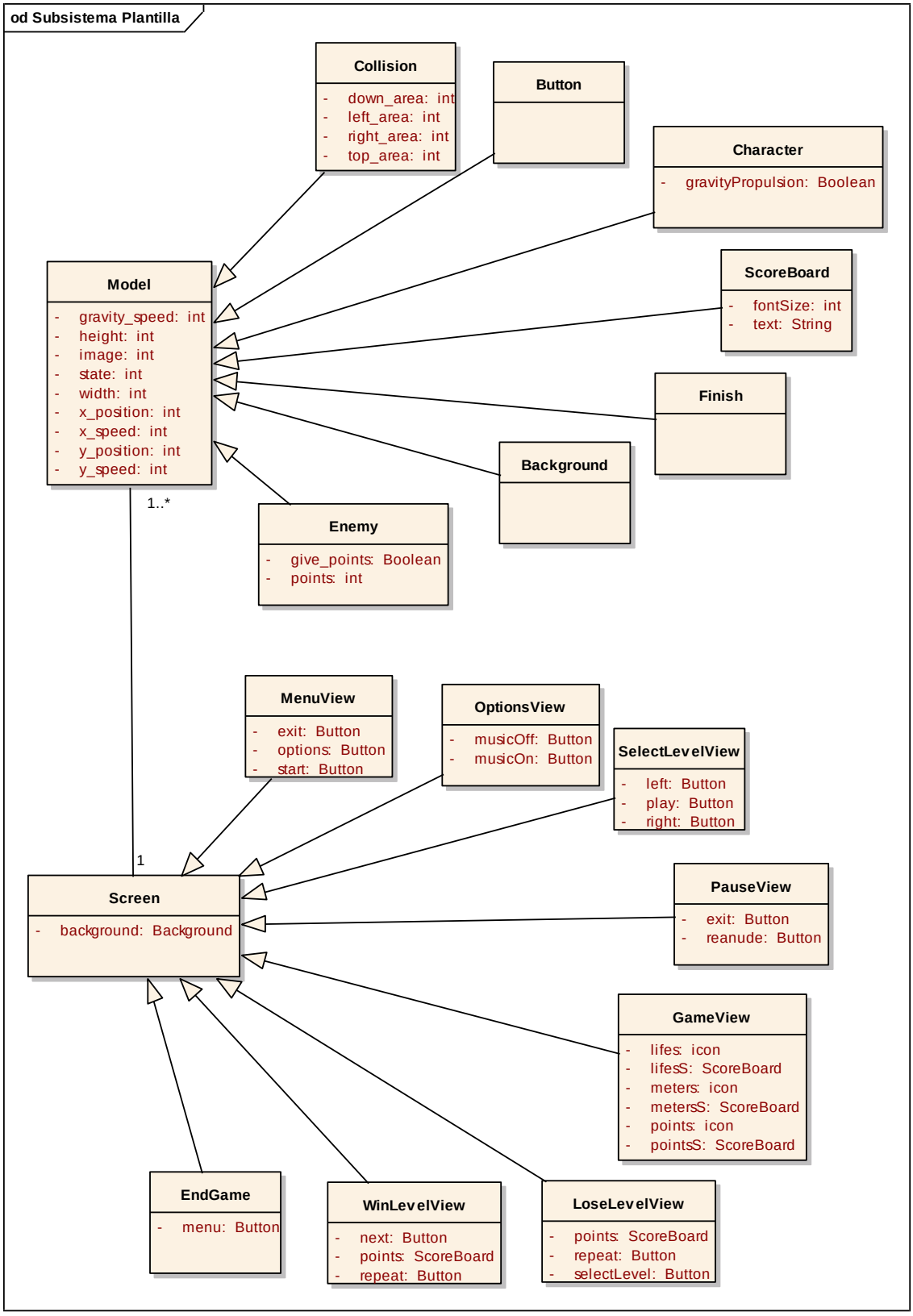


Ilustración 5.27. Diagrama de Clases Análisis Subsistema Plantilla

5.4.2 Descripción de las Clases

Las clases del sistema deberán estar organizadas por los subsistemas anteriores, cubriendo una tabla como la que a continuación se indica, por cada clase.

5.4.2.1 Subsistema Interfaz

Nombre de la Clase
MainWindow
Descripción
Es la clase correspondiente a la pantalla principal de la interfaz.
Responsabilidades
Es la pantalla principal de la interfaz y su objetivo es contener a las pantallas que se vayan abriendo en la interfaz.
Atributos Propuestos
nextPage: Se utilizará para navegar hacia adelante dentro de la interfaz. **previousPage**: Servirá para navegar hacia atrás dentro de la interfaz.
Métodos Propuestos
nextPage: Abrirá la siguiente pantalla a la actual y salvará el estado de esa pantalla. **preivousPage**: Abrirá la anterior a la actual.
saveState: Llamará al método salvar de la pantalla que este cargada en ese momento.

Nombre de la Clase
PageMenus
Descripción
Representa las páginas de los menús.
Responsabilidades
Proporciona funcionalidad para configurar los elementos de cada pantalla.
Atributos Propuestos
background: imagen de fondo.
Métodos Propuestos

Nombre de la Clase
MainMenu
Descripción
Representa la pantalla de configuración del menú principal
Responsabilidades
Proporciona funcionalidad para modificar los elementos que tendrá la pantalla del menú principal del juego.
Atributos Propuestos
startButton: botón de jugar. **exitButton**: botón de salir. **optionsButton:** botón de opciones.
Métodos Propuestos
saveState: Método que salvará en disco el estado de la pantalla. **loadState**: Método que cargará el estado de la pantalla.

Nombre de la Clase
OptionsMenu
Descripción
Clase que representa la pantalla de configuración del menú de opciones
Responsabilidades
Proporciona funcionalidad para modificar los elementos que tendrá la pantalla del menú de opciones del juego.
Atributos Propuestos
musicOffButton: botón de desactivar música. **musicOnButton**: botón de activar música.
Métodos Propuestos
saveState: Método que salvará en disco el estado de la pantalla. **loadState**: Método que cargará el estado de la pantalla.

Nombre de la Clase
LevelSelectMenu
Descripción
Clase que representa la pantalla de configuración del menú de selección de nivel
Responsabilidades
Proporciona funcionalidad para modificar los elementos que tendrá la pantalla del menú de selección de nivel del juego.
Atributos Propuestos
leftButton: botón de mostrar imagen siguiente nivel. **rightButton**: botón de mostrar imagen nivel anterior. **playButton**: botón de jugar el nivel elegido.
Métodos Propuestos
saveState: Método que salvará en disco el estado de la pantalla. **loadState**: Método que cargará el estado de la pantalla.

Nombre de la Clase
PauseMenu
Descripción
Clase que representa la pantalla de configuración del menú de pausa.
Responsabilidades
Proporciona funcionalidad para modificar los elementos que tendrá la pantalla del menú de pausa del juego.
Atributos Propuestos
reanudeButton: botón de reanudar el juego. **exitButton**: botón de salir del nivel.
Métodos Propuestos
saveState: Método que salvará en disco el estado de la pantalla. **loadState**: Método que cargará el estado de la pantalla.

Nombre de la Clase
GameMenu
Descripción
Clase que representa la pantalla de configuración de la pantalla de juego.
Responsabilidades
Proporciona funcionalidad para modificar los elementos que tendrá la pantalla de juego del juego.
Atributos Propuestos
lifesIcon: icono de las vidas. **pointsIcon**: icono de los puntos**.** **metersIcon**: icono de los metros. **lifeScoreboard:** marcador de las vidas. **pointsScoreboard**: marcador de los puntos. **metersScoreboard**: marcador de los metros.
Métodos Propuestos
saveState: Método que salvará en disco el estado de la pantalla. **loadState**: Método que cargará el estado de la pantalla.

Nombre de la Clase
WinLevelMenu
Descripción
Clase que representa la pantalla de configuración de la pantalla de victoria de nivel.
Responsabilidades
Proporciona funcionalidad para modificar los elementos que tendrá la pantalla del menú de victoria de nivel del juego.
Atributos Propuestos
repeatButton: botón de repetir el nivel. **nextButton**: botón de jugar siguiente nivel.
Métodos Propuestos
saveState: Método que salvará en disco el estado de la pantalla. **loadState**: Método que cargará el estado de la pantalla.

Nombre de la Clase
LoseLevelMenu
Descripción
Clase que representa la pantalla de configuración de la pantalla de derrota de nivel.
Responsabilidades
Proporciona funcionalidad para modificar los elementos que tendrá la pantalla del menú de derrota de nivel del juego.
Atributos Propuestos
repeatButton: botón de repetir el nivel. **selectLevelButton**: botón de ir a la pantalla de selección de nivel
Métodos Propuestos
saveState: Método que salvará en disco el estado de la pantalla. **loadState**: Método que cargará el estado de la pantalla.

Nombre de la Clase
GameMenu
Descripción
Clase que representa la pantalla de configuración de la pantalla de victoria de nivel.
Responsabilidades
Proporciona funcionalidad para modificar los elementos que tendrá la pantalla del menú de victoria de nivel del juego.
Atributos Propuestos
repeatButton: botón de repetir el nivel. **nextButton**: botón de jugar siguiente nivel.
Métodos Propuestos
saveState: Método que salvará en disco el estado de la pantalla. **loadState**: Método que cargará el estado de la pantalla.

Nombre de la Clase
PageLevel
Descripción
Representa la pantalla de configuración de los niveles.
Responsabilidades
Proporciona funcionalidad para crear el escenario para cada nivel del juego.
Atributos Propuestos
numberLevels: número de niveles creados.
Métodos Propuestos
saveState: Método que salvará en disco el estado de la pantalla. **loadState**: Método que cargará el estado de la pantalla. **createLevel**: Método que crea la instancia del DSL asociado al nivel.

Nombre de la Clase
Pagecharacter
Descripción
Clase que representa la pantalla de configuración de propiedades de los personajes
Responsabilidades
Proporciona funcionalidad para configurar las propiedades de los personajes del juego.
Atributos Propuestos
animationSprite: imagen de la animación del personaje. **x_speed:** velocidad del personaje en el eje x. **y_speed:** velocidad del personaje en el eje y.
Métodos Propuestos
saveState: Método que salvará en disco el estado de la pantalla. **loadState**: Método que cargará el estado de la pantalla.

Nombre de la Clase
mainCharacter
Descripción
Clase que representa la pantalla de configuración de propiedades del personaje principal.
Responsabilidades
Proporciona funcionalidad para crear y configurar el personaje principal del juego.

Atributos Propuestos
Métodos Propuestos
saveState: Método que salvará en disco el estado de la pantalla. **loadState**: Método que cargará el estado de la pantalla. **createMainCharacter**: Método que creará el personaje principal.

Nombre de la Clase
finish
Descripción
Clase que representa la pantalla de configuración de propiedades de la meta.
Responsabilidades
Proporciona funcionalidad para crear y configurar la meta del juego.
Atributos Propuestos
Métodos Propuestos
saveState: Método que salvará en disco el estado de la pantalla. **loadState**: Método que cargará el estado de la pantalla. **createFinish**: Método que creará la meta.

Nombre de la Clase
enemies
Descripción
Clase que representa la pantalla de configuración de propiedades de los enemigos.
Responsabilidades
Proporciona funcionalidad para crear y configurar los enemigos del juego.
Atributos Propuestos
Métodos Propuestos
saveState: Método que salvará en disco el estado de la pantalla. **loadState**: Método que cargará el estado de la pantalla. **createEnemy**: Método que creará el enemigo.

5.4.2.2 Subsistema Analizador

Nombre de la Clase
Analyzer
Descripción
Clase utilizada para analizar los ficheros necesarios para la generación del juego, y también para hacer la sustitución de los valores obtenidos de la instancia del DSL en la plantilla para generar el juego.
Responsabilidades
Es la clase encargada de copiar los recursos y niveles y de crear la carpeta de destino donde se almacenará el juego generado. También le corresponde la labor más importante del analizador: analizar la instancia del DSL y sustituir los valores en la plantilla para crear el juego.
Atributos Propuestos
Dicctionary: Diccionario que contendrá como clave las etiquetas de la instancia del DSL y como valor los valores de cada etiqueta. **RutaImages**: Ruta de la carpeta de la plantilla donde serán almacenadas las imágenes. **RutaSounds**: Ruta de la carpeta de la plantilla donde serán almacenados los sonidos.
Métodos Propuestos
copyToDestiny: Método que copia todos los ficheros del juego creado con el editor a la carpeta correspondiente de la plantilla, las imágenes, sonidos y niveles. **analyzeDSL**: Método que analiza la instancia del DSL y sustituye los valores encontrados en él en las etiquetas de la plantilla.

Nombre de la Clase
Reader
Descripción
Clase utilizada para comprobar el proyecto generado por editor.
Responsabilidades
Es la clase encarga de comprobar que en el carpeta del juego creada por el editor se encuentra el fichero dsl.xml, los recursos de las imágenes y sonidos y los niveles creados.
Atributos Propuestos
rutaOrigen: Contiene la ruta donde se encuentra almacenada la plantilla. **rutaDestino**: Contiene la ruta que el usuario eligió para guardar el proyecto de Eclipse que construirá el subsistema analizador. **rutaFicheros:** Contiene la ruta donde se encuentra almacenado el proyecto del juego creado en el editor.
Métodos Propuestos
searchRoutes: Método que comprobará que las rutas que se reciben por parámetro existen y contienen los archivos necesarios para generar el juego.

5.4.2.3 Subsistema Plantilla

Nombre de la Clase
Model
Descripción
Clase que representa el objeto modelo, a partir del cual se crearán los demás objetos del juego.
Responsabilidades
Proporciona funcionalidad para el manejo de modelos.
Atributos Propuestos
gravity_speed: velocidad de la gravedad del modelo **height**: alto del modelo. **width**: ancho del modelo. **image**: imagen del modelo. **state**: estado del modelo, activo/inactivo. **x_pos**: coordenada x. **y_pos**: coordenada y. **x_speed:** velocidad del modelo en el eje x. **y_speed:** velocidad del modelo en el eje y.
Métodos Propuestos
isCollided: Método que comprobará si un modelo colisiona con otro. **onDraw**: Método que dibujará en la pantalla el modelo.

Nombre de la Clase
Collision
Descripción
Clase de implementación de modelo que representa la colisión entre dos modelos.
Responsabilidades
Proporciona funcionalidad para el manejo de las colisiones entre los objetos.
Atributos Propuestos
down_area: área inferior del rectángulo de colisión. **left_area**: área izquierda del rectángulo de colisión. **right_area**: área derecha del rectángulo de colisión. **top_area**: área superior del rectángulo de colisión.
Métodos Propuestos

Nombre de la Clase
Button
Descripción
Clase de implementación de modelo que representa un botón.
Responsabilidades
Proporciona funcionalidad para el manejo de los botones de las pantallas del juego.
Atributos Propuestos
Métodos Propuestos
isPressed: Método que detecta si un botón es presionado.

Nombre de la Clase
Character
Descripción
Clase implementación de modelo que representa un personaje principal.
Responsabilidades
Proporciona funcionalidad para el manejo del personaje principal del juego.
Atributos Propuestos
gravityPropulsion: indica si el personaje tiene propulsión.
Métodos Propuestos
onDraw: Método redefinido que dibuja el personaje en pantalla. **move**: Método que controlará el movimiento del personaje en pantalla.

Nombre de la Clase
Scoreboard
Descripción
Clase de implementación de modelo que representa un marcador.
Responsabilidades
Proporciona funcionalidad para el manejo de los marcadores que se mostrarán en la pantalla de juego.
Atributos Propuestos
fontSize: indica el tamaño de la fuente. **text**: texto que se mostrará en el marcador.
Métodos Propuestos
onDraw: Método redefinido que dibuja el marcador en pantalla.

Nombre de la Clase
finish
Descripción
Clase de implementación de modelo que representa la meta.
Responsabilidades
Proporciona funcionalidad para el manejo de la meta que encontrará al final de cada nivel.
Atributos Propuestos
Métodos Propuestos
onDraw: Método redefinido que dibuja la meta en pantalla. **move**: Método que controlará el movimiento de la meta en pantalla.

Nombre de la Clase
Background
Descripción
Clase de implementación de modelo que representa el fondo de un modelo.
Responsabilidades
Proporciona funcionalidad para el manejo de fondo de los modelos.
Atributos Propuestos
Métodos Propuestos
onDraw: Método redefinido que dibuja el fondo en pantalla.

move: Método que controlará el movimiento del fondo en pantalla.

Nombre de la Clase
Enemy
Descripción
Clase de implementación de modelo que representa un enemigo.
Responsabilidades
Proporciona funcionalidad para el manejo de los enemigos del juego.
Atributos Propuestos
givePoints: indica si el enemigo dará puntos al colisionar con él. **points**: número de puntos que da.
Métodos Propuestos
onDraw: Método redefinido que dibuja el enemigo en pantalla. **move**: Método que controlará el movimiento del enemigo en pantalla.

Nombre de la Clase
Screen
Descripción
Clase que representa el objeto pantalla.
Responsabilidades
Proporciona funcionalidad para el manejo de las pantallas.
Atributos Propuestos
background: objeto background que representa el fondo de cada pantalla.
Métodos Propuestos
onDraw: Método que dibuja una pantalla.

Nombre de la Clase
MenuView
Descripción
Clase que permite desplegar la pantalla del menú principal del juego.
Responsabilidades
Proporciona funcionalidad para el manejo de la pantalla del menú principal.
Atributos Propuestos
exitButton: objeto button que permitirá salir del juego. **startButton:** objeto button que permitirá comenzar el juego. **optionsButton:** objeto button que permitirá el acceso al menú de opciones.
Métodos Propuestos
exit: Método que saldrá de la aplicación. **start**: Método que lanzará la pantalla siguiente del juego. **options**: Método que lanzará la pantalla de opciones.

Nombre de la Clase
OptionsView
Descripción
Clase que permite desplegar la pantalla del menú de opciones del juego.
Responsabilidades
Proporciona funcionalidad para el manejo de la pantalla de opciones.
Atributos Propuestos
musicOffButton: objeto button que permitirá desactivar la música. **musicOnButton:** objeto button que permitirá activar la música.
Métodos Propuestos
musicOn: Método que activará la música. **musicOff**: Método que desactivará la música.

Nombre de la Clase
SelectLevelView
Descripción
Clase que permite desplegar la pantalla del menú de selección de niveles del juego.
Responsabilidades
Proporciona funcionalidad para el manejo de la pantalla de selección de nivel.
Atributos Propuestos
leftButton: objeto button que permitirá cambiar la miniatura del nivel actual al siguiente. **rightButton:** objeto button que permitirá cambiar la miniatura del nivel actual al anterior. **playButton**: objeto button que permitirá jugar al nivel elegido.
Métodos Propuestos
left: Método que cambiará la miniatura del nivel que se muestra en pantalla por la del nivel anterior. **right:** Método que cambiará la miniatura del nivel que se muestra en pantalla por la del nivel siguiente. **play**: Método que lanzará el nivel seleccionado para jugarlo.

Nombre de la Clase
PauseView
Descripción
Clase que permite desplegar la pantalla de pausa del juego.
Responsabilidades
Proporciona funcionalidad para el manejo de la pantalla de pausa del juego.
Atributos Propuestos
reanudeButton: objeto button que permitirá reanudar el nivel pausado. **exitButton:** objeto button que permitirá salir del nivel a la pantalla de selección de nivel.
Métodos Propuestos
reanude: Método que reanudará el nivel. **exit:** Método que saldrá del nivel a la pantalla de selección de nivel.

Nombre de la Clase
GameView
Descripción
Clase que permite desplegar la pantalla de juego.
Responsabilidades
Proporciona funcionalidad para el manejo de la pantalla de juego.
Atributos Propuestos
lLife: imagen del icono de las vidas. **lifeScoreboard**: objeto Scoreboard que mostrará y actualizará el número de vidas. **points**: imagen del icono de los puntos. **pointScoreboard:** objeto Scoreboard que mostrará y actualizará el número de puntos. **meters**: imagen del icono de los metros. **metersScoreboard:** objeto Scoreboard que mostrará y actualizará el número de metros recorridos.
Métodos Propuestos
update: Método que actualizará el valor de cada marcador.

Nombre de la Clase
loseLevelView
Descripción
Clase que permite desplegar la pantalla de derrota de nivel del juego.
Responsabilidades
Proporciona funcionalidad para el manejo de la pantalla de derrota de nivel.
Atributos Propuestos
repeatButton: objeto button que permitirá repetir el nivel jugado. **selectLevelButton:** objeto button que permitirá ir a la pantalla de selección de nivel. **points**: objeto Scoreboard que mostrará los puntos obtenidos en el nivel.
Métodos Propuestos
repeat: Método que recargará el nivel. **selectLevel:** Método que cargará la pantalla de selección de nivel.

Nombre de la Clase
winLevelView
Descripción
Clase que permite desplegar la pantalla de victoria de nivel del juego.
Responsabilidades
Proporciona funcionalidad para el manejo de la pantalla de victoria de nivel.
Atributos Propuestos
repeatButton: objeto button que permitirá repetir el nivel jugado. **nextButton:** objeto button que permitirá jugar al siguiente nivel. **points**: objeto Scoreboard que mostrará los puntos obtenidos en el nivel.
Métodos Propuestos
repeat: Método que recargará el nivel. **next:** Método que cargará el siguiente nivel.

Nombre de la Clase
endGameView
Descripción
Clase que permite desplegar la pantalla de fin de juego.
Responsabilidades
Proporciona funcionalidad para el manejo de la pantalla de fin de juego.
Atributos Propuestos
menuButton: objeto button que permitirá volver al menú principal.
Métodos Propuestos
menu: Método que cargará la pantalla principal del juego.

5.5 Análisis de Casos de Uso y Escenarios

5.5.1 Caso de Uso: Crear Videojuego

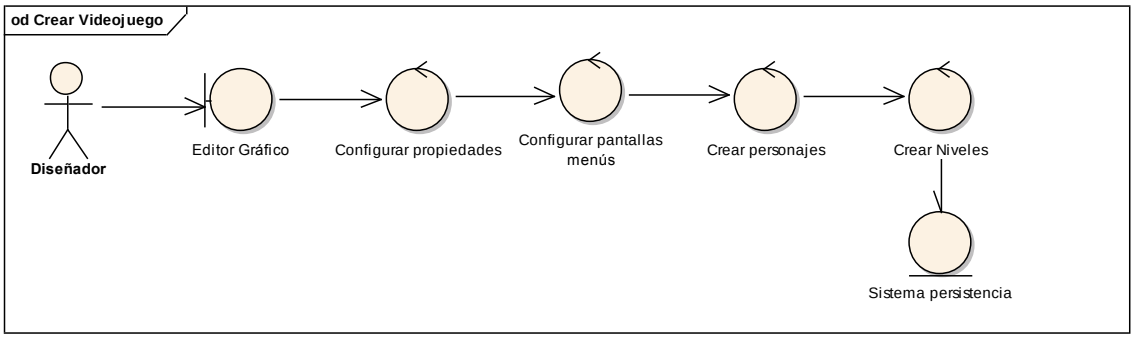


Ilustración 5.28. Descripción actividades escenario: Crear videojuego

Crear Videojuego	
Precondiciones	Ninguna.
Poscondiciones	El juego generado debe estar creado correctamente.
Actores	Iniciado y finalizado por el usuario diseñador.
Descripción	El usuario diseñador: 1. Accederá a la pantalla principal de la aplicación y escogerá la ruta donde almacenar el juego a crear. 2. Podrá configurar las propiedades globales del juego (Ver caso de uso Configurar propiedades globales). 3. Podrá configurar las propiedades específicas del juego (Ver caso de uso Configurar propiedades específicas). 4. Podrá configurar la pantalla principal del juego (Ver caso de uso Configurar pantalla principal). 5. Podrá configurar la pantalla de opciones del juego (Ver caso de uso Configurar pantalla opciones). 6. Podrá configurar la pantalla de selección de nivel del juego (Ver caso de uso Configurar pantalla de selección de nivel). 7. Podrá configurar la pantalla de carga del juego (Ver caso de uso Configurar pantalla de carga). 8. Podrá configurar la pantalla de pausa del juego (Ver caso de uso Configurar pantalla pausa). 9. Podrá configurar la pantalla de juego (Ver caso de uso Configurar pantalla juego). 10. Podrá configurar la pantalla de derrota de nivel (Ver caso de uso Configurar pantalla derrota nivel). 11. Podrá configurar la pantalla de victoria de nivel (Ver caso de uso Configurar pantalla victoria nivel). 12. Podrá configurar la pantalla fin de juego (Ver caso de uso Configurar pantalla fin juego). 13. Podrá crear el personaje principal (Ver caso de uso Crear personaje principal). 14. Podrá crear los enemigos (Ver caso de uso Crear enemigos). 15. Podrá crear la meta del juego (Ver caso de uso Crear meta). 16. Podrá crear los niveles que desee (Ver caso de uso Crear nivel). 17. Para poder generar el juego deberá crear al menos un nivel con un personaje principal, enemigos y una meta. A continuación se generará la instancia del DSL y se escogerá la ruta donde almacenar el juego generado. El analizador examinará la instancia e insertará los datos en la plantilla, creando el juego diseñado en el editor.
Variaciones (escenarios secundarios)	• Ninguno.
Excepciones	• Errores en los distintos subsistemas: No se puede completar la operación correctamente. ○ Notificar un error asociado al problema encontrado
Notas	En cada paso de creación del juego se guarda automáticamente el estado del juego en cada momento. Este se almacenará en disco para

	posteriores modificaciones.

5.5.2 Caso de Uso: Guardar Estado Videojuego

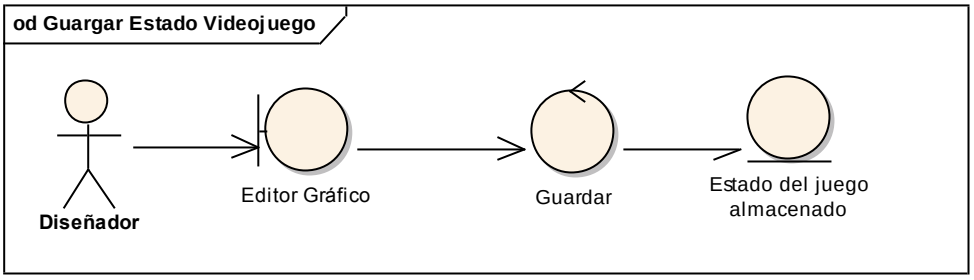


Ilustración 5.29. Descripción actividades escenario: Guardar Estado Videojuego

Guardar Videojuego	
Precondiciones	Se debe haber arrancado la aplicación previamente y haber escogido la ruta de destino.
Poscondiciones	Debe estar almacenado un archivo con el estado de cada pantalla configurada hasta el momento de realizar la acción de guardar, así como las imágenes y recursos utilizados en la creación del juego.
Actores	Iniciado y finalizado por el usuario diseñador.
Descripción	El usuario diseñador: 1. Podrá guardar el estado de creación del nuevo juego en cualquier momento mediante la opción "Guardar" del editor. También se guardará automáticamente al navegar de una pantalla de configuración a otra del editor gráfico. Se guardará en la ruta que se escogió en la pantalla principal del editor y se almacenará un archivo por cada pantalla configurada hasta el momento de guardar.
Variaciones (escenarios secundarios)	• **Guardar juego cargado previamente:** en este caso se modificarán los archivos ya existentes que contienen el estado del juego.
Excepciones	• Errores en el proceso de almacenaje del estado del juego. o Notificar un error asociado al problema encontrado
Notas	De cada pantalla se almacena un fichero binario .dat que contiene la configuración de la misma. También se almacenan los ficheros de imágenes y sonidos utilizados en cada pantalla.

5.5.3 Caso de Uso: Cargar Estado Videojuego

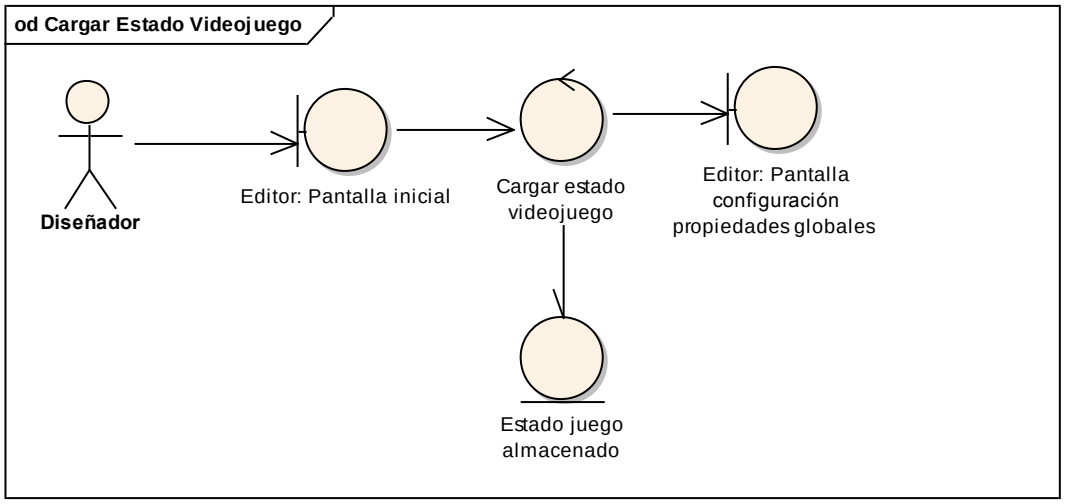


Ilustración 5.30. Descripción actividades escenario: Cargar Estado Videojuego

Cargar Videojuego	
Precondiciones	Debe estar almacenado el estado del juego que se quiere cargar.
Poscondiciones	Debe ser cargado el estado de todas las pantallas que fueron guardadas correctamente.
Actores	Iniciado y finalizado por el usuario diseñador.
Descripción	El usuario diseñador: 1. Accederá a la pantalla principal de la aplicación. 2. Escogerá el proyecto que desea cargar. 3. Continuará con el diseño del videojuego.
Variaciones (escenarios secundarios)	• Ninguno

Excepciones	• **Se intenta cargar un fichero que no es de la aplicación:** No se puede cargar el archivo.
	  ○ Notificar un error asociado al problema encontrado
	• **Error en el proceso de carga del estado de alguna de las pantallas:** No se encuentra el estado del juego como se esperaba.
	  ○ Notificar un error asociado al problema encontrado
	• **Error en el proceso de carga por falta de recursos que habían sido almacenados:** Se borró alguna de las imágenes o recursos que se almacenaron durante el proceso de almacenamiento.
	  ○ Notificar un error asociado al problema encontrado

Notas	Solamente se puede cargar el estado de un videojuego desde la pantalla principal de la aplicación.

5.5.4 Caso de Uso: Exportar Videojuego

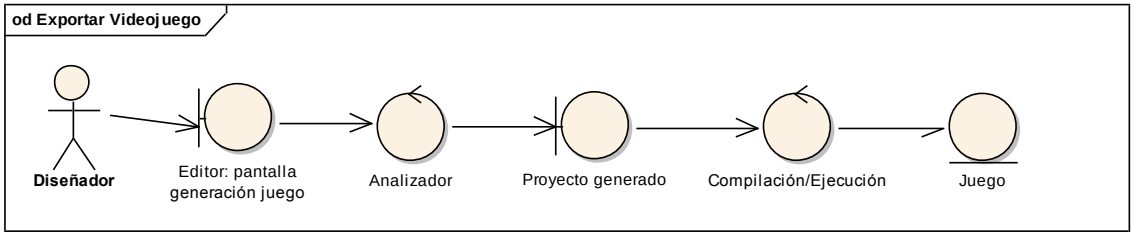


Ilustración 5.31. Descripción actividades escenario: Exportar Videojuego

Exportar Videojuego	
Precondiciones	El juego que se va a generar debe tener al menos un nivel creado, en que ha de haber un personaje principal, enemigos y una meta añadidos.
Poscondiciones	El videojuego exportado debe ser un proyecto Eclipse listo para ser importado en el entorno de desarrollo y únicamente tener que compilarlo para generar el videojuego listo para instalar en el dispositivo móvil.
Actores	Iniciado y finalizado por el usuario diseñador.
Descripción	El usuario diseñador: 1. Accederá a la pantalla de generación del juego. 2. Escogerá la ruta donde desea almacenar el proyecto generado. 3. Esperará mientras el analizador construye el proyecto. 4. Importará el proyecto generado al Eclipse. 5. Compilará el proyecto y ya obtendrá el juego listo para ser jugado.
Variaciones (escenarios secundarios)	• Antes de ir a la pantalla de generación de escenarios el usuario diseñador podrá modificar la instancia del DSL generada manualmente, y posteriormente ir a la pantalla de generación.
Excepciones	• **El analizador no hace bien la sustitución de los elementos la instancia del DSL en la plantilla:** El proyecto generado no compilará. • **El juego generado no funciona correctamente.**
Notas	Es necesario tener instalado Eclipse para poder completar el proceso de exportación del juego.

5.5.5 Caso de Uso: Crear/Editar instancia DSL

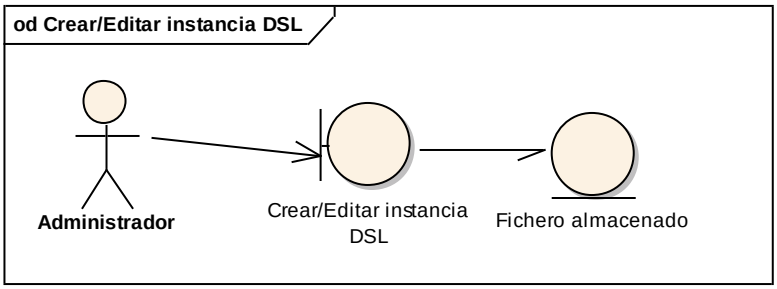


Ilustración 5.32. Descripción actividades escenario: Crear/Editar instancia DSL

Crear / Editar instancia DSL	
Precondiciones	Para modificar la instancia del DSL es necesario que este fichero exista. Para crearlo no es necesaria ninguna precondición.
Poscondiciones	El fichero de la instancia del DSL no debe contener errores.
Actores	Iniciado y Finalizado por el usuario Administrador.
Descripción	El usuario administrador: 1. Creará el fichero o modificará la instancia del DSL. 2. Lo almacenará en disco. 3. Si añadió nuevos recursos de imagen o sonido deberá incluirlos en la carpeta de imágenes del proyecto.
Variaciones (escenarios secundarios)	• El fichero se puede modificar mediante el editor gráfico.
Excepciones	• **El fichero está siendo utilizado por otra aplicación:** No se puede abrir el archivo. o Notificar un error asociado al problema encontrado • **El fichero presenta errores de formato.** o Notificar un error asociado al problema encontrado
Notas	- Si se va a crear un fichero nuevo, se deberán respetar las etiquetas existentes en la instancia, solamente se podrán asignar sus valores. - Si se va a modificar solamente se podrá modificar los valores de las etiquetas, no las etiquetas. - Para asegurarse de que el fichero se ha construido/modificado correctamente, se abrirá el fichero con un navegador, y así se mostrarán los errores, si los hubiera.

5.5.6 Caso de Uso: Crear/Editar Plantilla

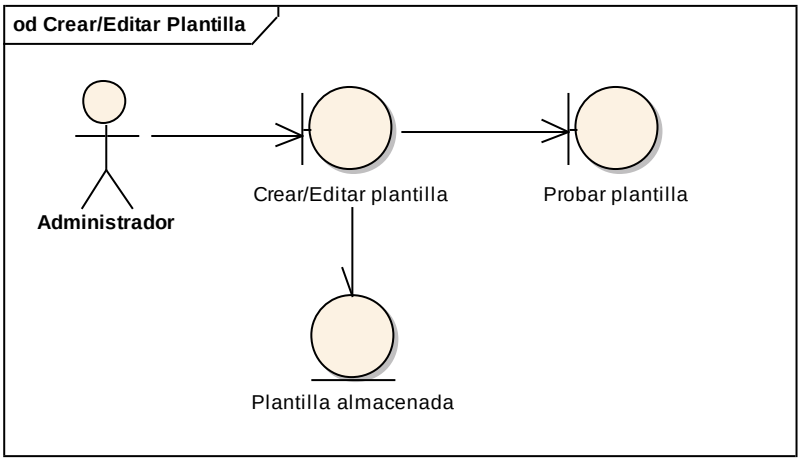


Ilustración 5.33. Descripción actividades escenario: Crear/Editar Plantilla

Crear / Editar Plantilla	
Precondiciones	Para modificar la plantilla es necesario que esta exista. Para crearla no es necesaria ninguna precondición.
Poscondiciones	La plantilla no debe contener errores.
Actores	Iniciado y Finalizado por el usuario Administrador.
Descripción	El usuario administrador: 1. Creará o modificará la plantilla. 2. La almacenará en disco. 3. Deberá generar un proyecto con esa plantilla para comprobar que esta no contiene errores. 4. Podrá incluirla en el editor para probar si funciona correctamente, o bien probarla rellenándola manualmente.
Variaciones (escenarios secundarios)	• Ninguna.
Excepciones	• **El fichero está siendo utilizado por otra aplicación:** No se puede abrir el archivo. o Notificar un error asociado al problema encontrado • **La plantilla presenta errores en algunos de los ficheros que contiene.** o Notificar un error asociado al problema encontrado
Notas	Para asegurarse de que la plantilla se ha creado/modificado correctamente, se puede hacer la sustitución de los datos con la instancia del DSL válida para esta plantilla mediante el analizador. Así se verán los errores de forma más rápida.

5.5.7 Caso de Uso: Jugar Videojuego

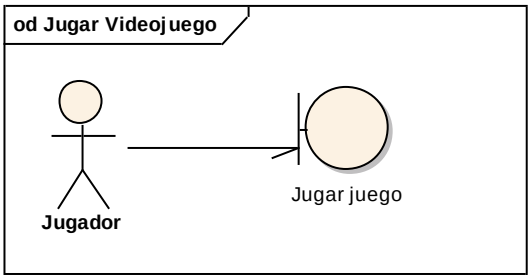


Ilustración 5.34. Descripción actividades escenario: Jugar Videojuego

Jugar Videojuego	
Precondiciones	Debe existir un juego y tuvo que ser creado por el usuario Diseñador.
Poscondiciones	El usuario jugador y/o el diseñador pueden jugar al videojuego.
Actores	Iniciado y finalizado por el usuario Jugador.
Descripción	El usuario jugador: 1. Instala el juego compilado por el usuario diseñador en el móvil y juega.
Variaciones (escenarios secundarios)	• El juego se puede jugar desde el emulador de Eclipse en vez desde el dispositivo móvil.
Excepciones	• **La aplicación no puede lanzarse:** El juego contiene errores. o Notificar un error asociado al problema encontrado
Notas	Ninguna

5.6 Análisis de Interfaces de Usuario

Siguiendo la premisa de que en el diseño de una interfaz de usuario es preciso garantizar su usabilidad, permitir manejar el programa de forma eficiente y que no genere errores, se analizará en este apartado la creada para el presente proyecto.

5.6.1 Descripción de la Interfaz

La interfaz de esta aplicación está basada en una pantalla principal en la que se explica brevemente que tipos de juegos se podrán crear con ella.

Una vez seleccionada la ruta donde almacenar el juego nuevo o donde se encuentra el juego a cargar, se abre una ventana contendedora que incluirá todas las pantallas de la interfaz de usuario. Esta ventana contenedora se puede dividir en tres zonas:

- Barra de menú
- Menú navegación, a través del cual el usuario podrá ir a la pantalla que desee sin tener que recorrer todas.
- Área de trabajo, donde se abrirán las diferentes pantallas de la interfaz.
- Botones navegación, que permitirá al usuario ir hacia adelante o hacia atrás en la interfaz.

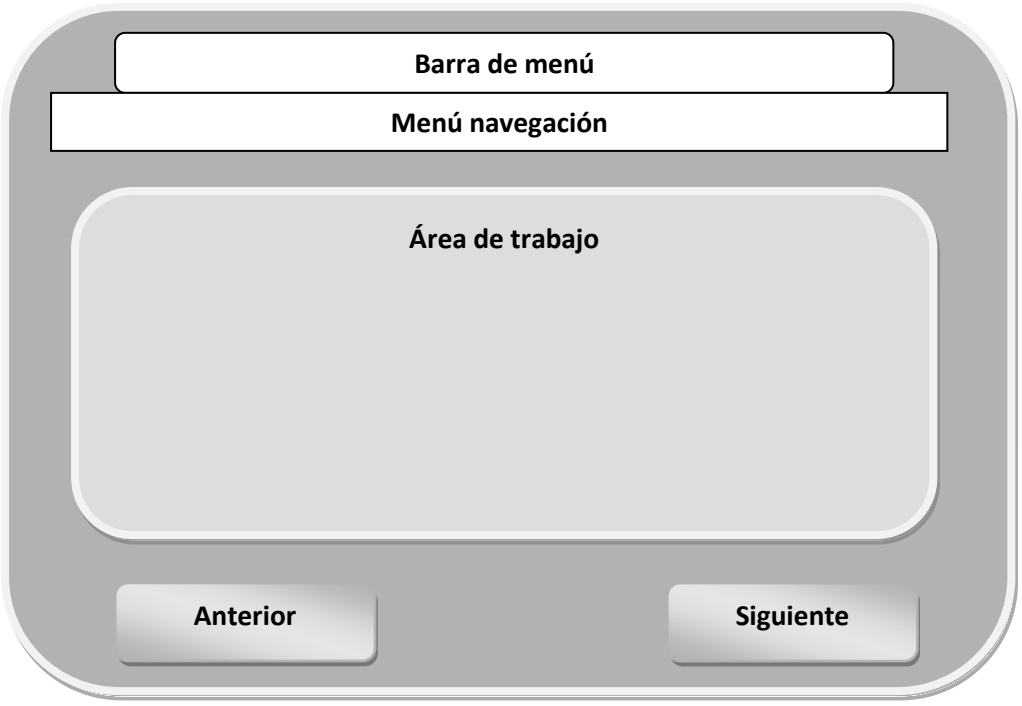


Ilustración 5.35. Ventana contenedora interfaz

5.6.1.1 Pantalla principal

Esta pantalla contará con dos botones, uno para acceder a la aplicación para crear un juego desde cero o bien cargar uno ya creado anteriormente.

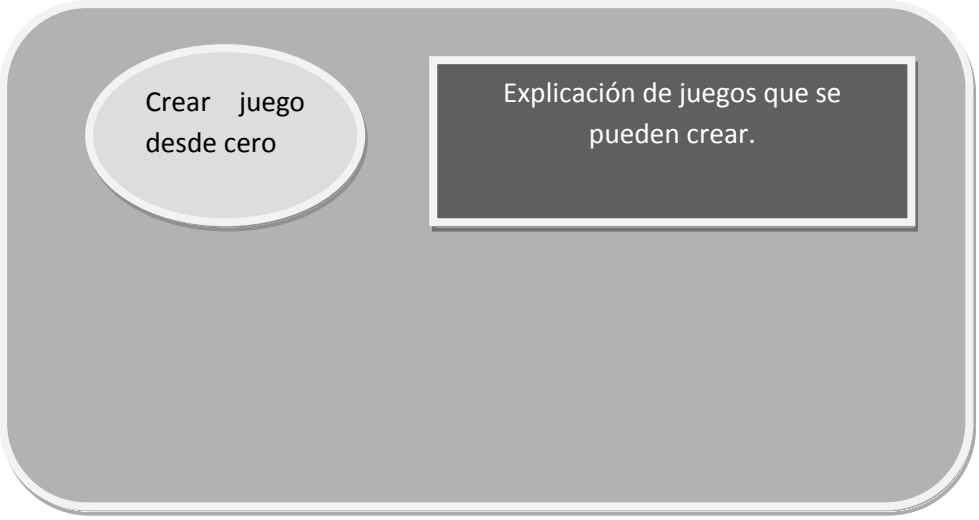


Ilustración 5.36. Análisis, pantalla principal

5.6.1.2 Pantalla configuración propiedades globales

En esta pantalla se configurarán las propiedades generales del juego. Se podrán establecer varias opciones y se indicará el orden en el que se deberá configurar. Tanto esta pantalla como las siguientes se mostrarán dentro de la ventana contenedora referida más arriba, por lo que aquí solo se indicará el diseño que tendrán las pantallas que se ubicarán en su interior.

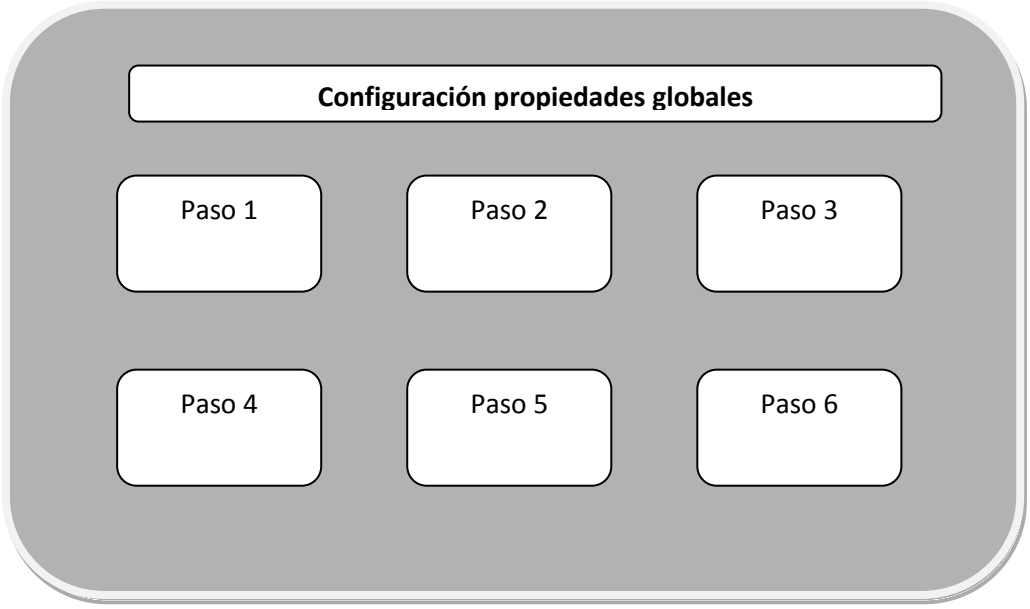


Ilustración 5.37. Análisis, pantalla configuración globales

5.6.1.3 Pantalla configuración propiedades específicas

En esta pantalla se podrá escoger el tipo de juego a crear y configurar sus propiedades específicas.

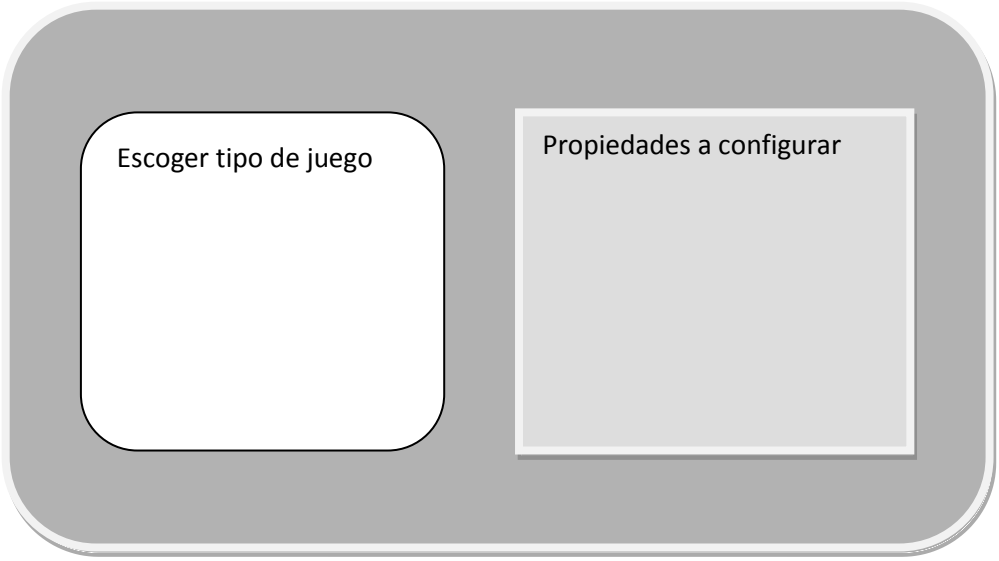


Ilustración 5.38. Análisis, pantalla configuración específicas

5.6.1.4 Pantalla configuración de los menús del juego

En esta pantalla se configurarán las de los menús del juego. Se dividirá en dos zonas:

- Zona de configuración de propiedades de los elementos que contiene cada pantalla.
- Zona de diseño: donde se encuentran los botones y fondos que contendrá cada pantalla. En cada pantalla habrá botones diferentes, ya que cada menú del juego tiene una funcionalidad distinta.

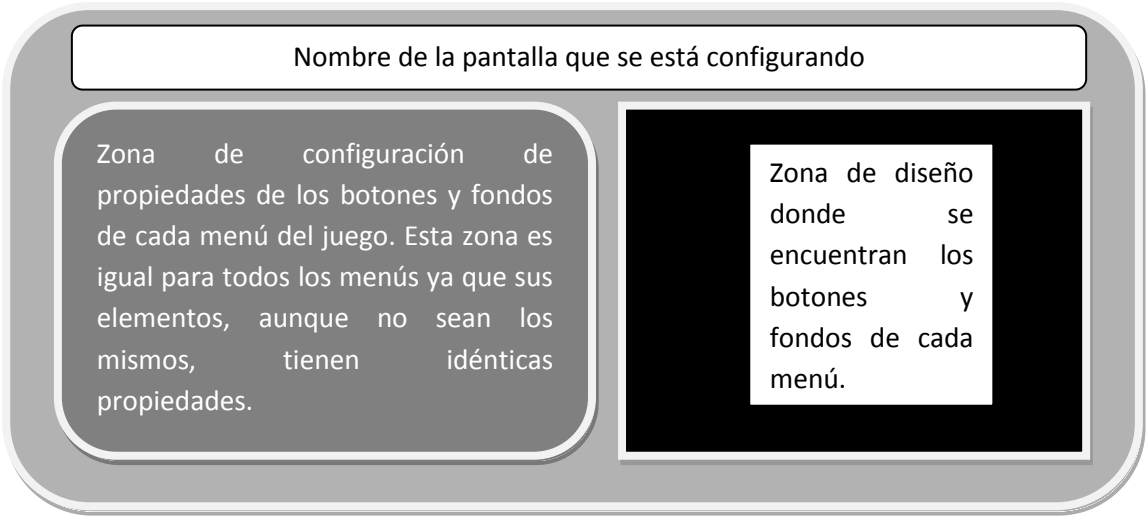


Ilustración 5.39. Análisis, pantalla configuración menús

5.6.1.5 Pantalla creación personajes, enemigos y meta

En esta pantalla crearán el personaje principal del juego, los enemigos y la meta que se encontrará al final de cada nivel. Se dividirá en 4 zonas:

- Zona de configuración de propiedades: las de cada objeto, ya que cada uno tiene unas diferentes.
- Zona de visualización de objetos creados: se mostrarán los objetos que ya se han creado.
- Zona de diseño: aquí se añadirán las imágenes de las diferentes animaciones de cada objeto.
- Zona de menú: incluirá botones para realizar diferentes acciones con cada objeto: Actualizar, Nuevo, Eliminar y Guardar.

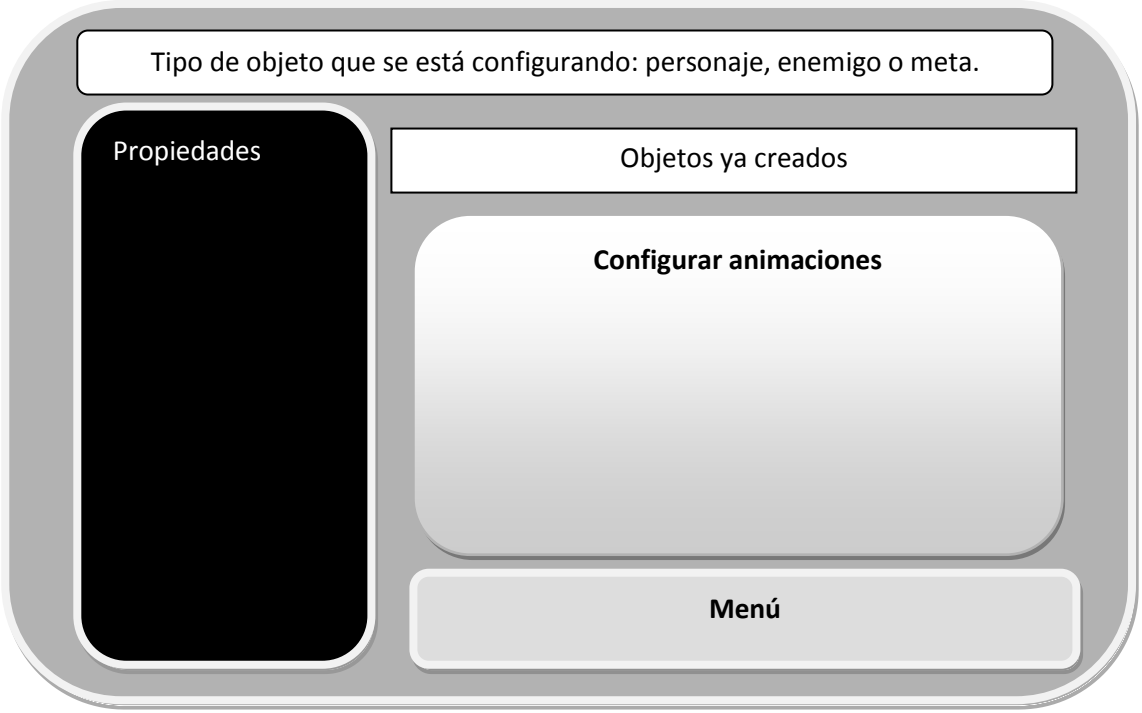


Ilustración 5.40. Análisis, pantalla creación personajes

5.6.1.6 Pantalla creación nivel

En esta pantalla de diseñaran los diferentes niveles que contendrá el juego. Para poder crear un nivel se deben haber creado los personajes con anterioridad. Esta pantalla se dividirá en varias zonas:

- Zona de personajes: se mostrarán todos los personajes, enemigos y metas que se hayan creado, para añadirlos al nivel.
- Zona de niveles: se mostrará una lista con los niveles creados.
- Zona de propiedades: se configurarán las propiedades del nivel.
- Zona de diseño: será donde se colocarán los elementos del nivel.
- Zona de menú: zona donde se encuentran las opciones que se pueden llevar a cabo como: guardar nivel, borrar nivel.

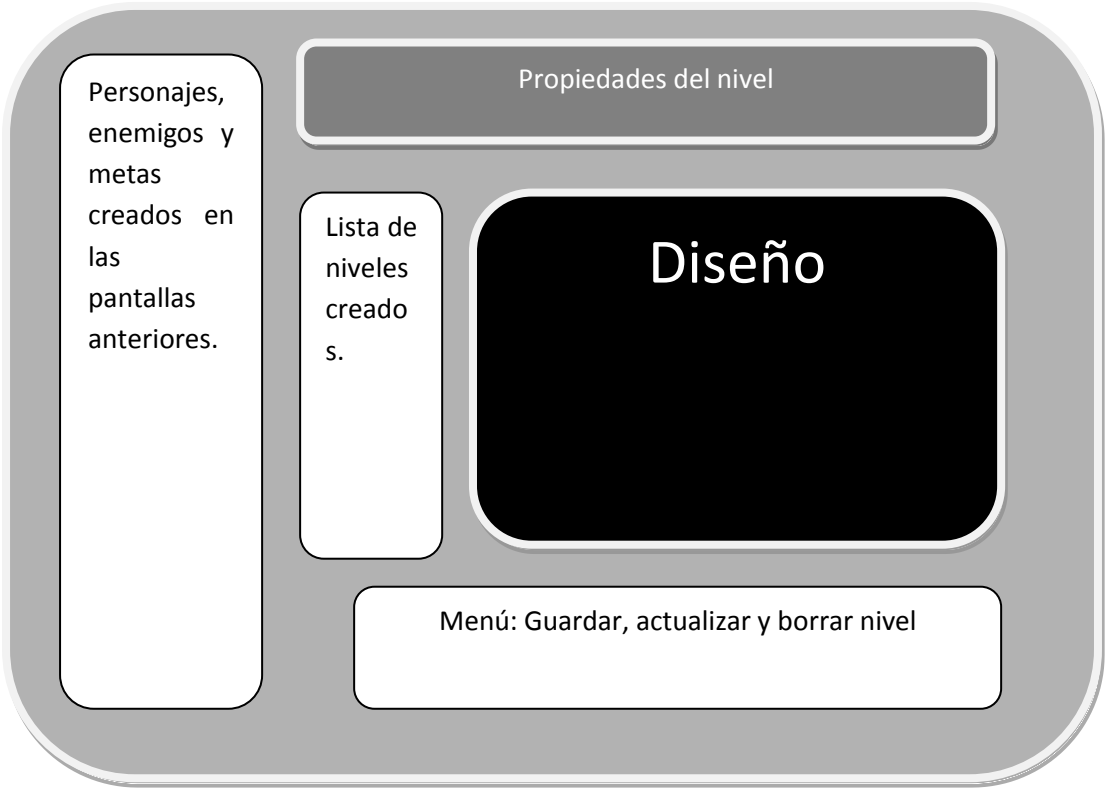


Ilustración 5.41. Análisis, pantalla creación nivel

5.6.1.7 Pantalla generación juego

Es la última pantalla de la interfaz. En ella se escogerá la ruta donde se desea almacenar el juego que se va a generar. Constará de dos zonas:

- Zona ruta: será donde el usuario escogerá la ruta donde almacenar el juego generado.
- Zona generación: donde se encontrará un botón para comenzar el proceso de generación.

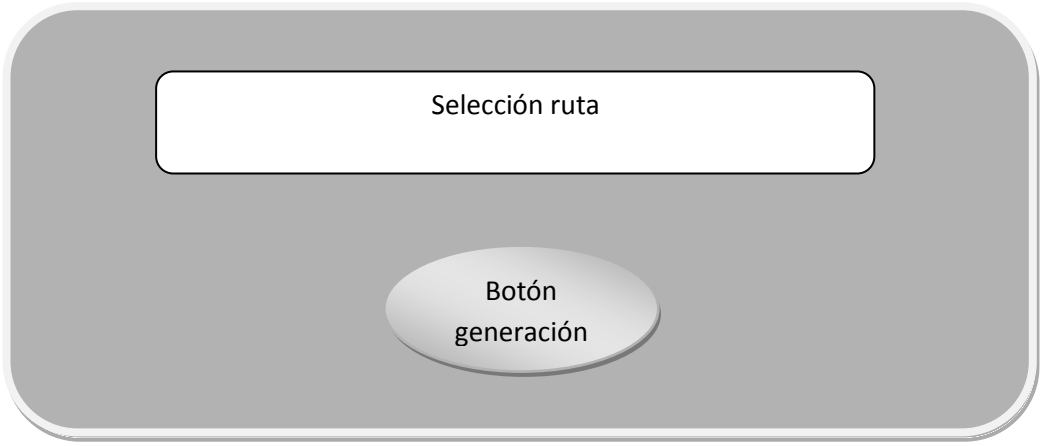


Ilustración 5.42. Análisis, pantalla generación juego

5.6.2 Descripción del Comportamiento de la Interfaz

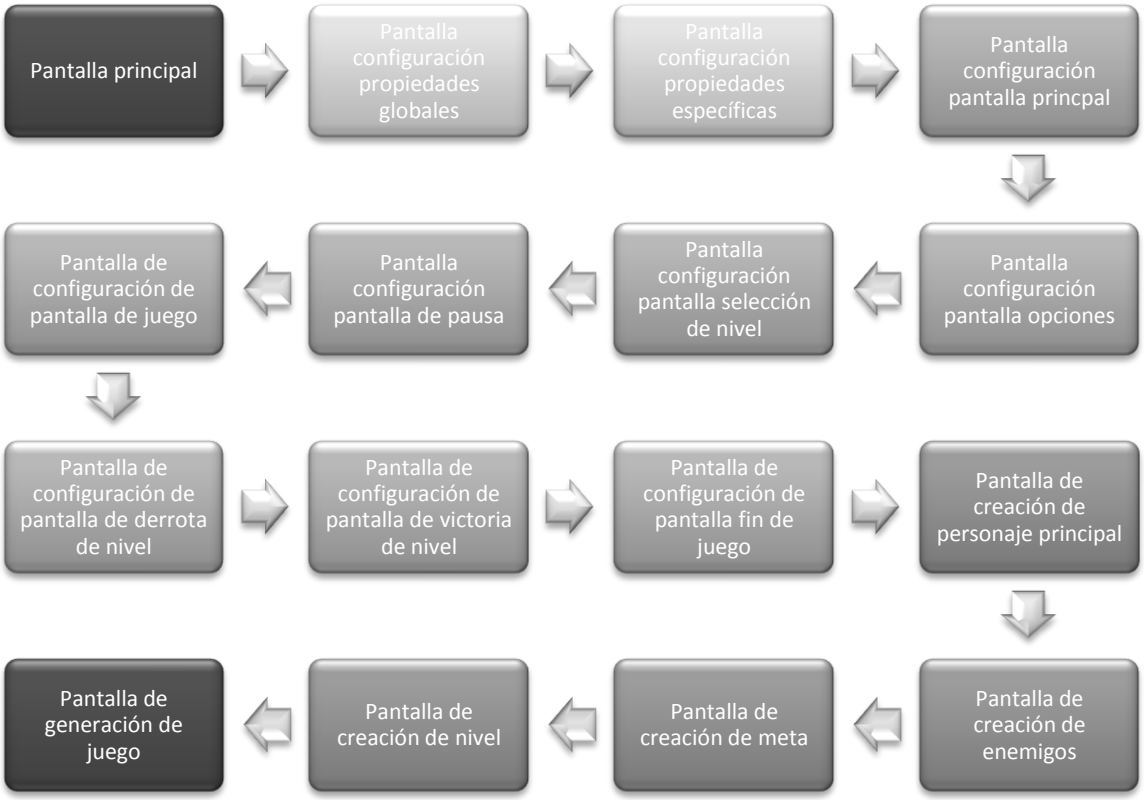
La interfaz mantendrá siempre una apariencia similar para facilitar la interacción del usuario inexperto con la aplicación y que pueda seguir el proceso de creación del juego sin dificultad ni cambios bruscos. Para ello las pantallas estarán agrupadas por tipos: primero, las de configuración de propiedades, luego las pantallas de configuración de los menús del juego, a continuación las de creación de personajes y finalmente la pantalla de creación de niveles. Así el proceso de creación del juego seguirá un orden lógico que ayudará a que el usuario no se pierda en el proceso de diseño. Se mostrarán mensajes de aviso y de error para guiar al usuario, indicándole cómo proceder en algunos casos o qué cosas debe hacer en otros.

Se ha buscado que en todas las pantallas de configuración de menús y de personajes, la zona de la configuración de propiedades esté a la izquierda y la zona de diseño a la derecha.

En lo referente a la validación de los formularios, se mostrarán mensajes de error con los campos erróneos y estos se mostrarán en color rojo, en todos los formularios.

5.6.3 Diagrama de Navegabilidad

A continuación se va a mostrar un diagrama que muestra la navegación entre las diferentes pantallas que forman la interfaz. Se puede ir para adelante y para atrás desde cualquier pantalla.



5.7 Especificación del Plan de Pruebas

En este punto del documento se va a diseñar el plan de pruebas que permitirá detectar errores en la aplicación y corregirlos, y que se aplicará durante la fase de implementación.

Se van a contemplar cuatro tipos de pruebas, que se describen a continuación.

5.7.1 Pruebas Unitarias

Una prueba unitaria es una forma de probar el correcto funcionamiento de un módulo de código, o en este caso, de una clase individual que cumple con una función concreta. Esto sirve para asegurar que cada uno de los módulos funcione correctamente por separado.

5.7.2 Pruebas de Integración

Las pruebas de integración comprenden verificaciones asociadas a grupos de componentes, comprobando que éstos funcionan correctamente cuando son ensamblados para cumplir con una función concreta.

5.7.3 Pruebas del Sistema

Son pruebas del sistema son pruebas de integración del sistema construido completo, que permiten testar el conjunto de todo el sistema una vez que se ensamblados todos los componentes.

5.7.4 Pruebas de Usabilidad

Este tipo de pruebas determinan la satisfacción del cliente con el producto final.

5.7.5 Plan de pruebas

Las pruebas se han desarrollado a partir de los casos de uso y escenarios descritos anteriormente.

5.7.5.1 Caso de uso: Crear nuevo videojuego

5.7.5.1.1 Subcaso de uso: Configurar propiedades globales

Caso de Uso 1: Crear nuevo videojuego. Subcaso: Configurar propiedades globales	
Prueba	**Resultado Esperado**
Introducir un nombre para el videojuego con tildes, paréntesis y otros caracteres especiales.	El sistema avisa al usuario de que no se pueden introducir nombres que contengan esos caracteres.
Prueba	**Resultado Esperado**
Introducir un nombre para el videojuego que solo contenga números.	Se asigna ese nombre introducido al nombre del videojuego y al paquete.
Prueba	**Resultado Esperado**
Introducir un nombre para el videojuego que solo contenga letras.	Se asigna ese nombre introducido al nombre del videojuego y al paquete.
Prueba	**Resultado Esperado**
Introducir un nombre para el videojuego que contenga letras y números.	Se asigna ese nombre introducido al nombre del videojuego y al paquete.
Prueba	**Resultado Esperado**
Se intenta modificar el nombre del paquete introduciendo tildes, paréntesis y otros caracteres especiales.	El sistema avisa al usuario de que no se puede introducir un nombre para el paquete que contenga esos caracteres.
Prueba	**Resultado Esperado**
Se intenta modificar el nombre del paquete introduciendo números y letras.	Se asigna ese nombre introducido al nombre del paquete.
Prueba	**Resultado Esperado**
Se intenta elegir un icono para el videojuego con formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger el icono tiene un filtro, así que solamente mostrará imágenes con esta extensión.
Prueba	**Resultado Esperado**
Se elige un icono para el videojuego con extensión .png.	Se mostrará una vista previa en la pantalla el icono seleccionado. Y se le asignará al videojuego.
Prueba	**Resultado Esperado**
Se marca la opción de	El sistema avisará de que se tiene que introducir un ID.

publicidad y no se añade ningún ID.	
Prueba	**Resultado Esperado**
Se marcar la opción de publicidad y se introduce un ID.	Se asigna ese valor al identificador de la publicidad
Prueba	**Resultado Esperado**
Se escoge una orientación para el videojuego.	Se asigna esa orientación al videojuego.
Prueba	**Resultado Esperado**
Se pulsa sobre el botón de reproducir sin haber introducido un sonido.	El sistema no reproducirá ningún sonido.
Prueba	**Resultado Esperado**
Se intenta elegir una melodía para los menús del juego con formato .mp4.	El sistema no permite seleccionar archivos de sonido con formato distinto a .mp3. El explorador de carpetas para escoger el icono tiene un filtro, así que solamente mostrará archivos de sonido con esta extensión.
Prueba	**Resultado Esperado**
Se selecciona un archivo de sonido en formato .mp3.	Se asigna ese sonido a la música de fondo de los menús.
Prueba	**Resultado Esperado**
Se pulsa sobre el botón de reproducir después de elegir una melodía.	El sistema reproducirá ese sonido.
Prueba	**Resultado Esperado**
Se pulsa sobre el botón de pausa después de elegir una melodía.	El sistema pausará la melodía.
Prueba	**Resultado Esperado**
Se pulsa sobre el botón de reproducir después de haber pausado la reproducción.	El sistema reanudará la reproducción.
Prueba	**Resultado Esperado**
Se pulsa sobre el botón de parar la melodía.	El sistema parará la reproducción.
Prueba	**Resultado Esperado**
Se pulsa sobre el botón de reproducir después de haber parado la reproducción.	El sistema iniciará la reproducción.
Prueba	**Resultado Esperado**
Se pulsa sobre el botón "Siguiente".	En la ruta donde se eligió que se iba a almacenar el juego, se creó una carpeta con el nombre del proyecto, y unas subcarpetas dentro llamadas "images", "sounds" y "levels"

5.7.5.1.2 Subcaso de uso: Configurar propiedades específicas

Caso de Uso 1: Crear nuevo videojuego. Subcaso: Configurar propiedades específicas	
Prueba	**Resultado Esperado**
Se elige el tipo de juego "Sigue el dedo" haciendo clic sobre la imagen correspondiente.	Este tipo de juego queda seleccionado y el resto de animaciones de los otros tipos de juegos quedan pausadas.
Prueba	**Resultado Esperado**
Se elige el tipo de juego "JetPack" haciendo clic sobre la imagen correspondiente.	Este tipo de juego queda seleccionado y el resto de animaciones de los otros tipos de juegos quedan pausadas.
Prueba	**Resultado Esperado**
Se elige el tipo de juego "Carriles" haciendo clic sobre la imagen correspondiente.	Este tipo de juego queda seleccionado y el resto de animaciones de los otros tipos de juegos quedan pausadas.
Prueba	**Resultado Esperado**
Se escoge un valor para el marcador de vidas. Si o No.	Si es escogió si el sistema registrará que el juego tendrá vidas, si escogió no, registrará que no.
Prueba	**Resultado Esperado**
Se escoge un valor para el marcador de puntos. Si o No.	Si es escogió si el sistema registrará que el juego tendrá puntos, si escogió no, registrará que no.
Prueba	**Resultado Esperado**
Se escoge un valor para el marcador de metros. Si o No.	Si es escogió si el sistema registrará que el juego tendrá metros, si escogió no, registrará que no.
Prueba	**Resultado Esperado**
Se escogerá la opción de que el juego tenga meta y que se dejen escapar a los enemigos sin restar vidas.	El sistema registrará que el juego tendrá meta y los enemigos no destruidos que se escapen de la pantalla no restarán vidas al personaje principal.
Prueba	**Resultado Esperado**
Se escogerá la opción de que el juego no tenga meta y que no se dejen escapar a los enemigos sin restar vidas.	El sistema registrará que el juego no tendrá meta y los enemigos no destruidos que se escapen de la pantalla restarán vidas al personaje principal.
Prueba	**Resultado Esperado**
Se escogerá la opción de que el juego tenga meta y que no se dejen escapar a los enemigos sin restar vidas.	El sistema registrará que el juego tendrá meta y los enemigos no destruidos que se escapen de la pantalla restarán vidas al personaje principal.
Prueba	**Resultado Esperado**
Se escogerá la opción de que el juego no tenga meta y que se dejen escapar a los enemigos sin restar vidas.	El sistema avisará de que se tiene que elegir una de las dos opciones para el final del nivel del juego.

5.7.5.1.3 Subcaso de uso: Configurar pantalla principal

Caso de Uso 1: Crear nuevo videojuego. Subcaso: Configurar pantalla principal	
Prueba	**Resultado Esperado**
Se selecciona el fondo y se intenta modificar su imagen con una en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
Prueba	**Resultado Esperado**
Se selecciona una imagen para el fondo en formato .png.	Se mostrará el fondo de la pantalla con la nueva imagen elegida.
Prueba	**Resultado Esperado**
Con la imagen de fondo seleccionada se modifica su posición X y Y.	El fondo de pantalla se desplazará automáticamente a su nueva posición.
Prueba	**Resultado Esperado**
Se selecciona uno de los botones.	El botón seleccionado aparecerá rodeado por un rectángulo de color verde.
Prueba	**Resultado Esperado**
Con un botón seleccionado se modifica su ancho y su alto.	El botón modificará su tamaño automáticamente.
Prueba	**Resultado Esperado**
Con un botón seleccionado se modifica su posición X e Y.	El botón modificará su posición en la pantalla automáticamente.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se intenta modificar su imagen con una en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen para el fondo en formato .png.	Se mostrará el botón con la nueva imagen elegida.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen de menor tamaño que la actual.	El botón cambiará su imagen y su tamaño por los de la nueva imagen.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen de mayor tamaño que la actual.	El botón cambiará su imagen y su tamaño por los de la nueva imagen.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se intenta elegir un sonido para que se reproduzca cuando se pulse el botón en el videojuego con formato	El sistema no permite seleccionar archivos de sonido con formato distinto a .mp3. El explorador de carpetas para escoger lo tiene un filtro, así que solamente mostrará archivos de sonido con esta extensión.

.mp4.	
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona un sonido para que se reproduzca cuando se pulse el botón en el videojuego en formato .mp3.	Se asigna ese sonido al botón, para que se reproduzca cuando se pulse.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se marca la opción de ocultar.	Ese botón se mostrará con una transparencia muy alta, pero sin llegar a desaparecer por si se quiere des ocultar. Este botón no aparecerá en el videojuego.
Prueba	**Resultado Esperado**
Con un botón seleccionado y que se ha ocultado previamente, se desmarca la opción de ocultar.	Se vuelve a mostrar el botón como estaba originalmente. El botón aparecerá en el juego.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se desplaza este por la pantalla.	El botón se moverá mientras se tenga pulsado el botón izquierdo del ratón. Cuando se deje de pulsar el botón dejará de moverse y quedará en esa posición. En los campos de texto de Ancho y Alto se actualizará el valor de sus coordenadas mientras se esté desplazando

5.7.5.1.4 Subcaso de uso: Configurar pantalla opciones

Caso de Uso 1: Crear nuevo videojuego. Subcaso: Configurar pantalla opciones	
Prueba	**Resultado Esperado**
Se selecciona el fondo y se intenta modificar su imagen con una en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
Prueba	**Resultado Esperado**
Se selecciona una imagen para el fondo en formato .png.	Se mostrará el fondo de la pantalla con la nueva imagen elegida.
Prueba	**Resultado Esperado**
Con la imagen de fondo seleccionada se modifica su posición X y Y.	El fondo de pantalla se desplazará automáticamente a su nueva posición.
Prueba	**Resultado Esperado**
Se selecciona uno de los botones.	El botón seleccionado aparecerá rodeado por un rectángulo de color verde.
Prueba	**Resultado Esperado**
Con un botón seleccionado se modifica su ancho y su alto.	El botón modificará su tamaño automáticamente.
Prueba	**Resultado Esperado**
Con un botón seleccionado se modifica su posición X e Y.	El botón modificará su posición en la pantalla automáticamente.

Prueba	**Resultado Esperado**
Con un botón seleccionado, se intenta modificar su imagen con una en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen para el fondo en formato .png.	Se mostrará el botón con la nueva imagen elegida.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen de menor tamaño que la actual.	El botón cambiará su imagen y su tamaño por los de la nueva imagen.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen de mayor tamaño que la actual.	El botón cambiará su imagen y su tamaño por los de la nueva imagen.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se intenta elegir un sonido para que se reproduzca cuando se pulse el botón en el videojuego con formato .mp4.	El sistema no permite seleccionar archivos de sonido con formato distinto a .mp3. El explorador de carpetas para escoger lo tiene un filtro, así que solamente mostrará archivos de sonido con esta extensión.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona un sonido para que se reproduzca cuando se pulse el botón en el videojuego en formato .mp3.	Se asigna ese sonido al botón, para que se reproduzca cuando se pulse.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se marca la opción de ocultar.	Ese botón se mostrará con una transparencia muy alta, pero sin llegar a desaparecer por si se quiere des ocultar. Este botón no aparecerá en el videojuego.
Prueba	**Resultado Esperado**
Con un botón seleccionado y que se ha ocultado previamente, se desmarca la opción de ocultar.	Se vuelve a mostrar el botón como estaba originalmente. El botón aparecerá en el juego.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se desplaza este por la pantalla.	El botón se moverá mientras se tenga pulsado el botón izquierdo del ratón. Cuando se deje de pulsar el botón dejará de moverse y quedará en esa posición. En los campos de texto de Ancho y Alto se actualizará el valor de sus coordenadas mientras se esté desplazando

5.7.5.1.5 Subcaso de uso: Configurar pantalla selección de nivel

Caso de Uso 1: Crear nuevo videojuego. Subcaso: Configurar pantalla selección de nivel	
Prueba	**Resultado Esperado**
Se selecciona el fondo y se intenta modificar su imagen con una en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
Prueba	**Resultado Esperado**
Se selecciona una imagen para el fondo en formato .png.	Se mostrará el fondo de la pantalla con la nueva imagen elegida.
Prueba	**Resultado Esperado**
Con la imagen de fondo seleccionada se modifica su posición X y Y.	El fondo de pantalla se desplazará automáticamente a su nueva posición.
Prueba	**Resultado Esperado**
Se selecciona uno de los botones.	El botón seleccionado aparecerá rodeado por un rectángulo de color verde.
Prueba	**Resultado Esperado**
Con un botón seleccionado se modifica su ancho y su alto.	El botón modificará su tamaño automáticamente.
Prueba	**Resultado Esperado**
Con un botón seleccionado se modifica su posición X e Y.	El botón modificará su posición en la pantalla automáticamente.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se intenta modificar su imagen con una en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen para el fondo en formato .png.	Se mostrará el botón con la nueva imagen elegida.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen de menor tamaño que la actual.	El botón cambiará su imagen y su tamaño por los de la nueva imagen.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen de mayor tamaño que la actual.	El botón cambiará su imagen y su tamaño por los de la nueva imagen.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se intenta elegir un sonido para que se reproduzca cuando se pulse el botón en el videojuego con formato	El sistema no permite seleccionar archivos de sonido con formato distinto a .mp3. El explorador de carpetas para escoger lo tiene un filtro, así que solamente mostrará archivos de sonido con esta extensión.

.mp4.	
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona un sonido para que se reproduzca cuando se pulse el botón en el videojuego en formato .mp3.	Se asigna ese sonido al botón, para que se reproduzca cuando se pulse.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se marca la opción de ocultar.	Ese botón se mostrará con una transparencia muy alta, pero sin llegar a desaparecer por si se quiere des ocultar. Este botón no aparecerá en el videojuego.
Prueba	**Resultado Esperado**
Con un botón seleccionado y que se ha ocultado previamente, se desmarca la opción de ocultar.	Se vuelve a mostrar el botón como estaba originalmente. El botón aparecerá en el juego.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se desplaza este por la pantalla.	El botón se moverá mientras se tenga pulsado el botón izquierdo del ratón. Cuando se deje de pulsar el botón dejará de moverse y quedará en esa posición. En los campos de texto de Ancho y Alto se actualizará el valor de sus coordenadas mientras se esté desplazando
Prueba	**Resultado Esperado**
Se selecciona la imagen de miniatura del nivel y se intenta modificar su imagen.	El sistema inhabilita el botón de añadir imagen, ya que la imagen de la miniatura del nivel solamente se podrá elegir cuando se defina el nivel.

5.7.5.1.6 Subcaso de uso: Configurar pantalla carga

Caso de Uso 1: Crear nuevo videojuego. Subcaso: Configurar pantalla carga	
Prueba	**Resultado Esperado**
Se selecciona el fondo y se intenta modificar su imagen con una en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
Prueba	**Resultado Esperado**
Se selecciona una imagen para el fondo en formato .png.	Se mostrará el fondo de la pantalla con la nueva imagen elegida.
Prueba	**Resultado Esperado**
Con la imagen de fondo seleccionada se modifica su posición X y Y.	El fondo de pantalla se desplazará automáticamente a su nueva posición.

5.7.5.1.7 Subcaso de uso: Configurar pantalla pausa juego

Caso de Uso 1: Crear nuevo videojuego. Subcaso: Configurar pantalla juego	
Prueba	**Resultado Esperado**
Se selecciona el fondo y se intenta modificar su imagen con una en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
Prueba	**Resultado Esperado**
Se selecciona una imagen para el fondo en formato .png.	Se mostrará el fondo de la pantalla con la nueva imagen elegida.
Prueba	**Resultado Esperado**
Con la imagen de fondo seleccionada se modifica su posición X y Y.	El fondo de pantalla se desplazará automáticamente a su nueva posición.
Prueba	**Resultado Esperado**
Se selecciona uno de los botones.	El botón seleccionado aparecerá rodeado por un rectángulo de color verde.
Prueba	**Resultado Esperado**
Con un botón seleccionado se modifica su ancho y su alto.	El botón modificará su tamaño automáticamente.
Prueba	**Resultado Esperado**
Con un botón seleccionado se modifica su posición X e Y.	El botón modificará su posición en la pantalla automáticamente.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se intenta modificar su imagen con una en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen para el fondo en formato .png.	Se mostrará el botón con la nueva imagen elegida.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen de menor tamaño que la actual.	El botón cambiará su imagen y su tamaño por los de la nueva imagen.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen de mayor tamaño que la actual.	El botón cambiará su imagen y su tamaño por los de la nueva imagen.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se intenta elegir un sonido para que se reproduzca cuando se pulse el botón en el videojuego con formato	El sistema no permite seleccionar archivos de sonido con formato distinto a .mp3. El explorador de carpetas para escoger lo tiene un filtro, así que solamente mostrará archivos de sonido con esta extensión.

.mp4.	
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona un sonido para que se reproduzca cuando se pulse el botón en el videojuego en formato .mp3.	Se asigna ese sonido al botón, para que se reproduzca cuando se pulse.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se marca la opción de ocultar.	Ese botón se mostrará con una transparencia muy alta, pero sin llegar a desaparecer por si se quiere des ocultar. Este botón no aparecerá en el videojuego.
Prueba	**Resultado Esperado**
Con un botón seleccionado y que se ha ocultado previamente, se desmarca la opción de ocultar.	Se vuelve a mostrar el botón como estaba originalmente. El botón aparecerá en el juego.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se desplaza este por la pantalla.	El botón se moverá mientras se tenga pulsado el botón izquierdo del ratón. Cuando se deje de pulsar el botón dejará de moverse y quedará en esa posición. En los campos de texto de Ancho y Alto se actualizará el valor de sus coordenadas mientras se esté desplazando

5.7.5.1.8 Subcaso de uso: Configurar pantalla juego

Caso de Uso 1: Crear nuevo videojuego. Subcaso: Configurar pantalla juego	
Prueba	**Resultado Esperado**
Se selecciona el fondo y se intenta modificar su imagen con una en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
Prueba	**Resultado Esperado**
Se selecciona una imagen para el fondo en formato .png.	Se mostrará el fondo de la pantalla con la nueva imagen elegida.
Prueba	**Resultado Esperado**
Con la imagen de fondo seleccionada se modifica su posición X y Y.	El fondo de pantalla se desplazará automáticamente a su nueva posición.
Prueba	**Resultado Esperado**
Se selecciona uno de los elementos de la barra de estado.	El elemento seleccionado aparecerá rodeado por un rectángulo de color verde.
Prueba	**Resultado Esperado**
Con un elemento seleccionado se modifica su ancho y su alto.	El elemento modificará su tamaño automáticamente.
Prueba	**Resultado Esperado**
Con un elemento	El elemento modificará su posición en la pantalla

seleccionado se modifica su posición X e Y.	automáticamente.
Prueba	**Resultado Esperado**
Con un elemento seleccionado, se intenta modificar su imagen con una en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión. Si se trata de un marcador, el botón de añadir imagen se deshabilitará al pulsar sobre el marcador.
Prueba	**Resultado Esperado**
Con un elemento seleccionado, se selecciona una imagen para el fondo en formato .png.	Se mostrará el elemento con la nueva imagen elegida. Si se trata de un marcador, el botón de añadir imagen se deshabilitará al pulsar sobre el marcador.
Prueba	**Resultado Esperado**
Con un elemento seleccionado, se selecciona una imagen de menor tamaño que la actual.	El elemento cambiará su imagen y su tamaño por los de la nueva imagen. Si se trata de un marcador, el botón de añadir imagen se deshabilitará al pulsar sobre el marcador.
Prueba	**Resultado Esperado**
Con un elemento seleccionado, se selecciona una imagen de mayor tamaño que la actual.	El elemento cambiará su imagen y su tamaño por los de la nueva imagen. Si se trata de un marcador, el botón de añadir imagen se deshabilitará al pulsar sobre el marcador.
Prueba	**Resultado Esperado**
Con un elemento seleccionado, se intenta elegir un sonido para que se reproduzca cuando se pulse el botón en el videojuego con formato .mp4.	El sistema no permite seleccionar archivos de sonido con formato distinto a .mp3. El explorador de carpetas para escoger lo tiene un filtro, así que solamente mostrará archivos de sonido con esta extensión.
Prueba	**Resultado Esperado**
Con un elemento seleccionado, se selecciona un sonido para que se reproduzca cuando se pulse el botón en el videojuego en formato .mp3.	Se asigna ese sonido al elemento, para que se reproduzca cuando se pulse.
Prueba	**Resultado Esperado**
Con un elemento seleccionado, se marca la opción de ocultar.	Ese elemento se mostrará con una transparencia muy alta, pero sin llegar a desaparecer por si se quiere des ocultar. Este botón no aparecerá en el videojuego.
Prueba	**Resultado Esperado**
Con un elemento seleccionado y que se ha ocultado previamente, se desmarca la opción de ocultar.	Se vuelve a mostrar el elemento como estaba originalmente. El botón aparecerá en el juego.
Prueba	**Resultado Esperado**
Con un elemento	El elemento se moverá mientras se tenga pulsado el botón

seleccionado, se desplaza este por la pantalla.	izquierdo del ratón. Cuando se deje de pulsar el botón dejará de moverse y quedará en esa posición. En los campos de texto de Ancho y Alto se actualizará el valor de sus coordenadas mientras se esté desplazando

5.7.5.1.9 Subcaso de uso: Configurar pantalla derrota nivel

Caso de Uso 1: Crear nuevo videojuego. Subcaso: Configurar pantalla derrota nivel	
Prueba	**Resultado Esperado**
Se selecciona el fondo y se intenta modificar su imagen con una en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
Prueba	**Resultado Esperado**
Se selecciona una imagen para el fondo en formato .png.	Se mostrará el fondo de la pantalla con la nueva imagen elegida.
Prueba	**Resultado Esperado**
Con la imagen de fondo seleccionada se modifica su posición X y Y.	El fondo de pantalla se desplazará automáticamente a su nueva posición.
Prueba	**Resultado Esperado**
Se selecciona uno de los botones.	El botón seleccionado aparecerá rodeado por un rectángulo de color verde.
Prueba	**Resultado Esperado**
Con un botón seleccionado se modifica su ancho y su alto.	El botón modificará su tamaño automáticamente.
Prueba	**Resultado Esperado**
Con un botón seleccionado se modifica su posición X e Y.	El botón modificará su posición en la pantalla automáticamente.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se intenta modificar su imagen con una en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen para el fondo en formato .png.	Se mostrará el botón con la nueva imagen elegida.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen de menor tamaño que la actual.	El botón cambiará su imagen y su tamaño por los de la nueva imagen.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen de mayor tamaño que la actual.	El botón cambiará su imagen y su tamaño por los de la nueva imagen.

Prueba	**Resultado Esperado**
Con un botón seleccionado, se intenta elegir un sonido para que se reproduzca cuando se pulse el botón en el videojuego con formato .mp4.	El sistema no permite seleccionar archivos de sonido con formato distinto a .mp3. El explorador de carpetas para escoger lo tiene un filtro, así que solamente mostrará archivos de sonido con esta extensión.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona un sonido para que se reproduzca cuando se pulse el botón en el videojuego en formato .mp3.	Se asigna ese sonido al botón, para que se reproduzca cuando se pulse.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se marca la opción de ocultar.	Ese botón se mostrará con una transparencia muy alta, pero sin llegar a desaparecer por si se quiere des ocultar. Este botón no aparecerá en el videojuego.
Prueba	**Resultado Esperado**
Con un botón seleccionado y que se ha ocultado previamente, se desmarca la opción de ocultar.	Se vuelve a mostrar el botón como estaba originalmente. El botón aparecerá en el juego.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se desplaza este por la pantalla.	El botón se moverá mientras se tenga pulsado el botón izquierdo del ratón. Cuando se deje de pulsar el botón dejará de moverse y quedará en esa posición. En los campos de texto de Ancho y Alto se actualizará el valor de sus coordenadas mientras se esté desplazando

5.7.5.1.10 Subcaso de uso: Configurar pantalla victoria nivel

Caso de Uso 1: Crear nuevo videojuego. Subcaso: Configurar pantalla victoria nivel	
Prueba	**Resultado Esperado**
Se selecciona el fondo y se intenta modificar su imagen con una en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
Prueba	**Resultado Esperado**
Se selecciona una imagen para el fondo en formato .png.	Se mostrará el fondo de la pantalla con la nueva imagen elegida.
Prueba	**Resultado Esperado**
Con la imagen de fondo seleccionada se modifica su posición X y Y.	El fondo de pantalla se desplazará automáticamente a su nueva posición.
Prueba	**Resultado Esperado**
Se selecciona uno de los botones.	El botón seleccionado aparecerá rodeado por un rectángulo de color verde.
Prueba	**Resultado Esperado**

Con un botón seleccionado se modifica su ancho y su alto.	El botón modificará su tamaño automáticamente.
Prueba	**Resultado Esperado**
Con un botón seleccionado se modifica su posición X e Y.	El botón modificará su posición en la pantalla automáticamente.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se intenta modificar su imagen con una en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen para el fondo en formato .png.	Se mostrará el botón con la nueva imagen elegida.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen de menor tamaño que la actual.	El botón cambiará su imagen y su tamaño por los de la nueva imagen.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen de mayor tamaño que la actual.	El botón cambiará su imagen y su tamaño por los de la nueva imagen.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se intenta elegir un sonido para que se reproduzca cuando se pulse el botón en el videojuego con formato .mp4.	El sistema no permite seleccionar archivos de sonido con formato distinto a .mp3. El explorador de carpetas para escoger lo tiene un filtro, así que solamente mostrará archivos de sonido con esta extensión.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona un sonido para que se reproduzca cuando se pulse el botón en el videojuego en formato .mp3.	Se asigna ese sonido al botón, para que se reproduzca cuando se pulse.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se marca la opción de ocultar.	Ese botón se mostrará con una transparencia muy alta, pero sin llegar a desaparecer por si se quiere des ocultar. Este botón no aparecerá en el videojuego.
Prueba	**Resultado Esperado**
Con un botón seleccionado y que se ha ocultado previamente, se desmarca la opción de ocultar.	Se vuelve a mostrar el botón como estaba originalmente. El botón aparecerá en el juego.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se desplaza este por la pantalla.	El botón se moverá mientras se tenga pulsado el botón izquierdo del ratón. Cuando se deje de pulsar el botón dejará de moverse y quedará en esa posición. En los campos de

	texto de Ancho y Alto se actualizará el valor de sus coordenadas mientras se esté desplazando

5.7.5.1.11 Subcaso de uso: Configurar pantalla fin juego

Caso de Uso 1: Crear nuevo videojuego. Subcaso: Configurar pantalla fin de juego	
Prueba	**Resultado Esperado**
Se selecciona el fondo y se intenta modificar su imagen con una en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
Prueba	**Resultado Esperado**
Se selecciona una imagen para el fondo en formato .png.	Se mostrará el fondo de la pantalla con la nueva imagen elegida.
Prueba	**Resultado Esperado**
Con la imagen de fondo seleccionada se modifica su posición X y Y.	El fondo de pantalla se desplazará automáticamente a su nueva posición.
Prueba	**Resultado Esperado**
Se selecciona uno de los botones.	El botón seleccionado aparecerá rodeado por un rectángulo de color verde.
Prueba	**Resultado Esperado**
Con un botón seleccionado se modifica su ancho y su alto.	El botón modificará su tamaño automáticamente.
Prueba	**Resultado Esperado**
Con un botón seleccionado se modifica su posición X e Y.	El botón modificará su posición en la pantalla automáticamente.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se intenta modificar su imagen con una en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen para el fondo en formato .png.	Se mostrará el botón con la nueva imagen elegida.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen de menor tamaño que la actual.	El botón cambiará su imagen y su tamaño por los de la nueva imagen.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen de mayor tamaño que la actual.	El botón cambiará su imagen y su tamaño por los de la nueva imagen.
Prueba	**Resultado Esperado**
Con un botón seleccionado,	El sistema no permite seleccionar archivos de sonido con

se intenta elegir un sonido para que se reproduzca cuando se pulse el botón en el videojuego con formato .mp4.	formato distinto a .mp3. El explorador de carpetas para escoger lo tiene un filtro, así que solamente mostrará archivos de sonido con esta extensión.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona un sonido para que se reproduzca cuando se pulse el botón en el videojuego en formato .mp3.	Se asigna ese sonido al botón, para que se reproduzca cuando se pulse.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se marca la opción de ocultar.	Ese botón se mostrará con una transparencia muy alta, pero sin llegar a desaparecer por si se quiere des ocultar. Este botón no aparecerá en el videojuego.
Prueba	**Resultado Esperado**
Con un botón seleccionado y que se ha ocultado previamente, se desmarca la opción de ocultar.	Se vuelve a mostrar el botón como estaba originalmente. El botón aparecerá en el juego.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se desplaza este por la pantalla.	El botón se moverá mientras se tenga pulsado el botón izquierdo del ratón. Cuando se deje de pulsar el botón dejará de moverse y quedará en esa posición. En los campos de texto de Ancho y Alto se actualizará el valor de sus coordenadas mientras se esté desplazando

5.7.5.1.12 Subcaso de uso: Crear personaje principal

Caso de Uso 1: Crear nuevo videojuego. Subcaso: Crear personaje principal	
Prueba	**Resultado Esperado**
Se pulsa sobre el botón "Añadir a biblioteca" sin haber creado ningún personaje.	El botón estará deshabilitado y no se podrá añadir.
Prueba	**Resultado Esperado**
Se intenta añadir una imagen a cada animación en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
Prueba	**Resultado Esperado**
Se selecciona una imagen para cada animación para el personaje en formato .png.	Se mostrará en pantalla la imagen elegida y en la vista previa se mostrará la ejecución de la animación
Prueba	**Resultado Esperado**
Se modifica el número de frames por uno mayor al número de animaciones que tiene la imagen.	En la vista previa se mostrará la ejecución de la animación y se verá que no es correcta porque no se ha introducido el número de frames adecuado.

Prueba	**Resultado Esperado**
Se modifica el número de frames por uno menos al número de animaciones que tiene la imagen.	En la vista previa se mostrará la ejecución de la animación y se verá que no es correcta porque no se ha introducido el número de frames adecuado.
Prueba	**Resultado Esperado**
Se modifica el número de frames por el número de animaciones que tiene la imagen.	En la vista previa se mostrará la ejecución de la animación y se verá que es correcta.
Prueba	**Resultado Esperado**
Se modificarán las propiedades en las que se tenga que introducir un valor manualmente, introduciendo letras en lugar de números.	El sistema avisará de que se han introducido valores erróneos.
Prueba	**Resultado Esperado**
Se modifica la velocidad del personaje mediante el desplegable. Se prueba con valores negativos, positivos y el 0.	Se ilumina la flecha correspondiente a la velocidad elegida. Al modificar la velocidad se ilumina automáticamente la flecha que corresponde.
Prueba	**Resultado Esperado**
Se intenta asignar un sonido al personaje que no está en formato .mp3.	El sistema no permite seleccionar archivos de sonido con formato distinto a .mp3. El explorador de carpetas para escoger lo tiene un filtro, así que solamente mostrará archivos de sonido con esta extensión.
Prueba	**Resultado Esperado**
Se selecciona un archivo de sonido .mp3 para cada sonido del personaje.	Los sonidos se cargan y almacenan correctamente.
Prueba	**Resultado Esperado**
Se pulsa sobre el botón "Borrar".	Si aún no se ha añadido a la biblioteca el personaje principal creado, no se podrá borrar. Si ya se ha añadido se borrará correctamente.
Prueba	**Resultado Esperado**
Se pulsa sobre el botón "Añadir a biblioteca" con un personaje creado.	El personaje se almacenará correctamente y se añadirá a la lista que hay en la parte superior de la pantalla.
Prueba	**Resultado Esperado**
Se crea otro personaje principal y se pulsa sobre "Añadir a biblioteca".	Se almacenará y se añadirá a lista que se encuentra situada en la parte superior de la pantalla.
Prueba	**Resultado Esperado**
Se hace clic sobre uno de los personajes creados.	Se cargarán en pantalla sus animaciones y el valor de sus propiedades.
Prueba	**Resultado Esperado**
Con un personaje seleccionado, se modifica alguna de sus propiedades y	Se guardarán los cambios realizados y se mostrarán en pantalla.

alguna de sus animaciones, y se hace clic en "Actualizar".	

5.7.5.1.13 Subcaso de uso: Crear enemigos

Caso de Uso 1: Crear nuevo videojuego. Subcaso: Crear enemigos	
Prueba	**Resultado Esperado**
Se pulsa sobre el botón "Añadir a biblioteca" sin haber creado ningún enemigo.	El botón estará deshabilitado y no se podrá añadir.
Prueba	**Resultado Esperado**
Se intenta añadir una imagen a cada animación en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
Prueba	**Resultado Esperado**
Se selecciona una imagen para cada animación del enemigo en formato .png.	Se mostrará en pantalla la imagen elegida y en la vista previa se mostrará la ejecución de la animación
Prueba	**Resultado Esperado**
Se modifica el número de frames por uno mayor al número de animaciones que tiene la imagen.	En la vista previa se mostrará la ejecución de la animación y se verá que no es correcta porque no se ha introducido el número de frames adecuado.
Prueba	**Resultado Esperado**
Se modifica el número de frames por uno menos al número de animaciones que tiene la imagen.	En la vista previa se mostrará la ejecución de la animación y se verá que no es correcta porque no se ha introducido el número de frames adecuado.
Prueba	**Resultado Esperado**
Se modifica el número de frames por el número de animaciones que tiene la imagen.	En la vista previa se mostrará la ejecución de la animación y se verá que es correcta.
Prueba	**Resultado Esperado**
Se modificarán las propiedades en las que se tenga que introducir un valor manualmente, introduciendo letras en lugar de números.	El sistema avisará de que se han introducido valores erróneos.
Prueba	**Resultado Esperado**
Se modifica la velocidad del enemigo mediante el desplegable. Se prueba con valores negativos, positivos y el 0.	Se ilumina la flecha correspondiente a la velocidad elegida. Al modificar la velocidad se ilumina automáticamente la flecha que corresponde.
Prueba	**Resultado Esperado**
Se intenta asignar un sonido	El sistema no permite seleccionar archivos de sonido con

al enemigo que no está en formato .mp3.	formato distinto a .mp3. El explorador de carpetas para escoger lo tiene un filtro, así que solamente mostrará archivos de sonido con esta extensión.
Prueba	**Resultado Esperado**
Se selecciona un archivo de sonido .mp3 para cada sonido del enemigo.	Los sonidos se cargan y almacenan correctamente.
Prueba	**Resultado Esperado**
Se pulsa sobre el botón "Borrar".	Si aún no se ha añadido a la biblioteca el enemigo creado, no se podrá borrar. Si ya se ha añadido se borrará correctamente.
Prueba	**Resultado Esperado**
Se pulsa sobre el botón "Añadir a biblioteca" con un enemigo creado.	El enemigo se almacenará correctamente y se añadirá a la lista que hay en la parte superior de la pantalla.
Prueba	**Resultado Esperado**
Se crea otro enemigo principal y se pulsa sobre "Añadir a biblioteca".	Se almacenará y se añadirá a lista que se encuentra situada en la parte superior de la pantalla.
Prueba	**Resultado Esperado**
Se hace clic sobre uno de los enemigos creados.	Se cargarán en pantalla sus animaciones y el valor de sus propiedades.
Prueba	**Resultado Esperado**
Con un enemigo seleccionado, se modifica alguna de sus propiedades y alguna de sus animaciones, y se hace clic en "Actualizar".	Se guardarán los cambios realizados y se mostrarán en pantalla.

5.7.5.1.14 Subcaso de uso: Crear meta

Caso de Uso 1: Crear nuevo videojuego. Subcaso: Crear meta	
Prueba	**Resultado Esperado**
Se pulsa sobre el botón "Añadir a biblioteca" sin haber creado ninguna meta.	El botón estará deshabilitado y no se podrá añadir.
Prueba	**Resultado Esperado**
Se intenta añadir una imagen a la animación en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
Prueba	**Resultado Esperado**
Se selecciona una imagen para la animación de la meta en formato .png.	Se mostrará en pantalla la imagen elegida y en la vista previa se mostrará la ejecución de la animación
Prueba	**Resultado Esperado**
Se modificarán las propiedades en las que se tenga que introducir un valor manualmente, introduciendo	El sistema avisará de que se han introducido valores erróneos.

letras en lugar de números.	
Prueba	**Resultado Esperado**
Se intenta asignar un sonido a la meta que no está en formato .mp3.	El sistema no permite seleccionar archivos de sonido con formato distinto a .mp3. El explorador de carpetas para escoger lo tiene un filtro, así que solamente mostrará archivos de sonido con esta extensión.
Prueba	**Resultado Esperado**
Se selecciona un archivo de sonido .mp3 para el sonido de la meta.	Los sonidos se cargan y almacenan correctamente.
Prueba	**Resultado Esperado**
Se pulsa sobre el botón "Borrar".	Si aún no se ha añadido a la biblioteca la meta creada, no se podrá borrar. Si ya se ha añadido se borrará correctamente.
Prueba	**Resultado Esperado**
Se pulsa sobre el botón "Añadir a biblioteca" con una meta creada.	La meta se almacenará correctamente y se añadirá a la lista que hay en la parte superior de la pantalla.
Prueba	**Resultado Esperado**
Se crea otra meta y se pulsa sobre "Añadir a biblioteca".	Se almacenará y se añadirá a lista que se encuentra situada en la parte superior de la pantalla.
Prueba	**Resultado Esperado**
Se hace clic sobre una de las metas creadas.	Se cargarán en pantalla sus animaciones y el valor de sus propiedades.
Prueba	**Resultado Esperado**
Con una meta seleccionada, se modifica alguna de sus propiedades y la imagen de la animación, y se hace clic en "Actualizar".	Se guardarán los cambios realizados y se mostrarán en pantalla.

5.7.5.1.15 Subcaso de uso: Crear nivel

Caso de Uso 1: Crear nuevo videojuego. Subcaso: Crear nivel	
Prueba	**Resultado Esperado**
Se crea un escenario vacío.	No se puede crear el escenario y el sistema avisa que es necesario añadir un personaje principal y una meta.
Prueba	**Resultado Esperado**
Se añaden elementos a un nivel.	Los elementos quedan añadidos en el escenario.
Prueba	**Resultado Esperado**
Se borra un elemento del nivel.	El elemento se elimina del escenario.
Prueba	**Resultado Esperado**
Se configuran las propiedades del nivel.	Se asignan los valores introducidos al nivel.
Prueba	**Resultado Esperado**
Se añade una imagen de descripción y una miniatura	Se añaden esas imágenes al nivel.

del nivel.	
Prueba	**Resultado Esperado**
Se modifican los carriles del nivel.	Se modifica la posición Y de los carriles.
Prueba	**Resultado Esperado**
Se añade un fondo al nivel.	Se muestra la imagen en el escenario.
Prueba	**Resultado Esperado**
Se añade un fondo multicapa al nivel.	Se muestran las imágenes en el escenario.
Prueba	**Resultado Esperado**
Se prueba a aumentar el escenario por la derecha y por la izquierda.	El escenario crece por la derecha e izquierda respectivamente.
Prueba	**Resultado Esperado**
Se guardan los cambios realizados al nivel pulsando sobre "Guardar nivel".	Se guardan los cambios en el nivel.
Prueba	**Resultado Esperado**
Se añade un nuevo nivel.	Si no se han guardado los cambios introducidos en el nivel anterior, el sistema mostrará un aviso informando de que es necesario guardar los cambios. Si se han guardado, se creará el nuevo el nivel con un número siguiente al anterior y se limpiará el escenario.
Prueba	**Resultado Esperado**
Se hace clic sobre alguno de los niveles ya creados.	Se cargará el escenario y las propiedades de ese nivel.

5.7.5.2 Caso de uso: Guardar estado videojuego

Caso de Uso 2: Guardar estado videojuego	
Prueba	**Resultado Esperado**
Se navega entre pantallas modificando alguna de sus propiedades o bien dejando los valores por defecto.	En la carpeta donde se está almacenando el juego se va creando un archivo .dat por cada pantalla que se configura y se pulsa sobre "Siguiente". Si ya existe se actualiza.
Prueba	**Resultado Esperado**
Se pulsa sobre la opción "Guardar" del menú.	En la carpeta donde se está almacenando el juego se va creando un archivo .dat por cada pantalla en la que se pulsó sobre "Guardar". Si ya existe se actualiza.
Prueba	**Resultado Esperado**
Se recorren todas las pantallas de la interfaz.	Deberá existir un archivo .dat por cada pantalla de la interfaz.

5.7.5.3 Caso de uso: Cargar estado videojuego

Caso de Uso 3: Cargar estado videojuego	
Prueba	**Resultado Esperado**
En la pantalla inicial, se elige un directorio que contiene un juego ya creado con la interfaz.	Se cargará el videojuego correctamente. Se cargará el estado de todas sus pantallas.
Prueba	**Resultado Esperado**
En la interfaz, se pulsa "Anterior" después de haber pulsado "Siguiente" en cada pantalla.	Se muestra el último estado almacenado de cada pantalla.
Prueba	**Resultado Esperado**
Se carga un videojuego con errores, como por ejemplo, faltan recursos.	El sistema mostrará un error avisando del fallo producido.

5.7.5.4 Caso de uso: Exportar videojuego

Caso de Uso 4: Exportar videojuego	
Prueba	**Resultado Esperado**
Se finaliza el proceso de crear videojuego y se llega a la pantalla se selección de ruta de despliegue.	Se muestra la pantalla de selección de ruta de despliegue.
Prueba	**Resultado Esperado**
Se escoge un directorio donde almacenar el proyecto del videojuego.	Se abre un cuadro de diálogo y se escoge un directorio.
Prueba	**Resultado Esperado**
Se pulsa sobre el botón "Generar videojuego".	Se mostrará un mensaje de que se está generando el videojuego y cuando termine el proceso, el sistema avisará de que se generó correctamente.
Prueba	**Resultado Esperado**
Se va al directorio donde se indicó que se iba a almacenar el proyecto.	Existe una carpeta llamada "Android_touch_with_character" que contiene el proyecto generado.
Prueba	**Resultado Esperado**
En ese directorio se abre la carpeta "drawable","sounds" y "levels".	Cada una de ellas contiene los ficheros de imágenes, de sonidos y de los niveles respectivamente.
Prueba	**Resultado Esperado**
Se importa el proyecto generado al entorno de desarrollo Eclipse.	Al importarlo no da errores de ningún tipo.
Prueba	**Resultado Esperado**
Se compila el proyecto.	Se compila correctamente y se obtiene la apk lista para instalar.

5.7.5.5 Caso de uso: Crear/Editar instancia DSL

Caso de Uso 5: Crear/Editar instancia DSL	
Prueba	**Resultado Esperado**
Crear archivo.	Se crea correctamente.
Prueba	**Resultado Esperado**
Modificar archivo.	Se modifica correctamente.
Prueba	**Resultado Esperado**
Se crear con fallos de formato.	Al abrirlo con un navegador se mostrarán los errores.
Prueba	**Resultado Esperado**
Se guarda con extensión .txt.	No es correcto, deberá ser .xml.
Prueba	**Resultado Esperado**
Se guarda con extensión .xml	Es correcto. Queda almacenado en disco.

5.7.5.6 Caso de uso: Crear/Editar Plantilla

Caso de Uso 6: Crear/Editar Plantilla	
Prueba	**Resultado Esperado**
Se crea una plantilla nueva.	La plantilla creada sigue la estructura de carpetas correcta.
Prueba	**Resultado Esperado**
Se sustituyen las partes variables con datos correctos.	La plantilla se convertirá en un proyecto funcionará correctamente. No dará errores ni de compilación ni de ejecución.
Prueba	**Resultado Esperado**
Se sustituyen las partes variables con datos incorrectos y se borrarán algunos recursos.	El proyecto generado será incorrecto no compilará. Se avisará de los fallos que contiene.
Prueba	**Resultado Esperado**
Se modifica una plantilla ya existente.	La plantilla modificada sigue la estructura de carpeta correcta.
Prueba	**Resultado Esperado**
Se sustituyen las partes variables con datos correctos.	La plantilla se convertirá en un proyecto funcionará correctamente. No dará errores ni de compilación ni de ejecución.
Prueba	**Resultado Esperado**
Se sustituyen las partes variables con datos incorrectos y se borrarán algunos recursos.	El proyecto generado será incorrecto no compilará. Se avisará de los fallos que contiene.
Prueba	**Resultado Esperado**
Se compila el proyecto creado o modificado, que se obtiene después de sustituir todas las partes variables por valores correctos, y se ejecuta.	Se mostrará un juego que funciona correctamente.

5.7.5.7 Caso de uso: Jugar videojuego

Caso de Uso 7: Jugar videojuego	
Prueba	**Resultado Esperado**
Se abre la aplicación del videojuego en un dispositivo móvil.	El juego funciona correctamente y se puede jugar.
Prueba	**Resultado Esperado**
Se abre la aplicación del videojuego en el emulador del Eclipse.	El juego funciona correctamente y se puede jugar.
Prueba	**Resultado Esperado**
Se prueba a jugar todos los niveles.	Todos funcionan correctamente.
Prueba	**Resultado Esperado**
Se pierde un nivel.	Se pierde y se muestra la pantalla de derrota.
Prueba	**Resultado Esperado**
Se gana un nivel.	Se gana y se muestra la pantalla de victoria.
Prueba	**Resultado Esperado**
Se terminan todos los niveles con éxito.	Se termina el videojuego y se muestra la pantalla de fin.

Capítulo 6. Diseño del Sistema

6.1 Arquitectura del Sistema

6.1.1 Diagramas de Paquetes

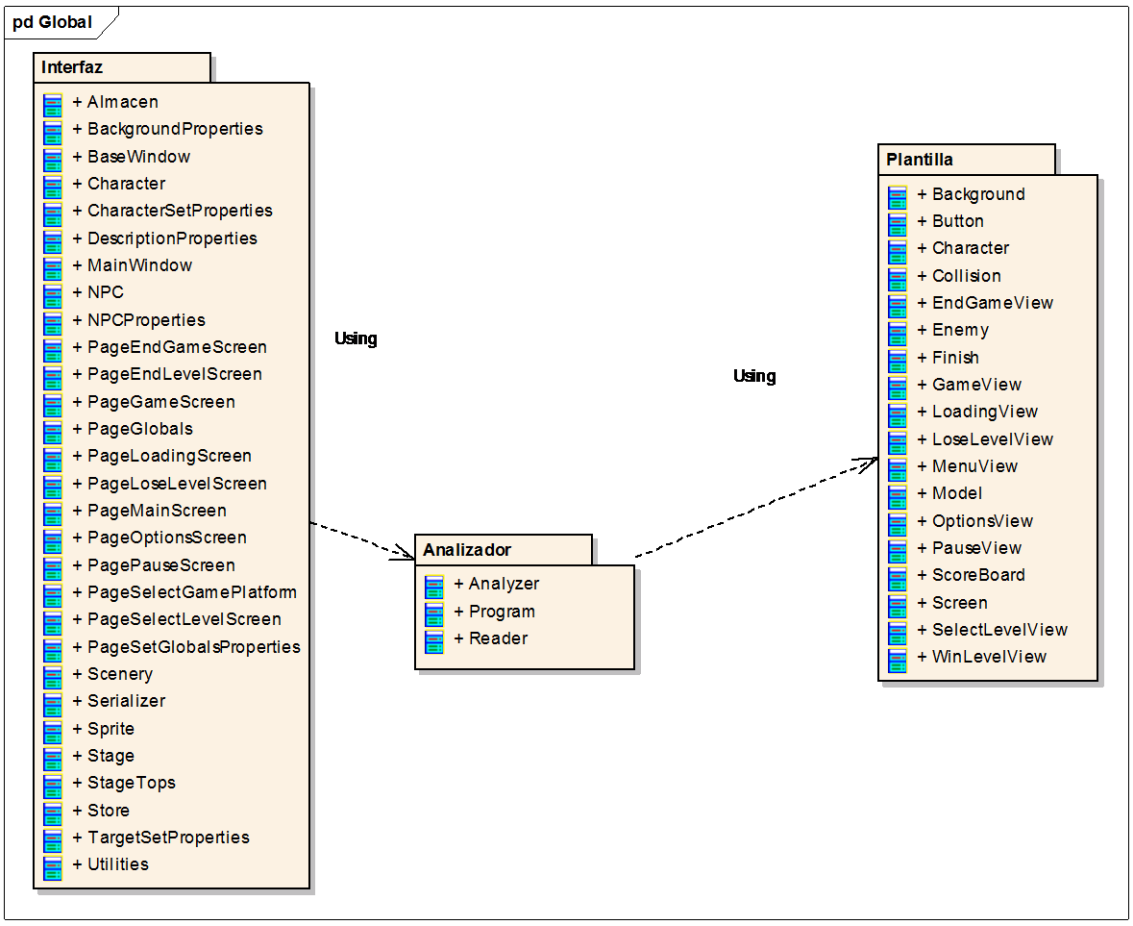


Ilustración 6.1. Diagrama paquetes Global

6.1.1.1 Paquete Interfaz

En este paquete se agrupan las clases que realizan la funcionalidad de la interfaz. La cual está provista de una serie de clases que hacen posible la interacción con el usuario.

Contiene clases que representan las propiedades del juego, los diferentes menús que contiene un juego y los diferentes elementos que formarán parte el juego: personaje principal, enemigos, meta y escenario.

Hay otras clases que se encargan, por ejemplo, de almacenar los recursos en disco y serializar los diferentes objetos del juego para guardar su estado en cada pantalla. Esta serialización permitirá también cargar su estado posteriormente.

6.1.1.2 Paquete Analizador

Este paquete contiene las clases que transforman los datos obtenidos en la interfaz, y que serán enviados al Paquete Plantilla.

Una de las clases analiza el fichero de la instancia del DSL, los recursos de imagen y sonido y los niveles obtenidos de la interfaz. Una vez comprobado que todos los archivos que se necesitan para la generación del juego existen, otra de las clases hace la sustitución de los valores obtenidos de la instancia del DSL en las etiquetas correspondientes existentes en la plantilla. Y así se obtiene el proyecto que contiene el juego que se va a desplegar en un dispositivo móvil.

6.1.1.3 Paquete Plantilla

En este paquete se encuentran las clases que corresponden a la plantilla. En ella están todas las clases que después de que el analizador haga la sustitución de las partes variables por los valores obtenidos de la instancia del DSL, permitirán crear un juego tal y como se diseñó en la interfaz.

6.2 Diseño de Clases

6.2.1 Diagrama de Clases

6.2.1.1 Diagrama de Clases Paquete Interfaz

En el diagrama que se muestra a continuación se detalla el diagrama de clases del paquete interfaz. Debido al tamaño que tiene solamente se mostrarán las clases más representativas de dicho paquete, y posteriormente se indicarán en otros diagramas las clases restantes.

Todas las clases existentes en el diagrama se relacionan con las siguientes tres clases:

- **Store**: sirve para almacenar datos en tiempo de ejecución.
- **Almacen**: se utiliza para guardar los estados de los objetos para su posterior carga.
- **Serializer**: sirve para serializar los objetos y guardarlos en disco.

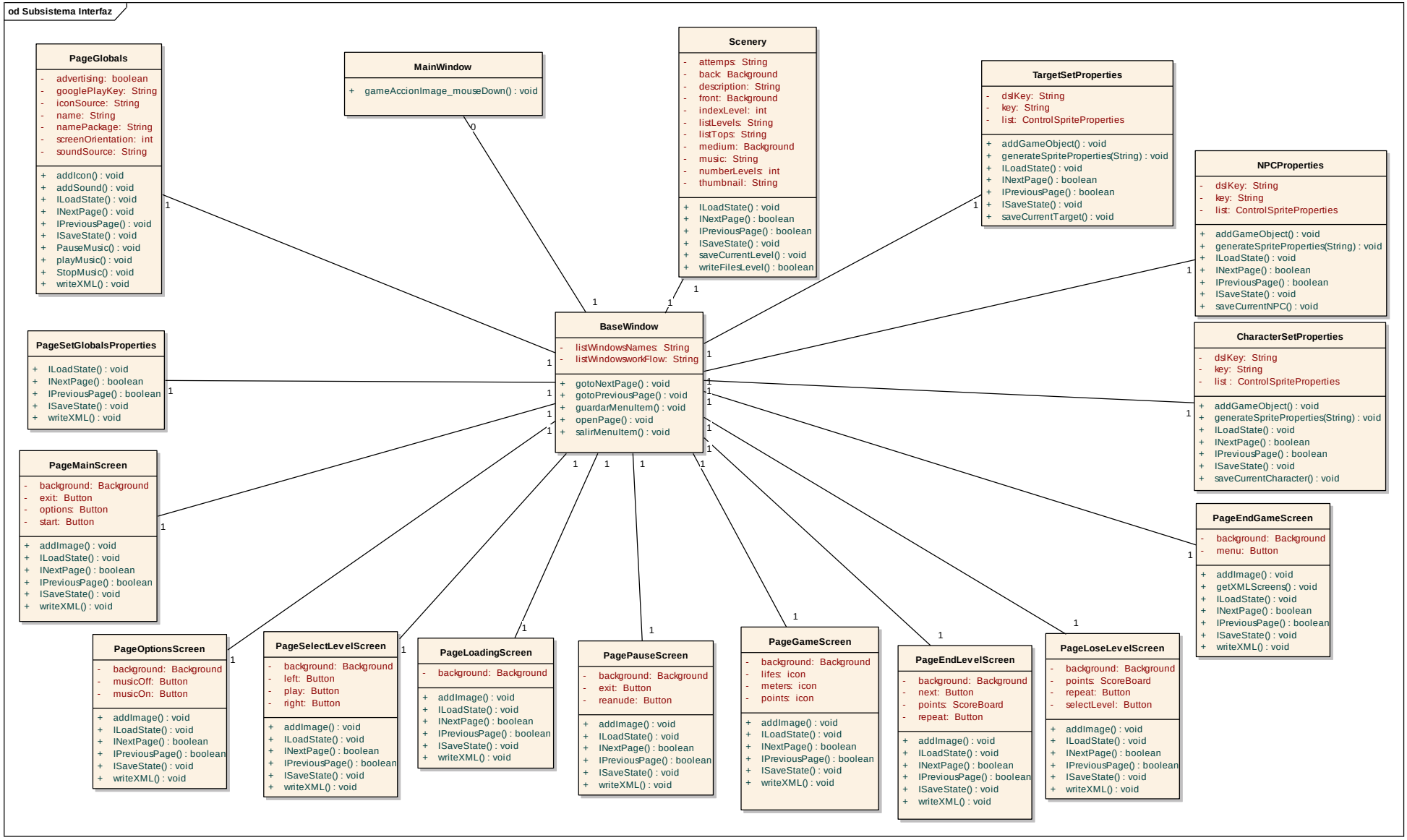


Ilustración 6.2. Diagrama de clases diseño: Subsistema Interfaz

6.2.1.1.1 Diagrama de Clases Paquete Interfaz: Scenery

La clase *Scenery* se relaciona con las clases "*Character*" y "*Stage*", que contendrán los datos de personaje principal y las propiedades del escenario respectivamente.

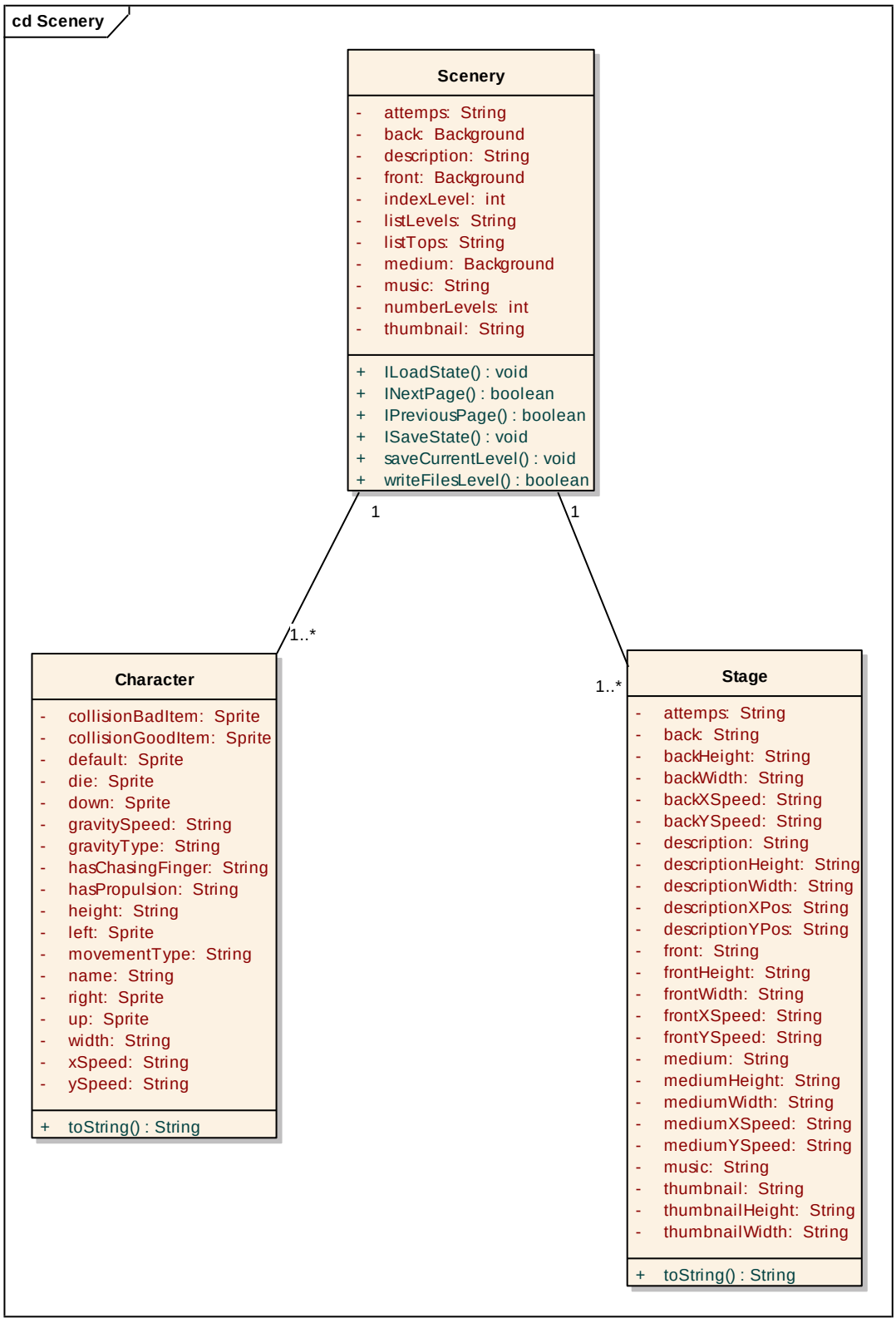


Ilustración 6.3. Diagrama de clases diseño: clase Scenery

6.2.1.1.2 Diagrama de Clases Paquete Interfaz: Store

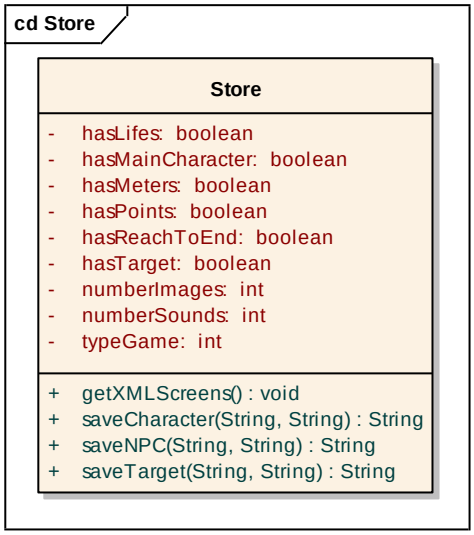


Ilustración 6.4. Diagrama de clases diseño: clase Store

6.2.1.1.3 Diagrama de Clases Paquete Interfaz: Almacen

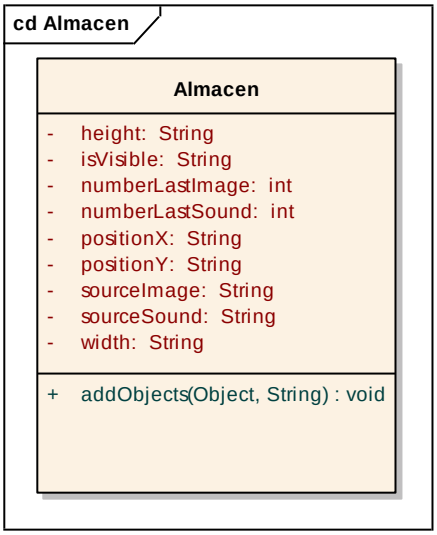


Ilustración 6.5. Diagrama de clases diseño: clase Almacen

6.2.1.1.4 Diagrama de Clases Paquete Interfaz: Serializer

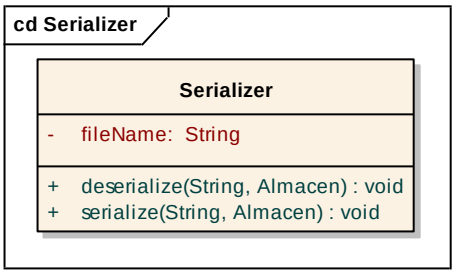


Ilustración 6.6. Diagrama de clases diseño: clase Serializer

6.2.1.2 Diagrama de Clases Paquete Analizador

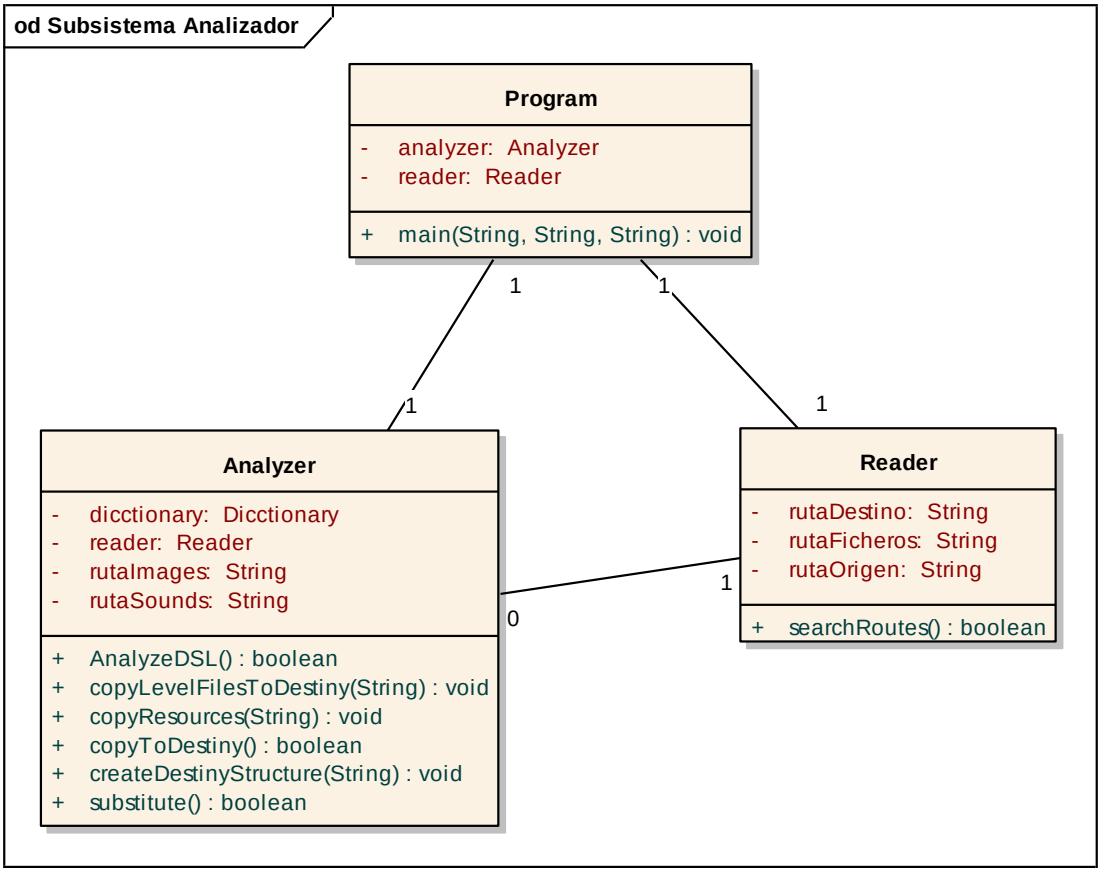


Ilustración 6.7. Diagrama de clases diseño Subsistema analizador

6.2.1.3 Diagrama de Clases Paquete Plantilla

Debido al tamaño del diagrama de clases correspondiente al paquete Plantilla, se va a dividir en tres diagramas:

- **Global**: engloba todas las clases del diagrama pero las clases Model y Screen no están detalladas y no se muestran las clases que dependen de ellas.
- **Model**: muestra la clase Model de forma detalla.
- **Screens**: contiene las clases de Screens.

6.2.1.3.1 Diagrama Global Paquete Plantilla

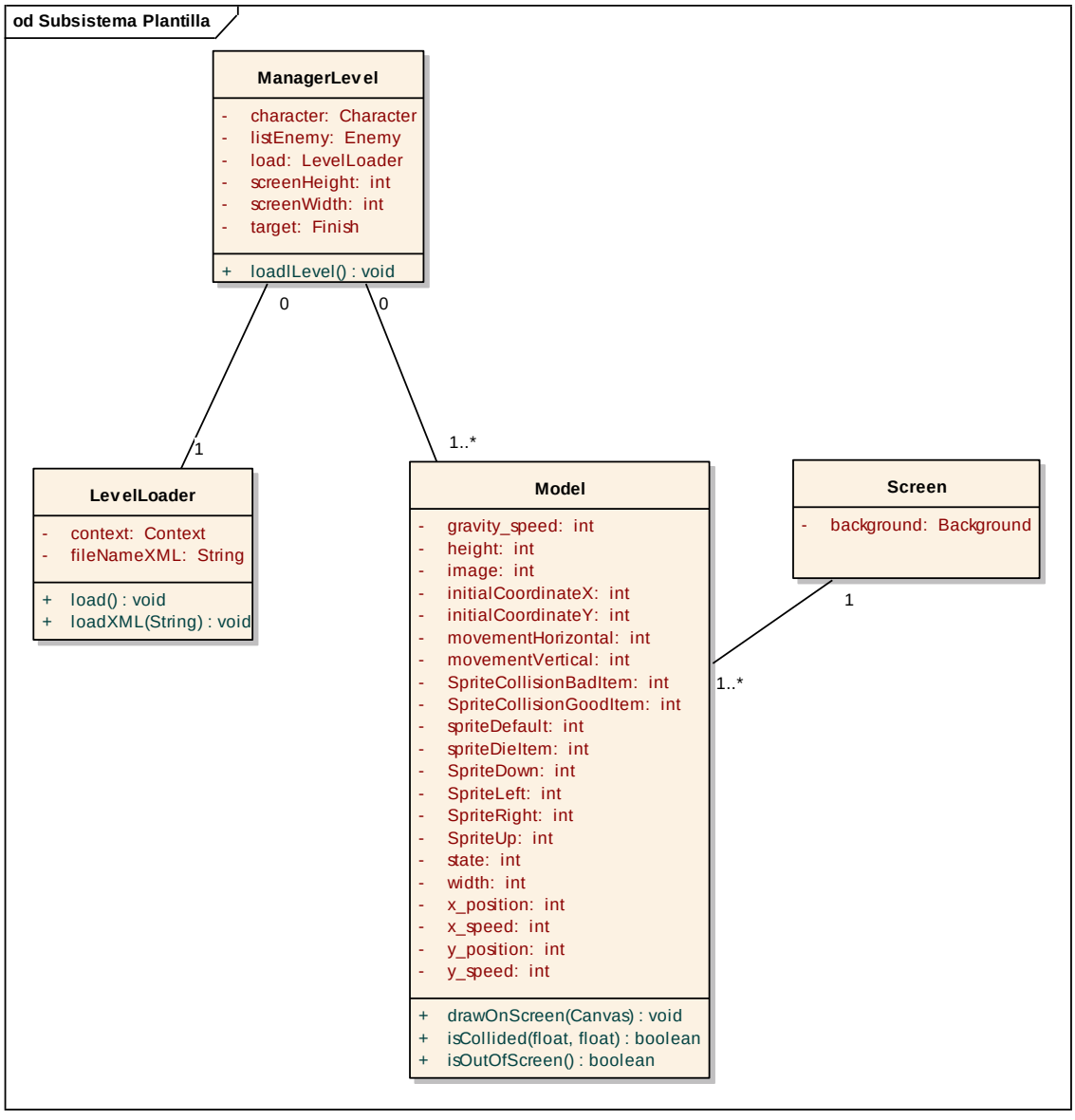


Ilustración 6.8. Diagrama de clases diseño Subsistema Plantilla

6.2.1.3.2 Diagrama de Clases Paquete Plantilla: Model

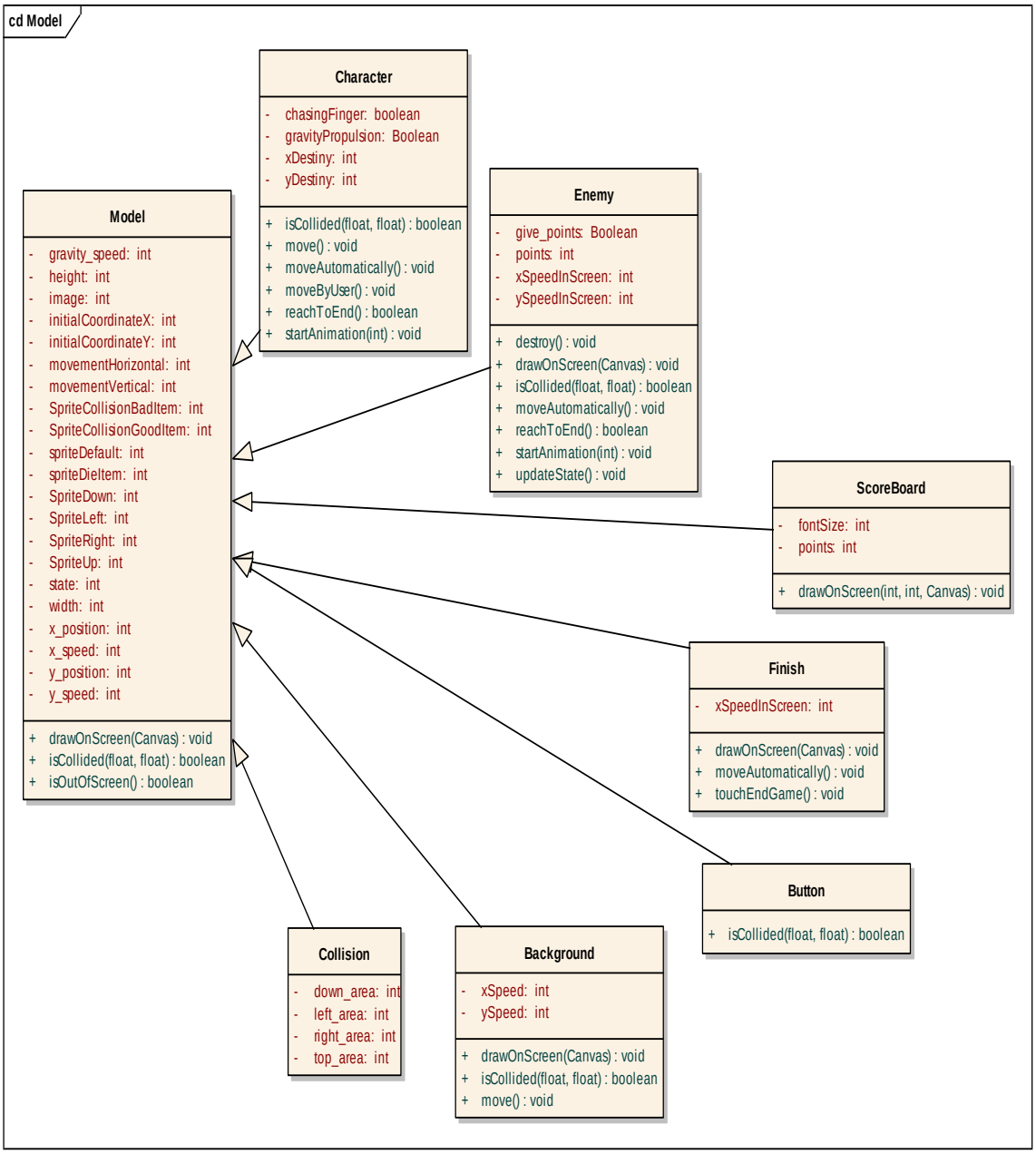


Ilustración 6.9. Diagrama de clases diseño: clase Model

6.2.1.3.3 Diagrama de Clases Paquete Plantilla: Screen

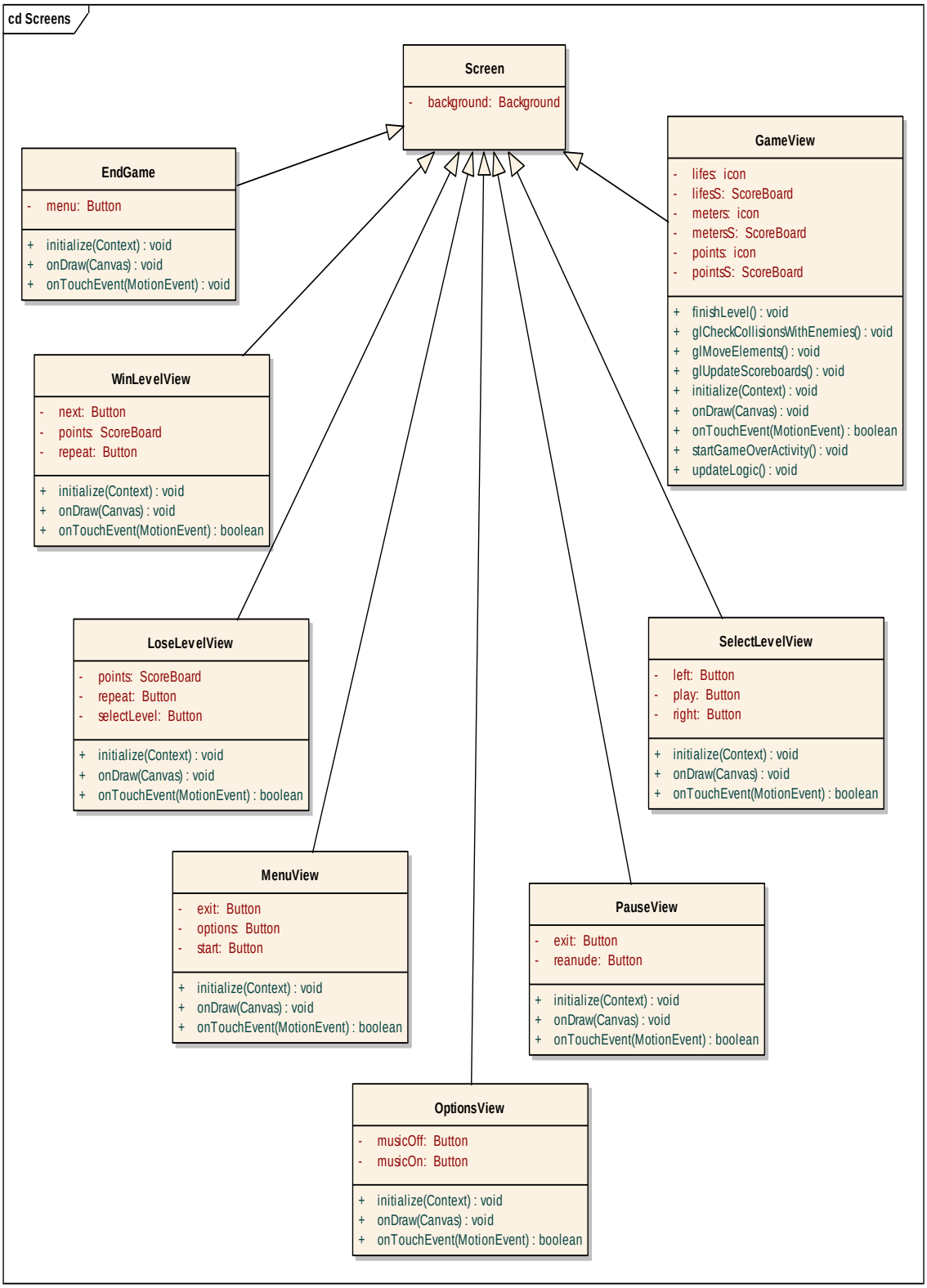


Ilustración 6.10. Diagrama de clases diseño: clase Screen

6.3 Diagramas de Interacción y Estados

Para los diagramas de interacción y estados, se ha optado por representar el caso de uso "Crear Videojuego" ya que es el principal proceso que se realiza con la herramienta.

6.3.1 Caso de Uso: Crear Videojuego

6.3.1.1 Diagramas de Interacción

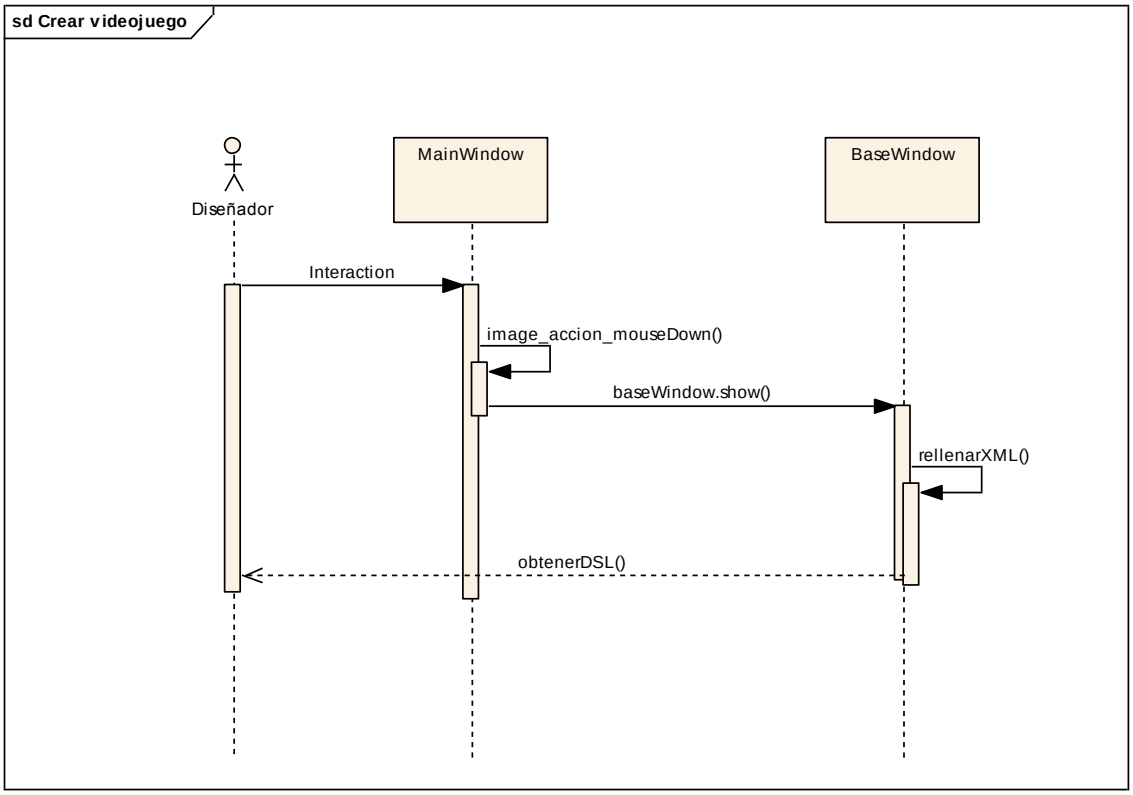


Ilustración 6.11. Diagrama interacción Caso de Uso: Crear Videojuego

La primera interacción del usuario se produce cuando ejecuta la aplicación. Cuando selecciona la tipología en la pantalla principal se lanza el evento asociado al botón, que lanzaría un cuadro de diálogo para escoger la ruta donde guardar/cargar el juego. Justo después de finalizar ese evento se abre la ventana contenedora de las pantallas de la interfaz.

El evento *"rellenarXML()"* es un resumen de todo el proceso que se realiza navegando por la interfaz, dentro de cada pantalla se van configurando las propiedades correspondientes a cada objeto y se van almacenado sus valores en el fragmento de la instancia del DSL que le corresponde, de esta manera se va construyendo la instancia mediante la navegación entre pantallas. Una vez finalizada la navegación entre pantallas y construidos todos los objetos del juegos, se obtiene la instancia del DSL con los valores del videojuego diseñado por el usuario.

6.4 Diagramas de Actividad

El proceso de crear un nuevo videojuego es el más complejo a realizar en el sistema, por tanto se ha tomado el caso de uso "Crear nuevo videojuego" para documentarlo mediante un diagrama de actividad.

Las actividades que se realizan en este proceso son las siguientes:

- **Configurar propiedades globales.** El usuario configurará las propiedades globales del juego que va a crear, y dentro de esta configuración tendrá que tomar la decisión de si el juego tendrá publicidad o no.
- **Configurar propiedades específicas.** El usuario podrá escoger el tipo de juego, dentro de la tipología de acción, y decidirá si tendrá puntos, vidas o metros, y si el juego finalizará cuando el personaje principal choque con una meta o bien cuando los enemigos no destruidos se escapen de la pantalla y agoten todas las vidas del personaje principal.
- **Pantalla del menú principal y de los otros menús del juego.** El usuario podrá configurar los botones y fondos de cada menú o dejar los que se ofrecen por defecto.
- **Crear personaje.** Se deberá definir el personaje principal del juego, dando valores a sus propiedades y eligiendo las imágenes para las animaciones.
- **Crear enemigos.** Se configurarán los enemigos que aparecerán en el juego, con sus propiedades y las imágenes asociadas a cada animación. El usuario decidirá si habrá que esquivar el enemigo o si dará puntos cuando se colisione con él.
- **Crear meta**. El usuario, en caso de haber tomado la decisión de que el juego tendrá una meta para finalizar un nivel, deberá configurar esta, escogiendo la imagen que tendrá.
- **Definir nivel.** El usuario creará los niveles que desee, añadiéndoles los personajes creados en las actividades anteriores. Una vez finalizada la actividad de crear y configurar los niveles se comprobará si cada uno de ellos tiene un personaje y una meta. En caso de no tenerlos se tendrá que seguir configurando el nivel, y si están en cada nivel, finalmente, se generará el videojuego creado.

Desde una actividad se accede a la siguiente y también a cualquier actividad que ya se haya realizado.

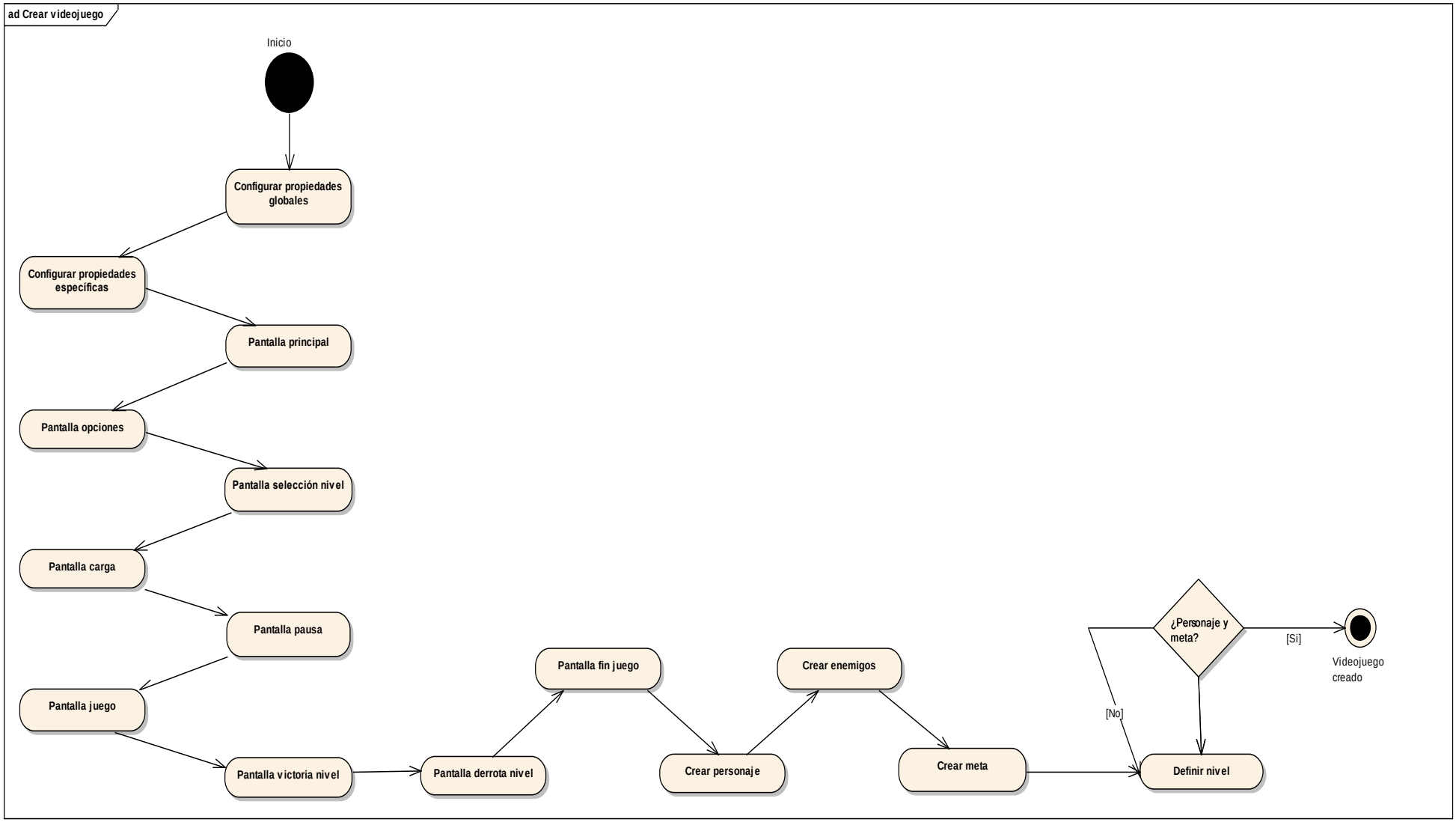


Ilustración 6.12. Diagrama actividad: Crear Videojuego

6.5 Diseño de la Interfaz

6.5.1 Pantalla principal

La interfaz consta de una pantalla principal en la que se podrá escoger la opción de crear un videojuego desde cero o cargar uno ya creado anteriormente.

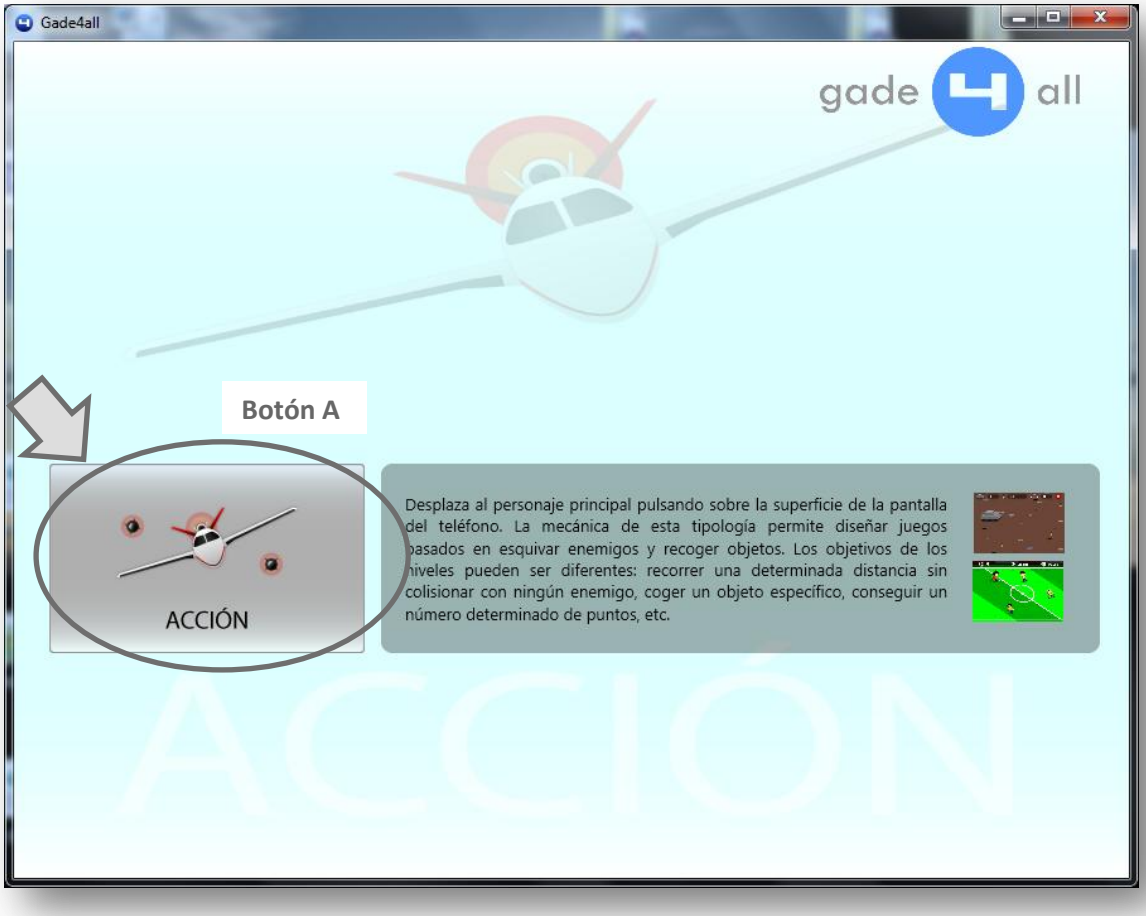


Ilustración 6.13. Pantalla principal

Al hacer clic sobre el botón A se abrirá un cuadro de diálogo en el que se escogerá la ruta donde se almacenarán los juegos que se vayan a crear. Si se desea cargar algún juego creado anteriormente, bastaría con seleccionar en ese cuadro de diálogo, el directorio que contiene el juego, y este se cargaría automáticamente.

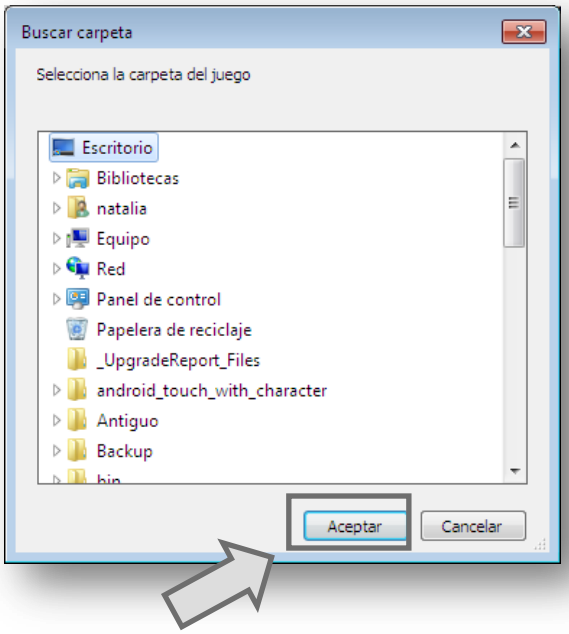


Ilustración 6.14. Seleccionar ruta almacenar

Una vez escogido el directorio, se hará clic sobre el botón "Aceptar".

6.5.2 Menú de la aplicación

Todas las pantallas de configuración, se abren dentro de una ventana contenedora, y esta tiene un menú en la parte superior (A) y una barra (B) en la que se muestra un botón por cada pantalla que el usuario ya ha visitado y que son también accesos directos a cada pantalla. Si se hace clic sobre uno de esos botones se cargará la pantalla correspondiente.

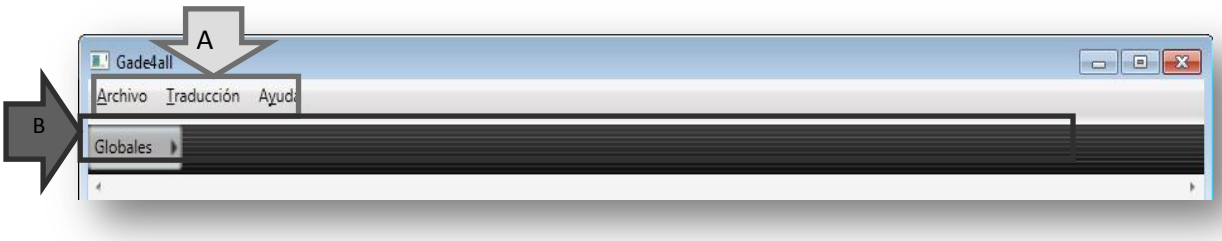


Ilustración 6.15. Menú de la aplicación

Este sería el aspecto de la barra de botones (B) cuando se ha visitado más de una pantalla:

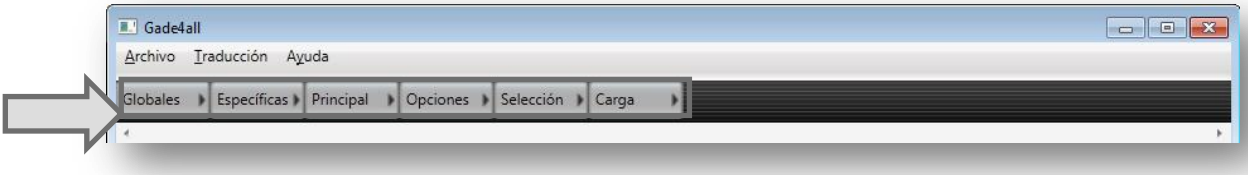


Ilustración 6.16. Barra botones interfaz

En el menú de la aplicación (A) hay dos elementos: "Archivo" y "Traducción". Al hacer clic sobre "Archivo" se despliega el siguiente menú:

Ilustración 6.17. Menú opciones

En el cual hay disponibles dos opciones:

- **Cerrar:** Se cerrará la aplicación.
- **Guardar:** Se guardará el estado de la pantalla en la que se encuentre el usuario en ese momento.

El elemento "*Traducción*" aún no tiene implementada funcionalidad, pero su objetivo será poder seleccionar un idioma y modificar así el de la herramienta.

En la parte inferior de la ventana contenedora de la aplicación hay dos botones "*Atrás*" y "*Siguiente*" que permitirán al usuario navegar por las pantallas de la aplicación.

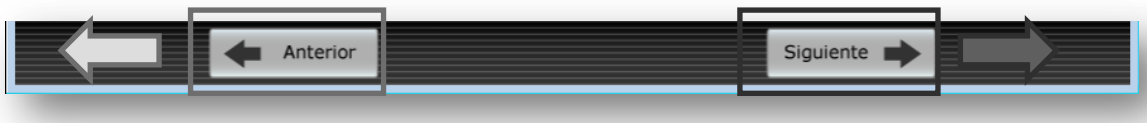


Ilustración 6.18. Botones navegación

Al hacer clic en "*Atrás*" se volverá a la pantalla anterior y se cargará el último estado de esa pantalla. En cambio, si se pulsa "*Siguiente*", se abrirá la siguiente pantalla y se guardará el estado de la página en la que se encontraba en el momento de pulsar el botón.

6.5.3 Pantalla configuración propiedades globales

Se muestra en la interfaz después de haber seleccionado la ruta donde almacenar el videojuego que se va a crear; es la de configuración de propiedades globales del videojuego.

Ilustración 6.19. Pantalla configuración propiedades globales

Se podrá configurar:

- El nombre del videojuego.
- El icono.
- El nombre de paquete del proyecto Eclipse que se obtendrá en el último paso de la creación del videojuego.
- Incluir publicidad.
- La orientación de la pantalla.
- La música de fondo de los menús.

Una vez configuradas todas las propiedades, se pulsará sobre el botón "*Siguiente*" para continuar.

6.5.4 Pantalla configuración propiedades específicas

En ella se configuran las propiedades específicas del videojuego.

Y se podrá elegir un tipo de juego basado en el control sobre el personaje principal, los marcadores que se incluirán en la pantalla de juego y el tipo de final de los niveles.

Ilustración 6.20. Pantalla configuración propiedades específicas

A continuación se hace clic en "*Siguiente*" para pasar a la siguiente pantalla de la interfaz.

6.5.5 Pantalla menú principal

Esta pantalla se corresponde con la del menú principal del juego, la primera que se mostrará cuando se entre en el videojuego.

En ella se podrá configurar la imagen de fondo y los tres botones de los que consta, mediante el panel de propiedades situado a la izquierda de la pantalla. Para configurar un elemento se deberá seleccionar haciendo clic sobre él.

Ilustración 6.21. Configuración menú principal

Una vez configurada la pantalla del menú principal, se hace clic sobre "*Siguiente*" para guardar los cambios y continuar con el diseño del videojuego.

6.5.6 Pantalla menú opciones

En esta pantalla se configurará el menú de opciones del videojuego. Se podrá elegir la imagen de fondo y configurar los botones de activar y desactivar música.

Ilustración 6.22. Configurar menú opciones

Mediante esos dos botones, en el videojuego, se podrá desactivar o activar la música.

Al igual que en las otras pantallas, se pulsa "*Siguiente*" para guardar los cambios y continuar el recorrido por la interfaz.

6.5.7 Pantalla menú de selección de nivel

La siguiente pantalla de la interfaz, se utiliza para configurar el menú de selección de nivel. En él habrá una imagen de fondo, una miniatura del nivel, de la que de momento solamente se podrá configurar su tamaño y posición, las flechas que servirán para navegar entre las miniaturas del nivel, y un botón para seleccionar un nivel y comenzar a jugar.

Ilustración 6.23. Configuración menú selección nivel

6.5.8 Pantalla carga

En esta pantalla de la interfaz se podrá configurar la pantalla de carga del videojuego, que se mostrará mientras se carga el nivel a jugar. Solamente contiene una imagen de fondo.

Ilustración 6.24. Configurar menú carga

6.5.9 Pantalla de juego

La siguiente pantalla de la interfaz se corresponde con la pantalla de juego, donde se cargarán y mostrarán los niveles.

Contiene marcadores de vidas, puntos y metros, y se mostrarán aquellos que se hayan elegido en la pantalla de configuración de propiedades específicas. Se podrá escoger la posición de los marcadores y los iconos que los acompañarán. También hay un botón de pausa y una imagen de fondo, sobre la que se dispondrán los marcadores y la pausa.

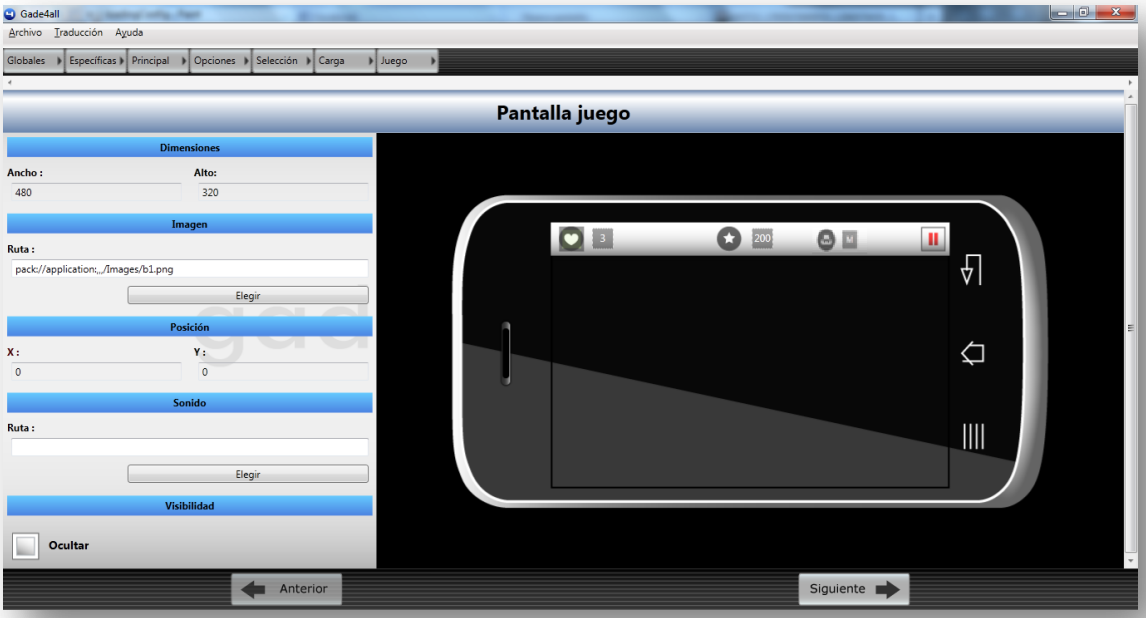


Ilustración 6.25. Configuración menú juego

6.5.10 Pantalla menú pausa

El menú que se va a configurar en esta pantalla de la interfaz, es el de pausa, que se mostrará cuando en la pantalla de juego (la descrita en el paso anterior) se pulse sobre el botón de pausa.

Está compuesta por una imagen de pausa y dos botones, uno para reanudar el juego y otro para salir al menú principal.

Ilustración 6.26. Configurar menú pausa

6.5.11 Pantalla menú victoria nivel

La siguiente pantalla es la de configuración del menú de victoria de nivel. En el videojuego se mostrará cuando se supere un nivel con éxito.

Contiene un marcador de puntos obtenidos, y dos botones, uno para repetir el nivel y otro para jugar el siguiente. También tiene una imagen de fondo. Todos estos elementos se pueden configurar.

Ilustración 6.27. Configurar menú victoria nivel

6.5.12 Pantalla menú derrota nivel

La siguiente pantalla de la interfaz se va a utilizar para configurar el menú de derrota de nivel. En el videojuego esta pantalla se corresponde con el menú que se muestra cuando se pierde un nivel.

Se podrán ver los puntos obtenidos en un marcador. Cuenta con una imagen de fondo y dos botones, uno para repetir el nivel y otro para acceder al menú de selección de nivel.

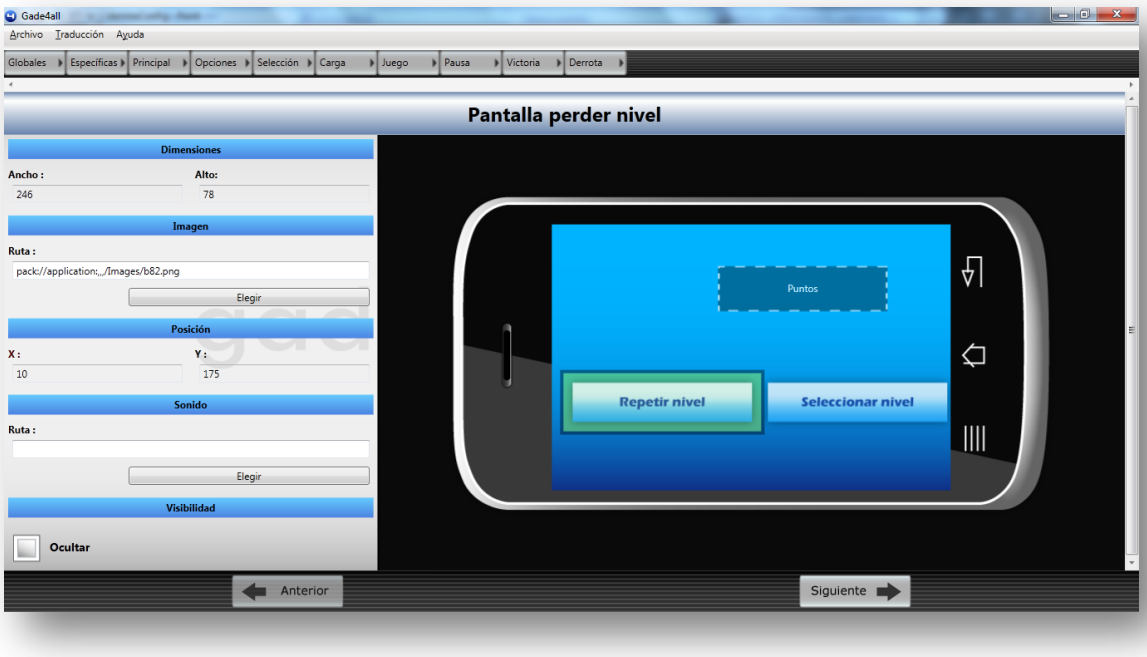


Ilustración 6.28. Configurar menú derrota nivel

6.5.13 Pantalla menú fin juego

Esta es la última pantalla en el proceso de configuración de los menús del videojuego. Es el menú que se muestra cuando se han superado todos los niveles.

Está compuesta por una imagen de fondo y un botón, que servirá para volver al menú principal del juego.

Ilustración 6.29. Configurar menú fin juego

6.5.14 Pantalla personaje principal

Una vez configurados los menús del juego, se continúa con la creación de los personajes del juego.

Se comienza con el personaje principal, configurando sus propiedades y eligiendo las imágenes de sus animaciones. Se podrán crear tantos como se deseen, pero en cada nivel solamente se podrá añadir uno.

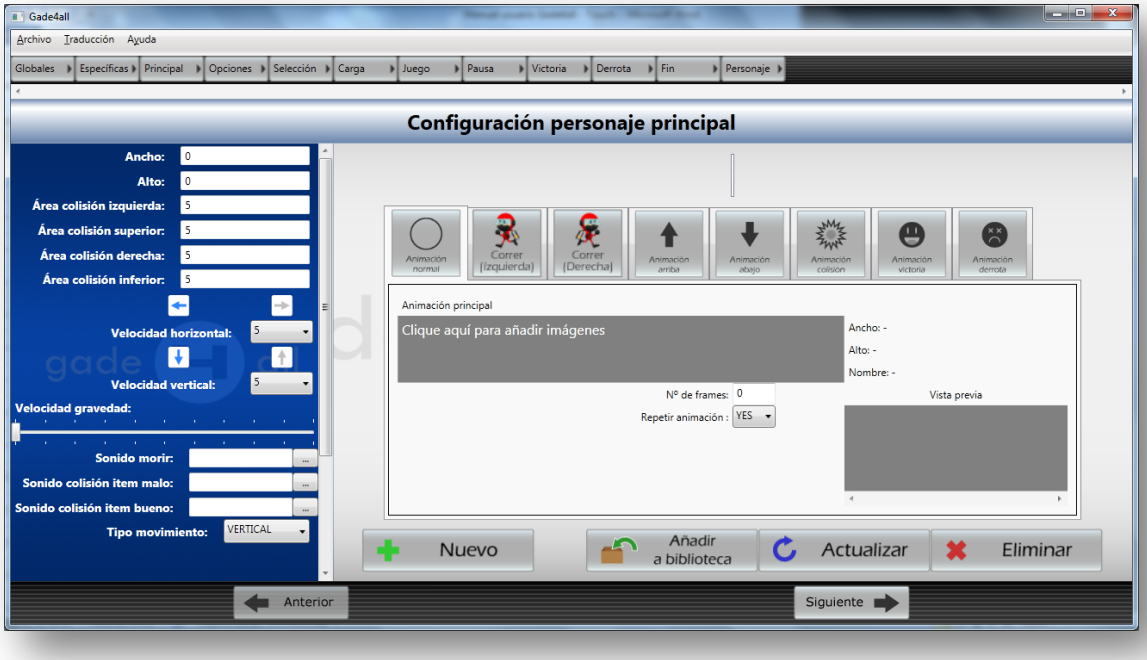


Ilustración 6.30. Configurar personaje principal

6.5.15 Pantalla enemigos

Una vez configurado el personaje o personajes principales, el siguiente paso a realizar en la interfaz, es la creación de los enemigos.

Se pueden generar tantos como se desee, y se podrán configurar sus propiedades y elegir las imágenes de cada animación.

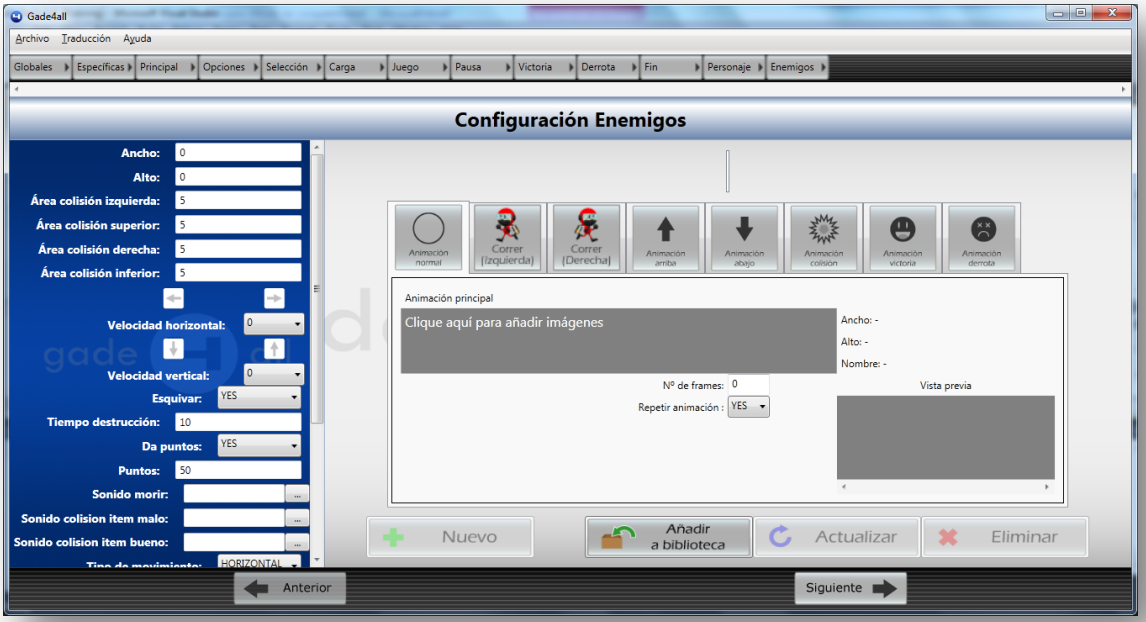


Ilustración 6.31. Configurar enemigos

6.5.16 Pantalla meta

Ya configurados los personajes del juego, el siguiente paso es crear una meta, para que el personaje principal colisione con ella para finalizar el nivel. Si en la pantalla de configuración de propiedades específicas se escogió que el videojuego no tenga meta, no se tiene que configurar esta pantalla y se puede pasar a la siguiente.

Se podrá configurar las propiedades de la meta y escoger la imagen que tendrá. Se pueden crear varias pero solo se puede colocar una en cada nivel.

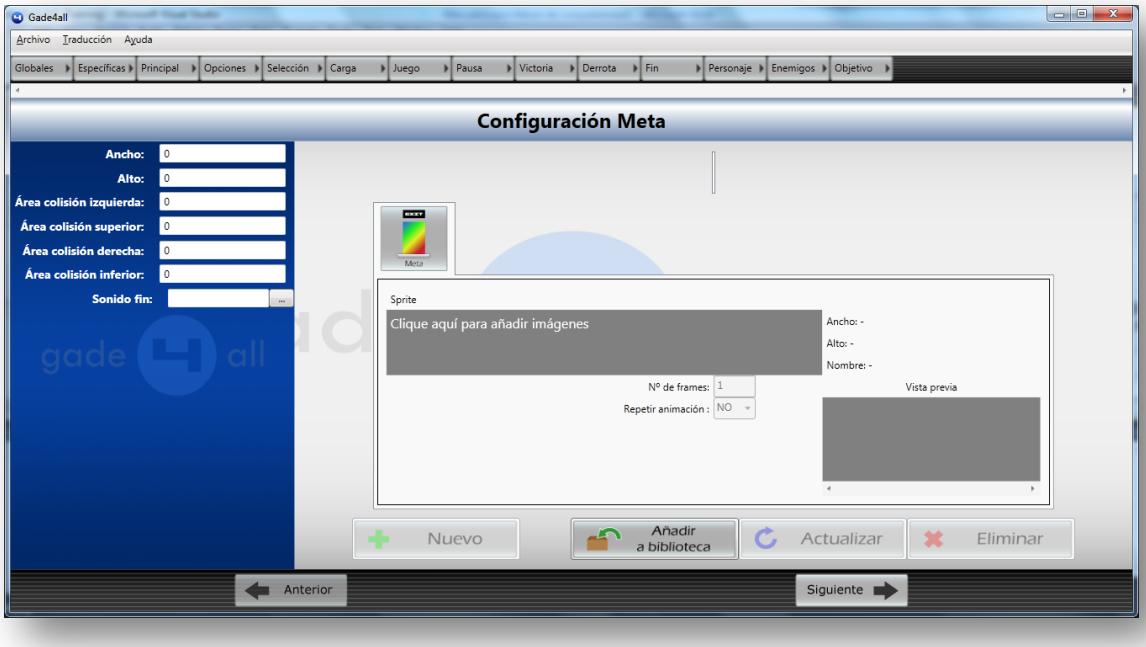


Ilustración 6.32. Configurar meta

6.5.17 Pantalla escenario

El siguiente y último paso a realizar es la definición de los niveles. Una vez que ya se tienen el personaje principal, los enemigos y la meta, hay que diseñar cada nivel del videojuego.

En esta pantalla se podrán colocar los diferentes elementos en el escenario y configurar las propiedades de cada nivel. Cada nivel tiene que tener de forma obligatoria un personaje principal y una meta, en caso de que se haya elegido que el videojuego tenga meta.

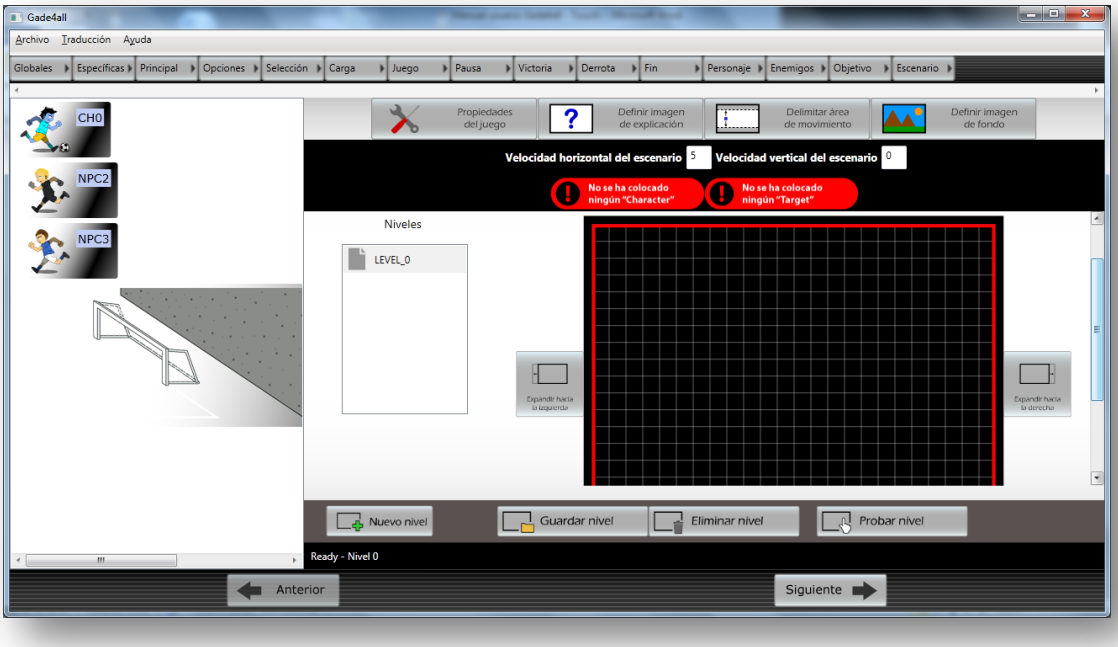


Ilustración 6.33. Definir escenario

6.5.17.1 Pantalla propiedades nivel

En el escenario se pueden configurar las propiedades de cada nivel mediante esta pantalla. Una vez se hayan configurado o mantenido las que vienen por defecto, se pulsará sobre el botón "*OK*" para guardar la configuración, o "*Cancelar*" para descartar los cambios.

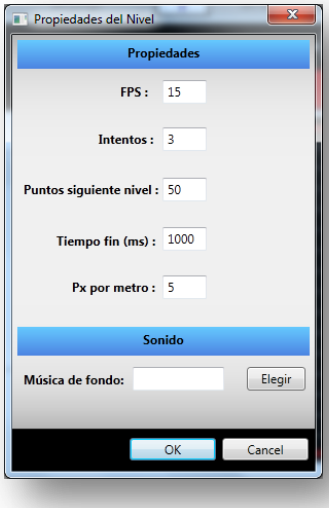


Ilustración 6.34. Configurar propiedades nivel

6.5.17.2 Pantalla Descripción nivel

Desde esta pantalla se podrá elegir la imagen de descripción que tendrá el nivel que se está configurando y también la imagen de miniatura del nivel, que se mostrará en el menú de selección de nivel del videojuego. Para guardar los cambios se pulsará "*Ok*" y "*Cancel*" para descartarlos. Si no se escoge ninguna se asignará una imagen en blanco.

Ilustración 6.35. Configurar descripción nivel

6.5.17.3 Pantalla Delimitadores

Para elegir entre qué coordenadas se va a desplazar el personaje principal o bien entre que topes de la pantalla se va a mover, se tienen que configurar. Se mostrará una lista con los topes y se podrán editar los existentes o bien añadir unos nuevos.

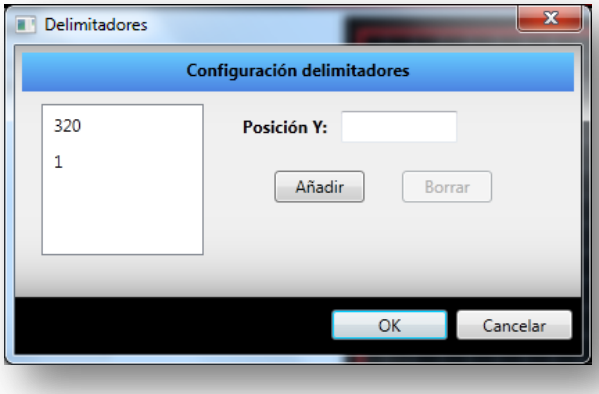


Ilustración 6.36. Configurar delimitadores

6.5.17.4 Pantalla Fondos

Desde esta pantalla se podrá seleccionar la imagen o imágenes de fondo del nivel. Podrá ser una única capa o multicapa, y así provocar la sensación de que el fondo tiene profundidad.

Una vez se haya seleccionado el fondo o fondos, se pulsará "*Ok*" para guardar y "*Cancelar*" para descartar.

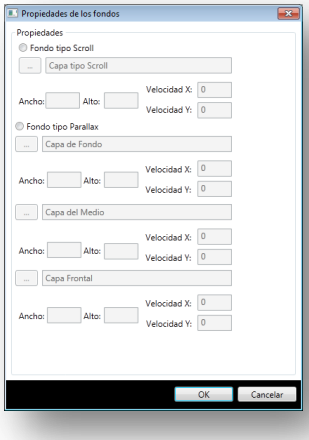


Ilustración 6.37. Configurar imagen fondo

6.5.18 Pantalla selección ruta generación

Una vez finalizado el proceso completo del diseño, se va a proceder a su generación. Para ello se escogerá en esta pantalla la ruta donde se desea almacenar el proyecto del videojuego creado, que solamente habrá que compilar y estará listo para ser ejecutado en un dispositivo móvil y comenzar a jugar.

Ilustración 6.38. Generación juego

6.5.19 Avisos y Mensajes

6.5.19.1 Guardar estado

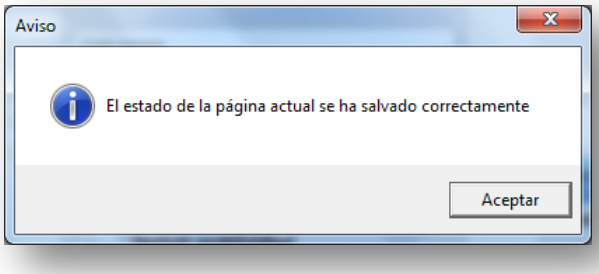


Ilustración 6.39. Mensaje guardar estado

Este mensaje se mostrará cuando el usuario pulse la opción "*Guardar*" del menú.

6.5.19.2 Creación personaje principal

Si se ha configurado un personaje principal pero no se ha guardado en la biblioteca se avisará al usuario mediante el siguiente mensaje:

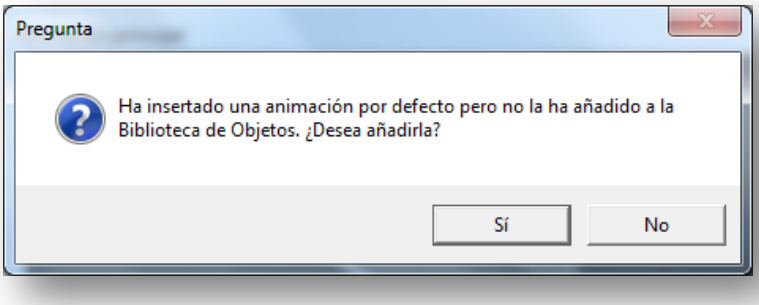


Ilustración 6.40. Mensaje creación personaje principal

6.5.19.3 Creación enemigos

Se mostrará el mismo mensaje que en la creación del personaje principal

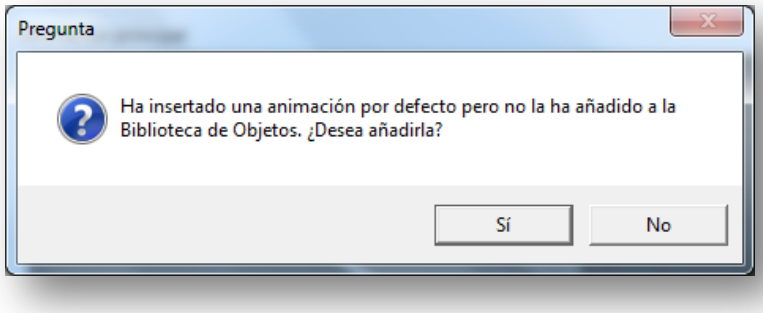


Ilustración 6.41. Mensaje creación enemigos

6.5.19.4 Creación meta

Ídem que con el personaje principal y enemigos.

6.5.19.5 Creación nivel

Cuando se creé un nivel se mostrará este aviso en la barra de estado situada en la parte inferior de la pantalla de escenario.

Ilustración 6.42. Mensaje nivel creado

6.5.19.6 Guardar nivel

Cuando se pulse "*Guardar nivel*" en el escenario se mostrará este aviso en la barra de estado del escenario.

Ilustración 6.43. Mensaje nivel guardado

6.5.19.7 Eliminar nivel 0

Si se intenta borrar el nivel 0 se mostrará este mensaje informando con la forma de proceder para borrar este nivel.

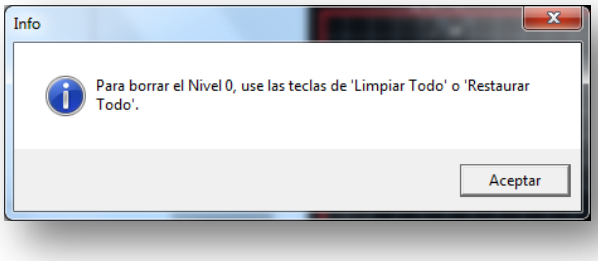


Ilustración 6.44. Mensaje eliminar nivel 0

6.5.19.8 Eliminar personajes posteriormente a la creación de un nivel

Si ya se han configurado el nivel o niveles del juego, se mostrará este aviso al usuario.

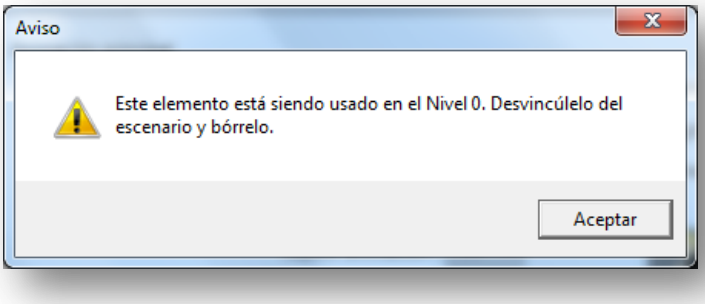


Ilustración 6.45. Mensaje eliminar personajes ya creados

6.5.19.9 Juego creado

Una vez finalizado el proceso de creación y diseño del videojuego y se va a acceder a la pantalla de generación se muestra este mensaje:

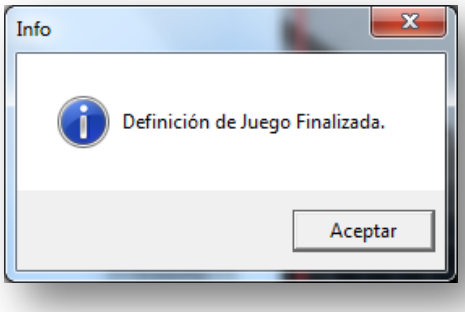


Ilustración 6.46. Mensaje creado

6.5.19.10 Proceso generación

Mientras se está generando el videojuego se muestra el siguiente mensaje de aviso al usuario.

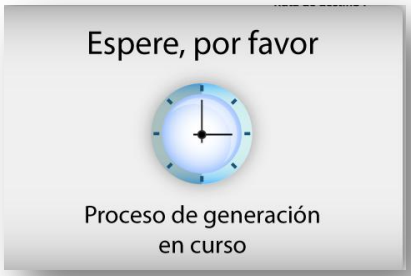


Ilustración 6.47. Mensaje generando juego

6.5.19.11 Juego generado

Una vez finalizada la generación del juego, se mostrará el siguiente mensaje:

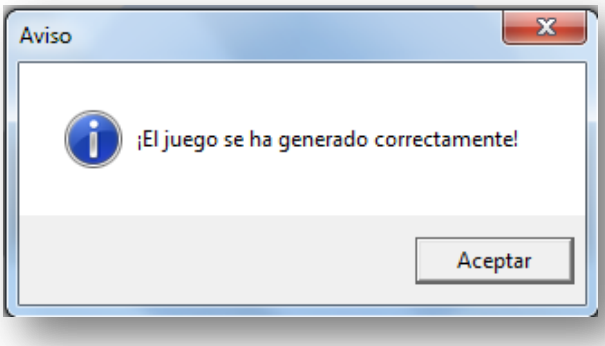


Ilustración 6.48. Mensaje juego generado

6.6 Diseño de la instancia del DSL

En este apartado se explicará el funcionamiento y se hará un análisis de los elementos que forman la instancia del DSL que genera la herramienta desarrollada en este proyecto, y que juega un papel muy importante dentro de la aplicación ya que a partir de ella, el analizador sustituirá en la plantilla los valores necesarios para generar el proyecto que contendrá el juego que el usuario ha diseñado con la interfaz.

En el Apéndice 13.4 se podrá ver un ejemplo completo de un fichero instancia del DSL generado con la herramienta.

6.6.1 Descripción y funcionamiento

El lenguaje específico del juego es un fichero en formato XML donde van impresos una serie de elementos que especifican una serie de reglas que se han de cumplir. Para formular las reglas se ha tenido en cuenta el realizar un lenguaje sencillo e intuitivo, siguiendo una serie de normas y conteniendo determinados elementos, que se detallan a continuación.

En este fichero instancia del DSL se recogen tanto las propiedades de los menús del juego, como los niveles del mismo.

```
<game>
    <global>
    ...../Subelementos
    </global>
    <screens>
    ...../Subelementos
    </screens>
    <game_properties>
    ...../Subelementos
    </game_properties>
    <levels>
        <level>
        ...../Subelementos
        </level>
        <level>
        ...../Subelementos
        </level>
    </levels>
</game>
```

6.6.2 Elementos

6.6.2.1 Elemento <game>

El elemento raíz de la instancia del DSL siempre será *<game>.* Este elemento solamente podrá aparecer una vez en el documento. Será la apertura y cierre del fichero.

```
<game>
    ...//Subelementos
</game>
```

Ilustración 6.49. Elemento game

6.6.2.2 Elemento <global>

El primer subelemento del elemento raíz es *<global>.* Este elemento solamente aparecerá una vez y contiene los valores de las propiedades globales del juego definidas por el usuario. Todos sus subelementos empezarán por la palabra "global_".

```
<global>
<global_name> </global_name>
<global_package_name> </global_package_name>
<global_screen_orientation> </global_screen_orientation>
<global_icon_source> </global_icon_source>
<global_has_advertising> </global_has_advertising>
<global_admob_id> </global_admob_id>
<global_sound_source> </global_sound_source>
</global>
```

Ilustración 6.50. Elemento global

6.6.2.2.1 Subelementos de <global>

Estos son los subelementos del elemento *<global>:*

- ***<global_name>:*** Representa el nombre del videojuego. Será el nombre que aparecerá en el móvil cuando se instale el videojuego.
- ***<global_package_name>:*** Representa el nombre del paquete del proyecto del Eclipse que contendrá los recursos que se generará con la herramienta. Este se rellenará automáticamente con el nombre del juego y tendrá el formato: "com.name".
- ***<global_screen_orientation>:*** Representa la orientación que tendrá el juego. Si su valor es 1 la orientación será vertical y 0 horizontal.

- ***<global_icon_source>:*** Representa la ruta de la imagen del icono que tendrá el videojuego. Se verá cuando se instale en un dispositivo móvil. Vendrá en formato *.png*.
- ***<global_has_advertising>:*** Representa si el videojuego tendrá publicidad o no. Si su valor es "YES", llevará publicidad, si es "NO", no la tendrá.
- ***<global_admob_id>:*** Representa el id de Android para la publicidad. Si no tiene publicidad se dejará vacío.
- ***<global_sound_source>:*** Representa la ruta del archivo de sonido asociado a la melodía que sonará de fondo en los menús del videojuego. Tendrá formato *.mp3*. En caso de no haber música de fondo se puede dejar vacío.

6.6.2.3 Elemento <screens>

El segundo subelemento del elemento raíz es *<screens>*. Este elemento solamente aparecerá una vez y contiene las propiedades de cada uno de los menús del juego diseñados por el usuario. Todos sus subelementos tendrán el siguiente nombre "name_*screen*", siendo *name* el nombre de cada pantalla. Cada subelemento solamente aparecerá una vez, ya que solo hay una pantalla de cada.

```
<screens>
    <main_screen>
    </main_screen>

    <options_screen>
    </options_screen>

    <select_level_screen>
    </select_level_screen>

    <loading_screen>
    </loading_screen>

    <pause_screen>
    </pause_screen>

    <end_level_screen>
    </lose_level_screen>

    <end_game_screen>
    </end_game_screen>

    <game_screen>
    </game_screen>

</screens>
```

Ilustración 6.51. Elemento screens

Todos estos subelementos de *<screens>* van a tener el mismo tipo de subelementos que se corresponde con botones e imágenes. Por eso, se va a explicar en este punto del documento lo que significa cada uno, ya que en todas las pantallas tienen la misma

funcionalidad. Solamente se describirán en cada pantalla aquellos elementos que sean únicos de cada tipo de pantalla.

- Botones:

```
<name_screen_name_button_image_source> </name_screen_name_button_image_source>
<name_screen_name_button_image_width> </name_screen_name_button_image_width>
<name_screen_name_button_image_height> </name_screen_name_button_image_height>
<name_screen_name_button_x_position> </name_screen_name_button_x_position>
<name_screen_name_button_y_position> </name_screen_name_button_y_position>
<name_screen_name_button_sound_source> </name_screen_name_button_sound_source>
```

Ilustración 6.52. Subelementos button

En *name_screen*, *name* se corresponde con el nombre de la pantalla y en *name_button*, se corresponde con el nombre del botón. Y los botones tienen las siguientes propiedades:

- ***<name_screen_name_button_image_source>:*** Representa la ruta de la imagen del botón. Tendrá formato *.png*.
- ***<name_screen_name_button_image_width>:*** Representa el ancho de la imagen del botón. Tiene que ser un número entero positivo mayor que 0.
- ***<name_screen_name_button_image_height>:*** Representa el alto de la imagen del botón. Tiene que ser un número entero positivo mayor que 0.
- ***<name_screen_name_button_x_position>:*** Representa la posición del botón en el eje de coordenadas X. Tiene que ser un número entero.
- ***<name_screen_name_button_y_position>:*** Representa la posición del botón en el eje de coordenadas Y. Tiene que ser un número entero.

- Imágenes:

```
<name_screen_image_source> </name_screen_image_source>
<name_screen_image_width> </name_screen_image_width>
<name_screen_image_height> </name_screen_image_height>
<name_screen_image_x_position> </name_screen_image_x_position>
<name_screen_image_y_position> </name_screen_image_y_position>
```

Ilustración 6.53. Subelementos image

En *name_screen*, *name* se corresponde con el nombre de la pantalla, y las imágenes tienen las siguientes propiedades:

- ***<name_screen_image_source>:*** Representa la ruta de la imagen. Tendrá formato *.png*.
- ***<name_screen_image_width>:*** Representa el ancho de la imagen. Tiene que ser un número entero positivo mayor que 0.

- ***<name_screen_image_height>:*** Representa el alto de la imagen. Tiene que ser un número entero positivo mayor que 0.
- ***<name_screen_image_x_position>:*** Representa la coordenada X de la imagen. Tiene que ser un número entero.
- ***<name_screen_image_y_position>:*** Representa la coordenada Y de la imagen. Tiene que ser un número entero.

6.6.2.3.1 Elemento <main_screen>

El primer subelemento de *<screens>* es *<main_screen>,* relativo a la pantalla principal del juego, la primera que se muestra al abrir el juego.

```
<main_screen>
    <main_screen_start_button>
    ......//Subelementos
    </main_screen_start_button>
    <main_screen_options_button>
......//Subelementos
    </main_screen_options_button>
    <main_screen_exit_button>
    ......//Subelementos
    </main_screen_exit_button>
    <main_screen_image>
    ......//Subelementos
    </main_screen_image>
</main_screen>
```

Ilustración 6.54. Elemento main_screen

Contendrá los siguientes subelementos:

- ***<main_screen_start_button>:*** Representa el botón de jugar.
- ***<main_screen_options_button>:*** Representa el botón de opciones, para acceder a las opciones del juego.
- ***<main_screen_exit_button>:*** Representa el botón de salir, para salir del juego.
- ***<main_screen_image>:*** Representa la imagen de fondo de la pantalla principal.

6.6.2.3.2 Elemento <options_screen>

El siguiente subelemento de *<screens>* es *<options_screen>*. Este subelemento solamente aparecerá una vez en la instancia del DSL. Hace referencia a la pantalla de opciones del juego, en la que se desactiva y activa la música de fondo.

```
<options_screen>
    <options_screen_music_off_button>
    .........//Subelementos
    </options_screen_music_off_button>
    <options_screen_music_on_button>
    .........//Subelementos
    </options_screen_music_on_button>
    <options_screen_image>
    .........//Subelementos
    </options_screen_image>
</options_screen>
```

Ilustración 6.55. Elemento options_screen

Contendrá dos botones y una imagen:

- ***<options_screen_music_off_button>:*** Representa el botón de desactivar la música de fondo.
- ***<options_screen_music_on_button>:*** Representa el botón de activar la música de fondo.
- ***<options_screen_image>:*** Representa la imagen de fondo de la pantalla de opciones.

6.6.2.3.3 Elemento <select_level_screen>

El tercer subelemento de *<screens>* es *<select_level_screen>*. Este subelemento hace referencia a la pantalla de selección de niveles del juego, en la que se escoge un nivel para jugar.

```
<select_level_screen>
    <select_level_screen_next_level_button>
    ......//Subelementos
    </select_level_screen_next_level_button>
    <select_level_screen_previous_level_button>
    ......//Subelementos
    </select_level_screen_previous_level_button>
    <select_level_screen_play_level_button>
    ......//Subelementos
    </select_level_screen_play_level_button>
    <select_level_screen_level_thumbnail_view>
    ......//Subelementos
    </select_level_screen_level_thumbnail_view>
    <select_level_screen_level_image>
    ......//Subelementos
    </select_level_screen_level_image>
</select_level_screen>
```

Ilustración 6.56. Elemento select_level_screen

Contendrá tres botones y dos imágenes:

- ***<select_level_screen_next_level_button>:*** Representa el selector de siguiente nivel.
- ***< select_level_screen_previous_level_button>:*** Representa el selector de nivel anterior.
- ***< select_level_screen_play_level_button>:*** Representa el botón de jugar el nivel seleccionado.
- ***<select_level_screen_level_thumbnail_view>:*** Representa la miniatura del nivel. Tiene las mismas propiedades que las imágenes, excepto la ruta de imagen, que se escogerá cuando se diseñen los niveles, ya que es diferente para cada nivel.
- ***<select_level_screen_level_image>:*** Representa la imagen de fondo de la pantalla de selección de nivel.

6.6.2.3.4 Elemento <loading_screen>

El cuarto subelemento de *<screens>* es *<loading_screen>.* Este subelemento hace referencia a la pantalla carga del juego, que se mostrará mientras se carga el nivel que se ha escogido para jugar.

```
<loading_screen>
    <loading_screen_image>
    ......//Subelementos
    </loading_screen_image>
</loading_screen>
```

Ilustración 6.57. Elemento loading_screen

Solamente contendrá una imagen, en esta pantalla no hay botones:

- ***<loading_screen_image>:*** Representa la imagen de fondo de la pantalla de carga.

6.6.2.3.5 Elemento <pause_screen>

El quinto subelemento de *<screens>* es *<pause_screen>.* Hace referencia a la pantalla de pausa, que se mostrará cuando el usuario jugador pulse el botón de pausa de la pantalla de juego.

```
<pause_screen>
    <pause_screen_reanude_button>
    .......//Subelementos
    </pause_screen_reanude_button>
    .......//Subelementos
    <pause_screen_exit_button>
    .......//Subelementos
    </pause_screen_exit_button>
    <pause_screen_image>
    .......//Subelementos
    </pause_screen_image>
</pause_screen>
```

Ilustración 6.58. Elemento pause_screen

Contendrá dos botones y una imagen:

- ***< pause_screen_reanude_button>:*** Representa el botón de reanudar el nivel.
- ***<pause_screen_exit_button>:*** Representa el botón de salir del nivel que se está jugando.
- ***<pause_screen_image>:*** Representa la imagen de fondo de la pantalla de pausa.

6.6.2.3.6 Elemento <end_level_screen>

El siguiente subelemento de *<screens>* es *<end_level_screen>.* Hace referencia a la pantalla de fin de nivel que se muestra cuando se gana.

```
<end_level_screen>
    <end_level_screen_retry_button>
    ......//Subelementos
    </end_level_screen_retry_button>
    <end_level_screen_play_next_level_button>
    ......//Subelementos
    </end_level_screen_play_next_level_button>
    <end_level_screen_image>
    ......//Subelementos
    </end_level_screen_image>
    <end_level_screen_scoreboard>
    ......//Subelementos
    </end_level_screen_scoreboard>
</end_level_screen>
```

Ilustración 6.59. Elemento end_level_screen

Este elemento *<end_level_screen>,* contiene un nuevo tipo de subelemento, el *scoreboard*, que es un marcador en el que se muestran los puntos obtenidos en el nivel, y tiene las siguientes propiedades:

- Width: ancho del marcador.
- Height: alto del marcador.
- X_position: posición en el eje X del marcador.
- Y_position: posición en el eje Y del marcador.
- Text_font_size: tamaño de la fuente.
- Text_font_color: color de la fuente.

```
<name_screen_scoreboard>
<name_screen_scoreboard_width> </name_screen_scoreboard_width>
<name_screen_scoreboard_height> </name_screen_scoreboard_height>
<name_screen_scoreboard_x_position> </name_screen_scoreboard_x_position>
<name_screen_scoreboard_y_position> </name_screen_scoreboard_y_position>
<name_screen_scoreboard_text_font_size> </name_screen_scoreboard_text_font_size>
<name_screen_scoreboard_text_font_color> </name_screen_scoreboard_text_font_color>
</name_screen_scoreboard>
```

Ilustración 6.60. Elemento scoreboard

Este elemento *<end_level_screen>* contendrá dos botones, una imagen y un marcador:

- ***<end_level_screen_retry_button>:*** Representa el botón de repetir el nivel.
- ***<end_level_screen_play_next_level_button>:*** Representa el botón para jugar el siguiente nivel.
- ***< end_level_screen _image>:*** Representa la imagen de fondo de la pantalla de victoria de nivel.
- ***<end_level_screen_scoreboard>:*** Representa el marcador en el que se mostrarán los puntos obtenidos en el nivel jugado.

6.6.2.3.7 Elemento <lose_level_screen>

El siguiente subelemento de *<screens>* es *<lose_level_screen>.* Este subelemento hace referencia a la pantalla de fin de nivel que se muestra cuando pierdes.

```
lose_level_screen>
   <lose_level_screen_retry_button>
   ......//Subelementos
   </lose_level_screen_retry_button>
   <lose_level_screen_select_level_button>
   ......//Subelementos
   </lose_level_screen_select_level_button>
   <lose_level_screen_image>
   ......//Subelementos
   </lose_level_screen_image>
   <lose_level_screen_scoreboard>
   ......//Subelementos
   </lose_level_screen_scoreboard>
/lose_level_screen>
```

Ilustración 6.61. Elemento lose_level_screen

Contendrá dos botones, una imagen y un marcador:

- ***<lose_level_screen_retry_button>:*** Representa el botón de repetir el nivel.
- ***<lose_level_screen_select_level_button>:*** Representa el botón para acceder a la pantalla de selección de nivel.
- ***< lose_level_screen _image>:*** Representa la imagen de fondo de la pantalla de derrota de nivel.
- ***<lose_level_screen_scoreboard>:*** Representa el marcador en el que se mostrarán los puntos obtenidos en el nivel jugado.

6.6.2.3.8 Elemento <end_game_screen>

El penúltimo subelemento de *<screens>* es *<end_game_screen>.* Este subelemento hace referencia a la pantalla de fin de juego, que se muestra cuando ya se han superado todos los niveles.

```
<end_game_screen>
    <end_game_screen_return_level_button>
    ...../Subelementos
    </end_game_screen_return_level_button>
    <end_game_screen_image>
    ...../Subelementos
    </end_game_screen_image>
</end_game_screen>
```

Ilustración 6.62. Elemento end_game_screen

Contendrá un botón y una imagen:

- ***<end_game_screen_return_level_button>:*** Representa el botón para volver a la pantalla principal del juego.
- ***< end_game_screen _image>:*** Representa la imagen de fondo de la pantalla de fin de juego.

6.6.2.3.9 Elemento <game_screen>

Y finalmente, el último subelemento de *<screens>* es *<game_screen>.* Este subelemento hace referencia a la pantalla de juego.

```
<game_screen>
    <game_screen_pause_button>
    ......//Subelementos
    </game_screen_pause_button>
    <game_screen_status_bar>
    ......//Subelementos
    </game_screen_status_bar>
</game_screen>
```

Ilustración 6.63. Elemento game_screen

Este elemento *<game_screen>* contiene un nuevo tipo de subelemento, que es una barra de estado. En ella se mostrarán los iconos de vidas, metros y/o puntos y sus correspondientes marcadores, y el botón de pausa del juego. Y tiene los siguientes subelementos:

```
<game_screen_status_bar>
    <game_screen_status_bar_background>
    </game_screen_status_bar_background>
    <game_screen_status_bar_life>
    ....//Subelemento imagen
    ....//Subelemento marcador
    </game_screen_status_bar_life>
    <game_screen_status_bar_points>
    ....//Subelemento imagen
    ....//Subelemento marcador
    </game_screen_status_bar_points>
    <game_screen_status_bar_meters>
    ....//Subelemento imagen
    ....//Subelemento marcador
    </game_screen_status_bar_meters>
</game_screen_status_bar>
```

Ilustración 6.64. Elemento status_bar

- **Background**: imagen de fondo de la barra de estado. Es obligatorio.
- **Life**: icono y marcador de las vidas. Es opcional.
- **Points**: icono y marcador de los puntos. Es opcional.
- **Meters**: icono y marcador de los metros. Es opcional.

Por tanto, contendrá un botón y una barra de estado:

- ***<game_screen_pause_button>:*** Representa el botón para pausar el juego.
- ***<game_screen_status_bar>:*** Representa la imagen que contiene los iconos de elementos que se contabilizan en el nivel y sus correspondientes marcadores.

6.6.2.4 Elemento <game_properties>

El siguiente elemento del raíz es *<game_properties>,* que hace referencia a las propiedades específicas del juego.

```
<game_properties>
<game_properties_has_meters> </game_properties_has_meters>
<game_properties_has_points> </game_properties_has_points>
<game_properties_has_lives> </game_properties_has_lives>
<game_properties_has_main_character> </game_properties_has_main_character>
<game_properties_has_control_user> </game_properties_has_control_user>
<game_properties_has_chasing_finger> </game_properties_has_chasing_finger>
<game_properties_has_target> </game_properties_has_target>
<game_properties_has_reach_to_end> </game_properties_has_reach_to_end>
</game_properties>
```

Ilustración 6.65. Elemento game_properties

Contiene los siguientes subelementos:

- ***<game_properties_has_meters>:*** Representa si en el juego se contabilizarán los metros recorridos por el personaje principal. Su valor será *"YES"* o *"NO".*
- ***<game_properties_has_points>:*** Representa si en el juego se van a contabilizar los puntos obtenidos en el nivel. Su valor será *"YES"* o *"NO".*
- ***<game_properties_has_lives>:*** Representa si el personaje principal poseerá vidas durante el juego. Su valor será *"YES"* o *"NO".*
- ***<game_properties_has_main_character>:*** Representa si el juego tiene un personaje principal. Para la tipología de juegos escogida para este proyecto su valor será siempre *"YES".*
- ***<game_properties_has_control_user>:*** Afectará al tipo de movimiento del personaje principal. Este valor se rellena automáticamente con el tipo de juego escogido por el usuario. Su valor también será *"YES"* o *"NO".*
- ***<game_properties_has_chasing_finger>:*** Representa si el personaje principal seguirá el movimiento del dedo del usuario jugador por la pantalla. Su valor será *"YES"* o *"NO"* y se le asignará automáticamente dependiendo del tipo de juego elegido.
- ***<game_properties_has_target>:*** Representa si el juego tendrá meta o no. Su valor depende de lo escogido por el usuario y puede ser *"YES"* o *"NO".*
- ***<game_properties_has_reach_to_end>:*** Representa si en el juego el personaje principal perderá vidas cuando los enemigos que no sean

destruidos se escapen fuera de la pantalla de juego. Su valor puede ser *"YES"* o *"NO"* dependiendo de lo escogido por el usuario en la interfaz.

6.6.2.5 Elemento <levels>

El siguiente elemento del raíz es *<levels>* que contendrá el fragmento de la instancia del DSL correspondiente a los niveles del juego.

Por cada nivel que contenga, incluirá un subelemento *<level>* que contendrá la instancia del DSL correspondiente a un nivel determinado.

```
<levels>
    <level>
    ......//Subelementos
    </level>
    <level>
    ......//Subelementos
    </level>
</levels>
```

Ilustración 6.66. Elemento levels

6.6.2.5.1 Elemento <stage>

Es el primer subelemento de *<level>.* Este elemento solo puede aparecer una vez por nivel. En él se definirán la imagen de descripción, la de la miniatura, las de fondo y las propiedades del nivel. Se corresponde con el escenario del nivel.

```
<stage>
    <stage_description_image_source> </stage_description_image_source>
    <stage_description_image_width> </stage_description_image_width>
    <stage_description_image_height> </stage_description_image_height>
    <stage_description_image_x_position> </stage_description_image_x_position>
    <stage_description_image_y_position> </stage_description_image_y_position>
    <stage_thumbnail_image_source> </stage_thumbnail_image_source>
    <stage_thumbnail_image_width> </stage_thumbnail_image_width>
    <stage_thumbnail_image_height> </stage_thumbnail_image_height>
    <stage_background_layer_back_image_source> </stage_background_layer_back_image_source>
    <stage_background_layer_back_image_width> </stage_background_layer_back_image_width>
    <stage_background_layer_back_image_height> </stage_background_layer_back_image_height>
    <stage_background_layer_back_x_speed> </stage_background_layer_back_x_speed>
    <stage_background_layer_back_y_speed> </stage_background_layer_back_y_speed>
    <stage_background_layer_medium_image_source> </stage_background_layer_medium_image_source>
    <stage_background_layer_medium_image_width> </stage_background_layer_medium_image_width>
    <stage_background_layer_medium_image_height> </stage_background_layer_medium_image_height>
    <stage_background_layer_medium_x_speed> </stage_background_layer_medium_x_speed>
    <stage_background_layer_medium_y_speed> </stage_background_layer_medium_y_speed>
    <stage_background_layer_front_image_source> </stage_background_layer_front_image_source>
    <stage_background_layer_front_image_width> </stage_background_layer_front_image_width>
    <stage_background_layer_front_image_height> </stage_background_layer_front_image_height>
    <stage_background_layer_front_x_speed> </stage_background_layer_front_x_speed>
    <stage_background_layer_front_y_speed> </stage_background_layer_front_y_speed>
    <stage_fps> </stage_fps>
    <stage_attemps> </stage_attemps>
    <stage_points> </stage_points>
    <stage_time_to_finish> </stage_time_to_finish>
    <stage_pixels_per_meter> </stage_pixels_per_meter>
    <stage_x_speed> </stage_x_speed>
    <stage_y_speed> </stage_y_speed>
    <stage_background_music_source> </stage_background_music_source>
</stage>
```

Ilustración 6.67. Elemento stage

Y contiene los siguientes subelementos que comenzarán por *"stage_":*

- ***<stage_description_image_source>:*** Representa la ruta de imagen que contiene la descripción del nivel.
- ***<stage_description_image_width>:*** Representa el ancho de la imagen de descripción. Tiene que ser un valor entero positivo mayor que 0.
- ***<stage_description_image_height>:*** Representa el alto de la imagen de descripción. Tiene que ser un valor entero positivo mayor que 0.
- ***<stage_description_image_x_position>:*** Indica la posición en el eje X de la imagen de descripción. Tiene que ser un valor entero.
- ***<stage_description_image_y_position>:*** Indica la posición en el eje Y de la imagen de descripción. Tiene que ser un valor entero.
- ***<stage_thumbnail_image_source>:*** Representa la ruta de la imagen de la miniatura del nivel.
- ***<stage_thumbnail_image_width>:*** Representa el ancho de la miniatura del nivel. Tiene que ser un valor entero positivo mayor que 0.

- ***<stage_thumbnail_image_height>:*** Representa el alto de la miniatura del nivel. Tiene que ser un valor entero positivo mayor que 0.
- ***<stage_background_layer_back_image_source>:*** Representa la ruta de la imagen de la capa de fondo del nivel.
- ***<stage_background_layer_back_image_width>:*** Representa el ancho de la capa de fondo del nivel. Tiene que ser un valor entero positivo mayor que 0.
- ***<stage_background_layer_back_image_height>:*** Representa el alto de la capa de fondo del nivel. Tiene que ser un valor entero positivo mayor que 0.
- ***<stage_background_layer_back_x_speed>:*** Indica la velocidad a la que se mueve la capa de fondo horizontalmente. Tiene que ser un valor entero.
- ***<stage_background_layer_back_y_speed>:*** Indica la velocidad a la que se mueve la capa de fondo verticalmente. Tiene que ser un valor entero.
- ***<stage_background_layer_medium_....>*** y ***<stage_background_layer_front_.....>:*** ídem capa de fondo.
- ***<stage_fps>:*** Representa el número de frames por segundo al que se refrescarán las imágenes. Tiene que ser un valor entero positivo mayor que 0.
- ***<stage_attemps>:*** Representa el número de vidas que tendrá el personaje principal. Tiene que ser un entero positivo valor que 0.
- ***<stage_points>:*** Representa el número de puntos que una vez conseguidos te permiten pasar directamente al siguiente nivel. Tiene que ser un valor entero positivo mayor que 0.
- ***<stage_pixels_per_meter>:*** Indica el número de píxeles que equivalen a un metro. Tiene que ser un valor entero positivo mayor que 0.
- ***<stage_x_speed>:*** Representa la velocidad a la que se desplaza el escenario horizontalmente. Tiene que ser un valor entero.
- ***<stage_y_speed>:*** Representa la velocidad a la que se desplaza el escenario verticalmente. Tiene que ser un valor entero.
- ***<stage_background_music_source>:*** Representa la ruta del archivo de sonido que se reproducirá mientras se juega al nivel. Deberá estar en formato .mp3.

6.6.2.5.2 Elemento <tops>

El siguiente elemento del nivel es <*tops*> que se corresponde con los carriles que hay dentro de un nivel y por los que se moverá el personaje principal. Puede haber varios, tantos como se desee. Solamente contiene la coordenada Y donde se situará cada carril. También sirve para delimitar las coordenadas del eje entre las que se podrá desplazar el personaje principal.

```
<tops>
    <top>
        <top_y_position> </top_y_position>
    </top>
    <top>
        <top_y_position> </top_y_position>
    </top>
</tops>
```

Ilustración 6.68. Elemento tops

6.6.2.5.3 Elemento <character>

El siguiente elemento se corresponde con el personaje principal del juego. Solo hay uno por nivel. Contiene una etiqueta por cada una de las propiedades del personaje, así como para las animaciones.

```
]<character>
    <character_name> </character_name>
    <character_x_position> </character_x_position>
    <character_y_position> </character_y_position>
    <character_width> </character_width>
    <character_height> </character_height>
    <character_collision_area_left> </character_collision_area_left>
    <character_collision_area_top> </character_collision_area_top>
    <character_collision_area_right> </character_collision_area_right>
    <character_collision_area_down> </character_collision_area_down>
    <character_x_speed> </character_x_speed>
    <character_y_speed> </character_y_speed>
    <character_gravity_speed> </character_gravity_speed>
    <character_sound_die_source> </character_sound_die_source>
    <character_sound_collision_bad_item_source> </character_sound_collision_bad_item_source>
    <character_sound_collision_good_item_source> </character_sound_collision_good_item_source>
    <character_movement_type> </character_movement_type>
    <character_gravity_type> </character_gravity_type>
    <character_has_chasing_finger> </character_has_chasing_finger>
    <character_has_propulsion> </character_has_propulsion>
    <character_x_max_speed> </character_x_max_speed>
    <character_x_min_speed> </character_x_min_speed>
    <character_y_max_speed> </character_y_max_speed>
    <character_y_min_speed> </character_y_min_speed>
    <character_has_gravity> </character_has_gravity>

    <character_sprite_default_source> </character_sprite_default_source>
    <character_sprite_default_width> </character_sprite_default_width>
    <character_sprite_default_height> </character_sprite_default_height>
    <character_sprite_default_frames> </character_sprite_default_frames>
    <character_sprite_default_has_loop> </character_sprite_default_has_loop>

</character>
```

Ilustración 6.69. Elemento character

Contiene los siguientes subelementos, que comenzarán por *"character_":*

- ***<character_name>:*** Representa el nombre del personaje principal
- ***<character_x_position>:*** Representa la posición del personaje en el eje X. Tiene que ser un valor entero.
- ***<character_y_position>:*** Representa la posición del personaje en el eje Y. Tiene que ser un valor entero.
- ***<character_width>:*** Representa el ancho del personaje. Tiene que ser un valor entero positivo mayor que 0.
- ***<character_height>:*** Representa el alto del personaje. Tiene que ser un valor entero positivo mayor que 0.
- ***<character_collision_area_left>:*** Representa el tamaño del área de colisión izquierdo. Tiene que ser un valor entero.
- ***<character_collision_area_top>:*** Representa el tamaño del área de colisión superior. Tiene que ser un valor entero.
- ***<character_collision_area_right>:*** Representa el tamaño del área de colisión derecho. Tiene que ser un valor entero.
- ***<character_collision_area_down>:*** Representa el tamaño del área de colisión inferior. Tiene que ser un valor entero.
- ***<character_x_speed>:*** Representa la velocidad del personaje en el eje X. Tiene que ser un valor entero.
- ***<character_y_speed>:*** Representa la velocidad del personaje en el eje Y. Tiene que ser un valor entero.
- ***<character_gravity_speed>:*** Representa la velocidad de la gravedad del personaje. Si no tiene gravedad su valor será 0. Tiene que ser un valor entero.
- ***<character_has_chasing_finger>:*** Representa si el personaje seguirá el movimiento del dedo del usuario por la pantalla. Su valor es *"YES"* o *"NO".*
- ***<character_has_propulsión>:*** Representa si el personaje tendrá propulsión. Su valor es *"YES"* o *"NO".*
- ***<character_x_max_speed>:*** Representa la velocidad máxima del personaje en el eje X. Tiene que ser un valor entero.
- ***<character_x_min_speed>:*** Representa la velocidad mínima del personaje en el eje X. Tiene que ser un valor entero.
- ***<character_y_max_speed>:*** Representa la velocidad máxima del personaje en el eje Y. Tiene que ser un valor entero.
- ***<character_y_min_speed>:*** Representa la velocidad mínima del personaje en el eje Y. Tiene que ser un valor entero.
- ***<character_has_gravity>:*** Indica si el personaje estará afectado por la fuerza de la gravedad. Su valor es *"YES"* o *"NO".*
- ***<character_sound_die_source>:*** Representa la ruta del archivo de sonido que se reproducirá cuando el personaje principal pierda todas sus vidas. Su formato debe ser .mp3. Puede estar vacío.

- ***<character_sound_collision_bad_item_source>:*** Representa la ruta del archivo de sonido que se reproducirá cuando el personaje colisione con un ítem malo. Su formato debe ser .mp3. Puede estar vacío.
- ***<carácter_sound_collision_good_item_source>:*** Representa la ruta del archivo de sonido que se reproducirá cuando el personaje choque con un ítem bueno. Su formato deber ser .mp3. Puede estar vacío.

En cuanto a las propiedades de las animaciones del personaje, son las mismas para todas las animaciones, solamente varía el valor. Por tanto se describirán de forma general para todas las animaciones.

- ***<character_sprite_animation_source>:*** Representa la ruta de la imagen de la animación. Tiene que estar en formato .png.
- ***<character_sprite_animation_width>:*** Representa el ancho de la animación. Tiene que ser un valor entero positivo mayor que 0.
- ***<character_sprite_animation_height>:*** Representa el alto de la animación. Tiene que ser un valor entero positivo mayor que 0.
- ***<character_sprite_animation_frames>:*** Indica el número de imágenes que componen la animación. Tiene que ser un valor entero positivo mayor que 0.
- ***<character_sprite_animation_has_loop>:*** Indica si la animación se repetirá una vez tras otra. Su valor es *"YES"* o *"NO".*

Las animaciones que posee el personaje son:

- Default
- Left
- Right
- Down
- Up
- Collision bad item
- Collision good item
- Die item

6.6.2.5.4 Elemento <npcs>

Este subelemento de <level> contiene los enemigos de cada nivel. Dentro este elemento puede haber tantos enemigos (<npc>….</npc) como se desee.

```
<npcs>
    <npc>
    ....//Sublementos
    </npc>
    <npc>
    ....//Subelementos
    </npc>
</npcs>
```

Ilustración 6.70. Elemento npcs

```
<npc>
    <npc_name> </npc_name>
    <npc_x_position> </npc_x_position>
    <npc_y_position> </npc_y_position>
    <npc_width> </npc_width>
    <npc_height> </npc_height>
    <npc_collision_area_left> </npc_collision_area_left>
    <npc_collision_area_top> </npc_collision_area_top>
    <npc_collision_area_right> </npc_collision_area_right>
    <npc_collision_area_down> </npc_collision_area_down>
    <npc_x_speed> </npc_x_speed>
    <npc_y_speed> </npc_y_speed>
    <npc_is_avoidable> </npc_is_avoidable>
    <npc_died_dissapear_time> </npc_died_dissapear_time>
    <npc_give_points> </npc_give_points>
    <npc_points> </npc_points>
    <npc_sound_die_source> </npc_sound_die_source>
    <npc_sound_collision_bad_item_source> </npc_sound_collision_bad_item_source>
    <npc_sound_collision_good_item_source> </npc_sound_collision_good_item_source>
    <npc_movement_type> </npc_movement_type>
    <npc_x_max_speed> </npc_x_max_speed>
    <npc_x_min_speed> </npc_x_min_speed>
    <npc_y_max_speed> </npc_y_max_speed>
    <npc_y_min_speed> </npc_y_min_speed>
    <npc_has_gravity> </npc_has_gravity>
    <npc_gravity_type> </npc_gravity_type>
    <npc_gravity_aceleration> </npc_gravity_aceleration>
</npc>
```

Ilustración 6.71. Elemento npc

Y el subelemento *<npc>* contiene los siguientes subelementos y comienzan por *"npc_":*

- ***<npc_name>:*** Representa el nombre del enemigo.
- ***<npc_x_position>:*** Representa la posición del enemigo en el eje X. Tiene que ser un valor entero.

- ***<npc_y_position>***: Representa la posición del enemigo en el eje Y. Tiene que ser un valor entero.
- ***<npc_width>:*** Representa el ancho del enemigo. Tiene que ser un valor entero positivo mayor que 0.
- ***<npc_height>:*** Representa el alto del enemigo. Tiene que ser un valor entero positivo mayor que 0.
- ***<npc_collision_area_left>:*** Representa el tamaño del área de colisión izquierdo. Tiene que ser un valor entero.
- ***<npc_collision_area_top>:*** Representa el tamaño del área de colisión superior. Tiene que ser un valor entero.
- ***<npc_collision_area_right>:*** Representa el tamaño del área de colisión derecho. Tiene que ser un valor entero.
- ***<npc_collision_area_down>:*** Representa el tamaño del área de colisión inferior. Tiene que ser un valor entero.
- ***<npc_x_speed>:*** Representa la velocidad del enemigo en el eje X. Tiene que ser un valor entero.
- ***<npc_y_speed>:*** Representa la velocidad del enemigo en el eje Y. Tiene que ser un valor entero.
- ***<npc_is_avoidable>:*** Indica si el enemigo tiene que ser esquivado o no. Su valor será "YES" o "NO".
- ***<npc_died_dissapear_time***>: Representa el tiempo que tarda el enemigo en desaparecer. Su valor deberá ser un valor entero positivo mayor que 0.
- ***<npc_give_points >:*** Indica si el enemigo da puntos al colisionar con él. Su valor será "YES" o "NO".
- ***<npc_points>:*** Representa el número de puntos que da el enemigo al ser destruido. Deberá ser un valor entero.
- ***<npc_sound_die_source>:*** Representa la ruta del archivo de sonido que se reproducirá cuando el enemigo cuando sea destruido. Su formato debe ser .mp3. Puede estar vacío.
- ***<npc_sound_collision_bad_item_source>:*** Representa la ruta del archivo de sonido que se reproducirá cuando el enemigo colisione con el personaje principal. Su formato debe ser .mp3. Puede estar vacío.
- ***<npc_sound_collision_good_item_source***>: Representa la ruta del archivo de sonido que se reproducirá cuando el enemigo choque con un ítem bueno. Su formato deber ser .mp3. Puede estar vacío.

En cuanto a las propiedades de las animaciones del enemigo, son las mismas que para el personaje principal.

6.6.2.5.5 Elemento <target>

El último elemento de *<level>* es *<target>* que se corresponde con la meta del juego. Solamente hay una por nivel.

```
<target>
    <target_name> </target_name>
    <target_x_position> </target_x_position>
    <target_y_position> </target_y_position>
    <target_width> </target_width>
    <target_height> </target_height>
    <target_collision_area_left> </target_collision_area_left>
    <target_collision_area_top> </target_collision_area_top>
    <target_collision_area_right> </target_collision_area_right>
    <target_collision_area_down> </target_collision_area_down>
    <target_sound_on_touched_source> </target_sound_on_touched_source>
    <target_image_source> </target_image_source>
    <target_image_width> </target_image_width>
    <target_image_height> </target_image_height>
    <target_image_frames> </target_image_frames>
    <target_image_has_loop> </target_image_has_loop>
</target>
```

Ilustración 6.72. Elemento target

Los subelementos que contiene empiezan por *"target_"* y son:

- ***<target_name>:*** Representa el nombre de la meta.
- ***<target_x_position>:*** Representa la posición en el eje X de la meta. Tiene que ser un valor entero.
- ***<target_y_position>:*** Representa la posición en el eje Y de la meta. Tiene que ser un valor entero.
- ***<target_width***>: Representa el ancho de la meta. Tiene que ser un valor entero positivo mayor que 0.
- ***<target_height>:*** Representa el alto de la meta. Tiene que ser un valor entero positivo mayor que 0.
- ***<target_collision_area_..>:*** Ídem área de colisión del personaje.
- ***<target_sound_on_touched_source>:*** Representa la ruta del archivo de sonido que se reproducirá cuando el personaje principal colisione con la meta. Tendrá formato .mp3.
- ***<target_image_source>:*** Representa la ruta de la imagen de la meta. Su formato tiene que ser .png.
- ***<target_image_width>:*** Representa el ancho de la imagen de la meta. Tiene que ser un valor entero positivo mayor que 0.
- ***<target_image_height>:*** Representa el alto de la imagen de la meta. Tiene que ser un valor entero positivo mayor que 0.
- ***<target_image_frames>:*** Indica el número de imágenes que tiene la animación. Por defecto su valor es 1.
- ***<target_image_has_loop>:*** Indica si la animación de la meta se repetirá una y otra vez. Su valor por defecto es "NO".

6.7 Diseño de la plantilla

La plantilla es la base de la herramienta diseñada, ya que con ella el analizador construirá el videojuego que el usuario ha desarrollado mediante la interfaz. Se explicará aquí su funcionamiento y se hará un análisis de las carpetas que contiene. También se indicará qué partes de la plantilla son las que son variables y van a ser modificadas por el analizador.

6.7.1 Descripción y funcionamiento

La plantilla es el esqueleto de un videojuego para dispositivos Android, por lo que a efectos de programación se trata de un proyecto de Eclipse. Dentro de la plantilla existen una serie de clases que toman los valores proporcionados por la instancia del DSL, creando así la estructura correspondiente al juego diseñado con el editor. Para que el juego funcione de forma correcta, la plantilla debe tener una estructura definida, tal y como se muestra a continuación.

6.7.2 Estructura de la plantilla

La plantilla debe tener la siguiente división de carpetas:

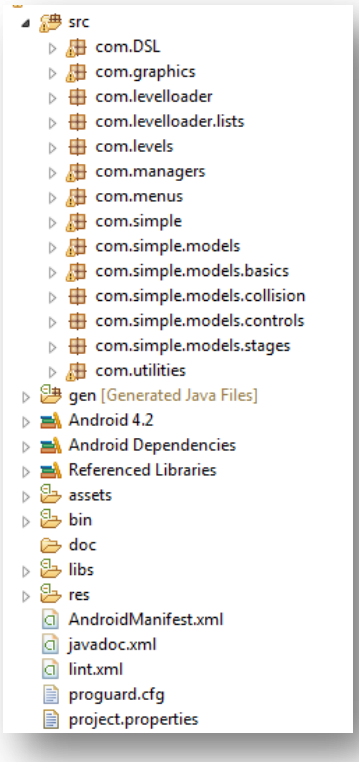


Ilustración 6.73. Estructura carpetas plantilla

Para analizar el contenido de cada carpeta se ha utilizado la siguiente tabla, en la que en la que se detalla cuál debe ser el contenido de cada subcarpeta de la plantilla.

Nombre	Descripción	Subcarpetas que contiene
assets	Contiene los niveles del juego. En ella el analizador copiará los niveles generados con la interfaz.	------
bin	Cuando se compila se hace copia de los recursos a esta carpeta.	------
gen	Contendrá la clase R, que se generará con todos los recursos que contiene el videojuego.	------
libs	Contiene las librerías necesarias para que un juego tenga publicidad.	------
res	Contiene las carpetas donde se almacenarán los recursos del videojuego.	Drawable, layout, raw, values.
drawable	Contiene las imágenes del videojuego. El analizador copiará aquí las imágenes utilizadas en la interfaz.	------
layout	Contiene los layouts que contiene cada pantalla del videojuego.	------
raw	Contiene los sonidos del videojuego. El analizador copiará aquí los recursos de sonido utilizados en la interfaz.	------
values	Contiene la clase que almacena las palabras utilizadas en los layouts.	------
src		com
com		Dsl, simple, levelLoader, Manager, levels, Utilities, Menus
DSL	Contiene la clase donde el analizador sustituirá los valores de la instancia del DSL.	------
Simple	Contiene los modelos y las vistas y actividades de cada pantalla del videojuego.	Models
Models	Contiene la clase modelo y los objetos creados construidos a partir de ese modelo.	------

levelLoader	Contiene las clases encargadas de leer los datos de los XML de los niveles.	------
Manager	Contiene la clase gestor de sonidos.	------
Graphics	Contiene la clase Sprite y Frame, utilizadas en las animaciones de los personajes.	------
levels	Contiene la clase que procesa los niveles.	------
Utilities	Contiene la clase utilidades y la clase adaptador de resolución.	------
Menus	Contiene la vista y la actividad que lanza la pantalla principal del videojuego.	------

6.7.3 Valores variables

Son los de las propiedades de los objetos, que al modificarse varían su aspecto o comportamiento, que es lo que permite crear gran variedad de juegos cambiando pocos valores.

En la clase DSL.java, que está en la carpeta "*DSL*", se encuentran los valores variables correspondientes a las pantallas de los menús y las propiedades del juego. Y tienen el siguiente aspecto:

```
public int screenOrientation = %global_screen_orientation% ;


// MenuView
public int imageButtonStart = %main_screen_start_button_image_source%;
public int heightImageButtonStart = %main_screen_start_button_image_height%;
public int widthtImageButtonStart = %main_screen_start_button_image_width%;
public int posXButtonStartMenu = %main_screen_start_button_x_position%;
public int posYButtonStartMenu = %main_screen_start_button_y_position%;
public int soundButtonStart = %main_screen_start_button_sound_source%;
public String textButtonStart = %main_screen_start_button_text_content%;
public int sizeTextButtonStart = %main_screen_start_button_text_font_size%;
public String fontTextButtonStart = %main_screen_start_button_text_font_family%;
public String colorTextButtonStart = %main_screen_start_button_text_font_color%;
```

Ilustración 6.74. Partes variables

En cuanto a los niveles, almacenados en la carpeta "*assets*", cada uno tiene un fichero asignado, que se procesará; con esos valores almacenados en el fichero de nivel se construirán

los objetos de cada nivel. Así cada personaje principal o enemigo tendrá un comportamiento u otro, dependiendo de los valores elegidos en el proceso del diseño del videojuego.

También serán diferentes los recursos de imagen y sonido. Estos se modifican con cada juego diseñado con la interfaz. Quedan almacenados en la carpeta "*drawable*" y "*raw*" respectivamente.

6.7.4 Vistas

Los diferentes elementos de cada pantalla (los botones de los menús, los enemigos de los niveles, las imágenes de fondo) se dibujan en pantalla utilizando una vista, y como cada pantalla del videojuego es diferente, cada una tiene asignada una vista.

La vista *gameView* es la que contiene la lógica del juego, que se encargará de pintar en pantalla los elementos de cada nivel y de ejecutar sus movimientos y comprobar la lógica del juego.

El resto de vistas son las encargadas de pintar en pantalla los botones y fondos de cada menú y de que cada uno realice la función que le corresponde.

Las clases que representan las vistas están almacenadas en la carpeta "*Simple*" y "*menú*". En esta segunda carpeta solamente se encuentra la correspondiente al menú principal.

6.7.5 Cargador de niveles

Para procesar los archivos XML de cada nivel que el analizador copió en la carpeta "assets" de la plantilla, se creó un cargador de niveles, que lee cada fichero y extrae los valores que contienen las etiquetas del XML y los asigna a variables, que posteriormente serán utilizadas para construir los objetos de cada nivel. Cada nivel solamente se analizará cuando se vaya a jugar, es decir, solamente se procesa uno de cada vez. Si el nivel ya se ha jugado una vez, el fichero correspondiente a ese nivel no se volverá a procesar, a menos que se salga de la aplicación del videojuego.

La clase que representa ese cargador de niveles está almacenada en la carpeta "*levelLoader*".

6.8 Especificación Técnica del Plan de Pruebas

6.8.1 Pruebas Unitarias

Las pruebas unitarias se van a realizar con la opción de test unitarios que viene integrada en el Visual Studio 2010.

Serán las pruebas que afectan a módulos independientes, como por ejemplo, configurar las propiedades globales del videojuego, configurar los menús, crear personajes del videojuego.

Se realizarán pruebas a lo largo de todo el proceso de desarrollo, comprobando su correcto funcionamiento. Para ello se utilizarán entradas de datos, tanto correctas como incorrectas. De esta forma se probará que la aplicación reacciona adecuadamente ante posibles errores que puedan cometer los usuarios.

Las pruebas que se van a realizar son las siguientes:

6.8.1.1 Configuración Propiedades Globales

Caso de Prueba 1: Configurar propiedades globales	
Entrada	**Resultado Esperado**
Se introdujo un nombre para el videojuego con tildes, paréntesis y otros caracteres especiales.	El sistema avisa al usuario de que no se pueden introducir nombres que contengan esos caracteres.
Entrada	**Resultado Esperado**
Se introdujo un nombre para el paquete con tildes, paréntesis y otros caracteres.	El sistema avisa al usuario de que no se pueden introducir nombres que contengan esos caracteres.
Entrada	**Resultado Esperado**
Se dejaron vacíos todos los campos.	El sistema aviará de qué propiedades no tienen valor.

6.8.1.2 Configuración propiedades específicas

Caso de Prueba 2: Configurar propiedades específicas	
Entrada	**Resultado Esperado**
Se eligió un tipo de juego.	El sistema asigna a la variable global que guarda el tipo de juego, el tipo de juego seleccionado.
Entrada	**Resultado Esperado**
Se escogió un valor para marcador.	El sistema asigna a las variables globales que guardan el valor de cada marcador, el valor elegido para cada uno.
Entrada	**Resultado Esperado**
Se escogió que el juego tenga meta y deje escapar a los enemigos no destruidos.	El sistema asigna a las variables globales correspondientes el valor de cada tipo de final.

6.8.1.3 Pantallas menús

Caso de Prueba 3: Pantallas menús	
Entrada	**Resultado Esperado**
Se dieron valores numéricos a aquellas propiedades que son de tipo texto.	El sistema avisa de que contiene datos incorrectos.
Entrada	**Resultado Esperado**
Se dieron valores conteniendo solo letras a aquellas propiedades que son de tipo numérico.	El sistema avisa de que contiene datos incorrectos.
Entrada	**Resultado Esperado**
Se introdujeron rutas inválidas para las imágenes y los sonidos.	El sistema avisa de que la ruta no existe.

6.8.1.4 Personaje principal

Caso de Prueba 4: Personaje principal	
Entrada	**Resultado Esperado**
Se dieron valores numéricos a aquellas propiedades que son de tipo texto.	El sistema avisa de que contiene datos incorrectos.
Entrada	**Resultado Esperado**
Se dieron valores conteniendo solo letras a aquellas propiedades que son de tipo numérico.	El sistema avisa de que contiene datos incorrectos.
Entrada	**Resultado Esperado**
Se introdujeron rutas inválidas para las imágenes y los sonidos.	El sistema avisa de que la ruta no existe.
Entrada	**Resultado Esperado**
Se creó un personaje con	El sistema no deja crear el personaje.

propiedades sin valores.	

6.8.1.5 Enemigos

Caso de Prueba 5: Enemigos	
Entrada	**Resultado Esperado**
Se dieron valores numéricos a aquellas propiedades que son de tipo texto.	El sistema avisa de que contiene datos incorrectos.
Entrada	**Resultado Esperado**
Se dieron valores conteniendo solo letras a aquellas propiedades que son de tipo numérico.	El sistema avisa de que contiene datos incorrectos.
Entrada	**Resultado Esperado**
Se introdujeron rutas inválidas para las imágenes y los sonidos.	El sistema avisa de que la ruta no existe.
Entrada	**Resultado Esperado**
Se creó un enemigo con propiedades sin valores.	El sistema no deja crear el enemigo.

6.8.1.6 Meta

Caso de Prueba 6: Meta	
Entrada	**Resultado Esperado**
Se dieron valores numéricos a aquellas propiedades que son de tipo texto.	El sistema avisa de que contiene datos incorrectos.
Entrada	**Resultado Esperado**
Se dieron valores conteniendo solo letras a aquellas propiedades que son de tipo numérico.	El sistema avisa de que contiene datos incorrectos.
Entrada	**Resultado Esperado**
Se introdujeron rutas inválidas para las imágenes y los sonidos.	El sistema avisa de que la ruta no existe.
Entrada	**Resultado Esperado**
Se introdujo una meta con propiedades sin valores.	El sistema no deja crear la meta.

6.8.1.7 Escenario

Caso de Prueba 7: Escenario	
Entrada	**Resultado Esperado**
Se dieron valores numéricos a aquellas propiedades que	El sistema avisa de que contiene datos incorrectos.

son de tipo texto.	
Entrada	**Resultado Esperado**
Se dieron valores conteniendo solo letras a aquellas propiedades que son de tipo numérico.	El sistema avisa de que contiene datos incorrectos.
Entrada	**Resultado Esperado**
Se introdujeron rutas inválidas para las imágenes y los sonidos.	El sistema avisa de que la ruta no existe.
Entrada	**Resultado Esperado**
Se introdujo un escenario sin elementos.	El sistema no deja crear el escenario.

6.8.1.8 Utilidades

Caso de Prueba 8: Utilidades	
Entrada	**Resultado Esperado**
Se introdujeron rutas inválidas para salvar las imágenes y los sonidos.	El sistema avisa de que la ruta no existe.
Entrada	**Resultado Esperado**
Se introdujeron rutas de imágenes y sonidos ya existentes para salvarlos en disco.	El sistema modifica el recurso por el nuevo.
Entrada	**Resultado Esperado**
Se crearon controles con datos vacíos para guardarlos.	El sistema avisa de que hay elementos con datos sin contenido.

6.8.2 Pruebas de Integración y del sistema

Estas se basan en el plan de pruebas especificado en el análisis.

Para estas pruebas, se hará uso de los casos que afectan a varios módulos, como el proceso completo de crear un videojuego.

6.8.3 Pruebas de Usabilidad

Las pruebas de usabilidad realizadas para esta herramienta determinan la satisfacción del usuario con el producto y por tanto son fundamentales.

Estas pruebas tratan de evaluar la aplicación en su funcionamiento real. Por tanto se ha realizado un conjunto de pruebas para verificar que las distintas partes del programa se pueden usar adecuadamente.

En aras de demostrar que la aplicación es utilizable por personas que pueden no tener conocimientos de informática, se han escogido cuatro perfiles de usuario, teniendo en cuenta su habilidad y/o conocimientos como usuarios de informática.

Estos perfiles son:

- Usuario muy familiarizado con la informática, con estudios de ingeniero técnico o superior.
- Usuario muy familiarizado con la informática, estudiantes de Ingeniería Informática.
- Usuario familiarizado con la informática, con estudios no vinculados a la informática.
- Usuario poco familiarizado con la informática, pero con estudios no vinculados a la informática.

Las pruebas se han realizado en varios lugares:

- OOTLAB, Facultad de Ciencias de Oviedo, Universidad de Oviedo.
- Otros.

La metodología que se aplicó para realizar estas pruebas fue la de dar total libertad de uso a todos los usuarios, independientemente de su perfil.

En las pruebas realizadas, a cada usuario se le proporcionó el Manual de Usuario, para consultar las dudas que les surgieran durante el proceso; y también una biblioteca de gráficos, de gran variedad, para utilizarlos en el diseño de sus videojuegos.

6.8.3.1 Diseño de Cuestionarios

6.8.3.1.1 Cuestionario de Evaluación

En cuanto al cuestionario, se optó por seguir un modelo de preguntas divididas en cuatro puntos, que a continuación se exponen:

- **1º**: **Preguntas de carácter general** acerca de la herramienta y los videojuegos.
- **2º**: **Actividades guiadas** sobre de la aplicación.
- **3º**: **Batería de preguntas cortas** sobre la herramienta.
- **4º**: **Observaciones** del usuario acerca de la aplicación.

6.8.3.1.2 Cuestionario para el Responsable de las Pruebas

En cuanto al cuestionario para el responsable de las pruebas, se optó uno en el que éste pueda anotar distintos aspectos que observe durante la realización de las pruebas.

6.8.3.2 Actividades de las Pruebas de Usabilidad

6.8.3.2.1 Preguntas de carácter general

1. ¿Usa un ordenador frecuentemente?
1. Todos los días. 2. Varias veces a la semana. 3. Ocasionalmente. 4. Nunca o casi nunca.
2. ¿Qué tipo de actividades realiza con el ordenador?
1. Es parte de mi trabajo o profesión. 2. Lo uso básicamente para ocio. 3. Solo empleo aplicaciones estilo Office. 4. Únicamente leo el correo y navego ocasionalmente.
3. ¿Es un usuario habitual de videojuegos? *[Debe señalar al menos una respuesta]
1. Sí. 2. No, pero juego algunas veces. 3. No, nunca juego en ningún dispositivo.
4. ¿Qué dispositivos prefiere a la hora de jugar a videojuegos? *[Puede señalar al más de una respuesta]
1. En mi ordenador personal. 2. En mi consola. 3. En mi dispositivo móvil. 4. En ninguno en particular.

5. ¿Está familiarizado con desarrollo de videojuegos? *[Puede señalar al más de una respuesta]

1. Sí.
2. No, pero me interesa.
3. No, no estoy familiarizado.

6. ¿Qué plataformas de desarrollo de videojuegos conoces? *[Puede señalar al más de una respuesta]

1. iOS.
2. Android.
3. Windows Phone.
4. HTML5.
5. Flash.
6. Otros.

7. ¿Conoce algún Editor o Framework de desarrollo de videojuegos similar a esta herramienta? *[Debe señalar al menos una respuesta]

1. No.
2. Otros: ……..

8. ¿Qué grado de complejidad daría al desarrollo de un videojuego con las herramientas actuales del mercado para Android? *[Debe señalar al menos una respuesta]

1. Fácil.
2. Complejo.
3. Muy complejo.

9. ¿Cómo le gustaría que fuera una herramienta para el desarrollo de videojuegos? *[Puede señalar al más de una respuesta]

1. Entorno amigable.
2. Intuitiva.
3. Dinámica.
4. Fácil de usar.
5. Multiplataforma.
6. Otros: ……..

10. ¿Qué grado de complejidad daría al desarrollo de un videojuego con la herramienta probada? *[Debe señalar al menos una respuesta]

1. Muy fácil.
2. Fácil.
3. Normal.
4. Complejo.
5. Muy complejo.

6.8.3.2.2 Actividades guiadas

A continuación, se muestra una lista de actividades parcialmente guiadas llevadas a cabo por parte de los cuatro perfiles que se han expuesto anteriormente. En dichas actividades se intentó buscar la sencillez, así como la dificultad, para obtener una valoración de extremos.

En cuanto a las actividades, se enumeran aquellas consideradas más importantes:

- Crear un videojuego desde cero.
- Añadir imágenes/sonidos al videojuego.
- Guardar/Cargar videojuego.
- Jugar al videojuego.

6.8.3.2.3 Preguntas Cortas sobre la Aplicación y Observaciones

El cuestionario referente a las preguntas cortas es el siguiente:

Facilidad de Uso	**Si**	**No**	**A veces**
¿El recorrido que se realiza por la aplicación es fácil?			
¿Las acciones que solicita la aplicación son fáciles de realizar?			
¿Sabe dónde está dentro de la aplicación?			
¿Tiene dudas acerca de cómo seguir avanzando en la aplicación?			
¿Le resulta sencillo el uso de la aplicación?			
¿Están los botones bien colocados?			
¿Se hace difícil la navegación entre pestañas?			
En general, ¿es fácil crear un videojuego?			
¿Le ha resultado sencillo configurar los menús de los videojuegos?			
En cuanto al proceso de crear un personaje principal nuevo y crear los enemigos, ¿le ha resultado sencillo?			
¿Le ha supuesto alguna dificultad crear una meta?			
En cuanto al editor de escenarios, ¿le parece fácil editar en él?			
Cuando se crea un personaje, ¿se pierde el hilo de la aplicación?			
¿Le parece fácil jugar con el videojuego?			

¿El manual de usuario soluciona los problemas que pueda tener?			
Funcionalidad	**Si**	**No**	**A veces**
¿Funciona cada tarea como Vd. Espera?			
¿El tiempo de respuesta de la aplicación es muy grande?			
¿Encuentra problemas en la creación de personajes?			
¿Encuentra problemas en la creación de escenarios?			
¿Le parece sencillo crear el personaje principal?			
¿La realización de las tareas sigue un estándar a os largo de la aplicación?			
¿Puedo encontrar fácilmente lo que quiero?			
¿Se obtiene el resultado deseado al finalizar el proceso completo?			
Al realizar alguna modificación, ¿se reflejan correctamente los cambios?			
¿Se puede navegar por cada una de las actividades realizadas durante todo el proceso de una forma fácil y ágil?			
¿Pierde el foco (del teclado/ratón) en alguna ocasión?			
Cuando juega al videojuego creado, ¿funciona como se espera?			
Cuando se produce un error o una advertencia, ¿sabe por qué se produjo?			
En el editor de escenarios, ¿le parece adecuado el uso de la propia imagen como herramienta de edición?			
¿Le resulta fácil crear un videojuego?			
¿Le ha resultado fácil jugar al videojuego?			

Contenido	**Si**	**No**
¿La distribución del contenido en la aplicación es adecuada?		
¿La información que presenta es comprensible?		
¿hay errores en el contenido que		

presenta la aplicación?		
¿Los textos y palabras utilizados se corresponden con el contenido?		
¿Los textos y palabra empleados son de fácil comprensión?		
¿Las imágenes son auto descriptivas?		
¿Los textos están bien redactados? (faltas de ortografía, oraciones mal escritas,…)		
Aspectos gráficos	**Si**	**No**
¿Los colores usados son adecuados?		
¿El tipo de letra es adecuado?		
¿El tamaño de letra es adecuado?		
¿Las imágenes utilizadas son adecuadas?		
¿Los iconos utilizados son adecuados?		
¿El tamaño y distribución de los elementos (botones, cajas de textos, etc.) es adecuado?		
¿Hay bastante contraste entre texto y fondo? (Se diferencia bien el texto del fondo)		
¿El aspecto general le parece adecuado?		

Diseño de la Interfaz	**Si**	**No**
¿Le resulta fácil de usar?		
¿El diseño de las pantallas es claro y atractivo?		
¿El contenido de las ventanas (Agrupaciones, botones, cuadros de texto, etc.) está bien estructurado?		
¿Están los botones bien ubicados?		
¿La cantidad de elementos estructurales utilizados en cada ventana es adecuada?		
¿La configuración de los distintos elementos es fácil de realizar?		
¿Los mensajes son claros?		
¿La forma de elegir los colores de fondo de las pantallas, le parece adecuada?		
¿Todo el proceso es entendible?(Pasos a seguir)		

6.8.3.2.4 Cuestionario para el Responsable de las Pruebas

Aspecto Observado	Notas
El usuario comienza a trabajar de forma rápida por las tareas	
Tiempo en realizar cada tarea	
Errores leves cometidos	
Errores graves cometidos	
El usuario sabe lo que está haciendo	
El usuario se pierde en algún punto	
El usuario crea un videojuego rápidamente	
El usuario carga correctamente imágenes y/o sonidos	
El usuario está entretenido en todo momento	
El usuario consulta el manual	
El usuario detiene su creación en un punto específico	
El usuario se lía con los botones	
El usuario se lía con el editor de los menús	
El usuario se lía con el editor del personaje principal	
El usuario se lía con el editor de los enemigos	
El usuario se lía con editor de escenarios	
El usuario se siente a gusto durante la creación	
Satisfacción del usuario mientras usa la herramienta	
Satisfacción del usuario con el resultado obtenido de su creación	

Capítulo 7. Implementación del Sistema

7.1 Lenguajes de Programación

Los lenguajes de programación utilizados para el desarrollo de este proyecto han sido los siguientes:

- Java, para los videojuegos base y para las plantillas.
- C#, para construir la interfaz gráfica y el analizador.
- XML, lenguaje para la instancia del DSL
- XAML, para la interfaz gráfica.

7.1.1 Java

Java es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

La *plataforma Java* consta de las siguientes partes:

- El lenguaje de programación mismo.
- La máquina virtual de Java o JRE, que permite la portabilidad en ejecución.
- El API Java, una biblioteca estándar para el lenguaje.

En este proyecto se ha utilizado para programar los videojuegos, recurriendo al entorno Eclipse y Android SDK ("Software Development Kit"), que es el kit de desarrollo necesario para crear aplicaciones para Android.

```
public class MenuView extends View {

    public Context context;
    public GameLoop gameloop;
    public Background background;
```

Ilustración 7.1 . Ejemplo código Java

7.1.2 C#

C# (C Sharp leído) es un lenguaje de programación orientado a objetos, desarrollado por Microsoft, que permite a los desarrolladores compilar diversas aplicaciones sólidas y seguras que se ejecutan en .NET Framework. Aunque C# forma parte de la plataforma .NET, ésta es una API, mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma.

La sintaxis de C# es muy expresiva, pero también sencilla y fácil de aprender. La sintaxis de C# está basada principalmente en aspectos tomados de C, C++ o Java.

En este proyecto se han utilizado clases que forman parte de la librería de C# del Microsoft Visual Studio 2010 para diseñar la interfaz gráfica.

```
public partial class Canvas : Window
{
  public string newCharacter = "";

    public string NewCharacter
    {
        get { return newCharacter; }
        set { newCharacter = value; }

    }
```

Ilustración 7.2 . Ejemplo código C#

7.1.3 XML

XML o "Extensible Markup Language", un lenguaje de programación extensible de etiquetas desarrollado por el W3C. Se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas.

En este proyecto se utilizará para comunicar la interfaz de usuario con el motor de transformación. A su vez, servirá como archivo portable a otros ordenadores como parte de un conjunto de ficheros de serialización, que es el fichero que se portará entre equipos y/o directorios como progreso del usuario.

```xml
<?xml version="1.0" encoding="utf-8"?>
<level>
    <stage>
        <stage_id>3</stage_id>
        <stage_description>
            <stage_description_image_source>explication_three</stage_description_image_source>
            <stage_description_image_width>299</stage_description_image_width>
            <stage_description_image_height>392</stage_description_image_height>
            <stage_description_image_x_position>15</stage_description_image_x_position>
            <stage_description_image_y_position>70</stage_description_image_y_position>
            <stage_description_text></stage_description_text>
            <stage_description_has_text>NO</stage_description_has_text>
        </stage_description>
```

Ilustración 7.3 . Ejemplo código XML

7.1.4 XAML

XAML ("Extensible Application Markup Language") es nuevo lenguaje de marcado extensible, basado en XML, para la definición de interfaces, desarrollado por Microsoft. Simplifica la creación de la UI(interfaz de usuario) para una aplicación .NET Framework. Es el lenguaje que subyace a la presentación visual de una aplicación desarrollada en Microsoft Expression Blend.

XAML forma parte de Microsoft Windows Presentation Foundation (WPF). WPF es la categoría de características de Microsoft .NET Framework 3.5 relacionadas con la presentación visual de aplicaciones basadas en Windows y de aplicaciones cliente basadas en exploradores web. WPF usa XAML para crear interfaces de usuario (UI) de una excelente calidad visual, por lo que se ha utilizado en este proyecto para diseñar las pantallas de la interfaz gráfica del usuario.

```xaml
<UserControl x:Class="OpcionesBuscar"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" Width="600" Height="230">
    <UserControl.Resources>
        <!-- Este estilo se aplicará a todos los CheckBox del control de usuario -->
        <Style TargetType="{x:Type CheckBox}">
            <Setter Property="Margin" Value="8,4,0,0"/>
        </Style>
    </UserControl.Resources>
    <StackPanel Name="StackMain" VerticalAlignment="Top">
    </StackPanel>
</UserControl>
```

Ilustración 7.4 . Ejemplo código XAML

7.2 Herramientas y Programas Usados para el Desarrollo

7.2.1 Visual Studio 2010

Microsoft Visual Studio es un entorno de desarrollo integrado para sistemas operativos Windows. Se escogió este IDE ya que la desarrolladora de este proyecto tenía experiencia previa en el diseño de aplicaciones con esta herramienta. Casi la totalidad del analizador y la interfaz de usuario se han diseñado con este entorno.

Ilustración 7.5 . Icono Visual Studio 2010

7.2.2 Eclipse

Eclipse es una plataforma de desarrollo *open source* basada en Java. Es un desarrollo de IBM cuyo código fuente fue puesto a disposición de los usuarios.

Android tiene un SDK disponible y se integra en Eclipse para programar aplicaciones para dispositivos móviles con sistema operativo Android.

Los videojuegos base se han realizado con este entorno de desarrollo.

Ilustración 7.6 . Icono Eclipse

7.3 Creación del Sistema

En este apartado se van a enumerar las clases más relevantes implementas y su información.

Además en el CD se adjuntan las clases del subsistema Plantilla documentadas con *Javadoc* y las del subsistema Interfaz y Analizador documentadas con *SandCastle*.

7.3.1 Descripción Detallada de las Clases

7.3.1.1 Clase Character

Nombre	**Tipo**	**Descripción**	**Hereda de**
Character	Privado	Para construir un objeto que represente al personaje principal del juego	
Responsabilidades			
Número	**Descripción**		
1	Construir un objeto Character con todas las propiedades que necesita.		
Métodos			
Acceso | Modo	**Tipo de Retorno**	**Nombre**	**Parámetros y tipos**
Público	String	toString	-
Atributos			
Acceso	**Modo**	**Tipo o Clase**	**Nombre**
Privado		Sprite	collisionBadItem
Privado		Sprite	collisionGoodItem
Privado		Sprite	default
Privado		Sprite	die
Privado		Sprite	down
Privado		String	gravitySpeed
Privado		String	gravityType
Privado		String	hasChasingFinger
Privado		String	hasPropulsion
Privado		String	height
Privado		String	left
Privado		String	movementType
Privado		String	name
Privado		Sprite	right
Privado		Sprite	up
Privado		String	width
Privado		String	xSpeed
Privado		String	ySpeed
Observaciones			

7.3.1.2 Clase Stage

Nombre	**Tipo**	**Descripción**	**Hereda de**
Stage	Privado	Sirve para construir un objeto escenario con todas las propiedades de un nivel.	
Responsabilidades			
Número	**Descripción**		
1	Construir un objeto Stage que contiene todas las propiedades configurables de un escenario.		
Métodos			
Acceso | Modo	**Tipo de Retorno**	**Nombre**	**Parámetros y tipos**
Público	String	toString	-
Atributos			
Acceso	**Modo**	**Tipo o Clase**	**Nombre**
Privado		String	attemps
Privado		String	back
Privado		String	backHeight
Privado		String	backWidth
Privado		String	backXSpeed
Privado		String	backYSpeed
Privado		String	description
Privado		String	descriptionHeight
Privado		String	descriptionWidth
Privado		String	descriptionXPos
Privado		String	descriptionYPos
Privado		String	front
Privado		String	frontHeight
Privado		String	frontWidth
Privado		String	frontXSpeed
Privado		String	frontYSpeed
Privado		String	medium
Privado		String	mediumHeight
Privado		String	mediumWidth
Privado		String	mediumXSpeed
Privado		String	mediumYSpeed
Privado		String	music
Privado		String	thumbnail
Privado		String	thumbnailHeight
Privado		String	thumbnailWidth
Observaciones			

7.3.1.3 Clase Store

Nombre	**Tipo**	**Descripción**	**Hereda de**
Store	Pública	Sirve para almacenar valores en tiempos de ejecución.	
Responsabilidades			
Número	**Descripción**		
1	Almacenar valores de variables mientras se está ejecutando la aplicación.		
2	Almacenar los fragmentos de código xml que genera cada pantalla.		
Métodos			
Acceso | Modo	**Tipo de Retorno**	**Nombre**	**Parámetros y tipos**
Público	void	getXMLScreens	-
Público	String	saveCharacter	String String
Público	String	saveNPC	String String
Público	String	saveTarget	String String
Atributos			
Acceso	**Modo**	**Tipo o Clase**	**Nombre**
Público		Boolean	hasLifes
Público		Boolean	hasMainCharacter
Público		Boolean	hasMeters
Público		Boolean	hasPoints
Público		Boolean	hasReachToEnd
Público		Boolean	hasTarget
Público		Int	numberImages
Público		Int	numberSounds
Público		Int	typeGame
Observaciones			

7.3.1.4 Clase Serializer

Nombre	**Tipo**	**Descripción**	**Hereda de**
Serializer	Pública	Sirve para serializar los objetos para almacenarlos en disco.	
Responsabilidades			
Número	**Descripción**		
1	Serializar un objeto y almacenarlo en disco en un fichero .dat.		
2	Des-serializar el fichero .dat y convertirlo en un objeto.		
Métodos			
Acceso | Modo	**Tipo de Retorno**	**Nombre**	**Parámetros y tipos**
Público	void	deserialize	String

			Almacen
Público	void	serialize	String Almacen
Atributos			
Acceso	**Modo**	**Tipo o Clase**	**Nombre**
Público		String	fileName
Observaciones			

7.3.1.5 Clase Almacen

Nombre	**Tipo**	**Descripción**	**Hereda de**
Almacen	Pública	Sirve para almacenar las propiedades de los elementos de las pantallas de los menús.	
Responsabilidades			
Número	**Descripción**		
1	Almacenar las propiedades de los elementos de las pantallas de los menús para guardar su estado y posteriormente cargarlo.		
Métodos			
Acceso | Modo	**Tipo de Retorno**	**Nombre**	**Parámetros y tipos**
Público	void	addObjects	Object String
Atributos			
Acceso	**Modo**	**Tipo o Clase**	**Nombre**
Público		String	height
Público		String	isVisible
Público		int	numberLastImage
Público		int	numberLastSound
Público		String	positionX
Público		String	positionY
Público		String	sourceImage
Público		String	sourceSound
Público		String	width
Observaciones			

7.3.1.6 Clase CharacterSetProperties

Nombre	**Tipo**	**Descripción**	**Hereda de**
CharacterSetProperties	Pública	Sirve para dar funcionalidad para configurar las propiedades del personaje/s principales.	Page, clase de Windows.Controls de Visual Studio 2010.
Responsabilidades			

Número	**Descripción**		
1	Funcionalidad para configurar y guardar el personaje principal y sus propiedades.		
Métodos			
Acceso | Modo	**Tipo de Retorno**	**Nombre**	**Parámetros y tipos**
Privado	void	addGameObject	-
Privado	void	generateSpriteProperties	String
Privado	void	ILoadState	-
Privado	boolean	INextPage	-
Privado	boolean	IPreviousPage	-
Privado	void	ISaveState	-
Privado	void	writeXML	-
Atributos			
Acceso	**Modo**	**Tipo o Clase**	**Nombre**
Privado	Estáticos	String	dslKey
Privado		String	key
Privado		ControlSpriteProperties	list
Observaciones			

7.3.1.7 Clase NPCProperties

Nombre	**Tipo**	**Descripción**	**Hereda de**
NPCProperties	Pública	Sirve para dar funcionalidad para configurar las propiedades de los enemigos	Page, clase de Windows.Controls de Visual Studio 2010.
Responsabilidades			
Número	**Descripción**		
1	Funcionalidad para configurar y guardar los enemigos y sus propiedades.		
Métodos			
Acceso | Modo	**Tipo de Retorno**	**Nombre**	**Parámetros y tipos**
Privado	void	addGameObject	-
Privado	void	generateSpriteProperties	String
Privado	void	ILoadState	-
Privado	boolean	INextPage	-
Privado	boolean	IPreviousPage	-
Privado	void	ISaveState	-
Privado	void	saveCurrentNPC	-
Atributos			
Acceso	**Modo**	**Tipo o Clase**	**Nombre**
Privado	Estático	String	dslKey
Privado		String	key
Privado		ControlSpriteProperties	list
Observaciones			

7.3.1.8 Clase PageMainScreen

Nombre	**Tipo**	**Descripción**	**Hereda de**
PageMainScreen	Pública	Sirve para dar funcionalidad para configurar los elementos del menú principal del juego.	Page, clase de Windows.Controls de Visual Studio 2010.
Responsabilidades			
Número	**Descripción**		
1	Funcionalidad para configurar los elementos del menú principal del videojuego.		
2	Generar el fragmento de la instancia del DSL que le corresponde.		
Métodos			
Acceso | Modo	**Tipo de Retorno**	**Nombre**	**Parámetros y tipos**
Privado	void	addImage	-
Privado	void	IloadState	-
Privado	boolean	INextPage	-
Privado	boolean	IPreviousPage	-
Privado	void	ISaveState	-
Privado	void	writeXML	-
Atributos			
Acceso	**Modo**	**Tipo o Clase**	**Nombre**
Público		Background	background
Público		Button	start
Público		Button	exit
Público		Button	options
Observaciones			

7.3.1.9 PageGlobals

Nombre	**Tipo**	**Descripción**	**Hereda de**
PageGlobals	Pública	Sirve para dar funcionalidad para configurar las propiedades globales del videojuego.	Page, clase de Windows.Controls de Visual Studio 2010.
Responsabilidades			
Número	**Descripción**		
1	Funcionalidad para configurar las propiedades globales del videojuego.		
2	Generar el fragmento de código de la instancia del DSL correspondiente.		
Métodos			
Acceso | Modo	**Tipo de Retorno**	**Nombre**	**Parámetros y tipos**
Privado	void	addIcon	-
Privado	void	addSound	-

Privado	void	IloadState	-
Privado	void	INextPage	-
Privado	void	IPreviousPage	-
Privado	void	ISaveState	-
Privado	void	PauseMusic	-
Privado	void	playMusic	-
Privado	void	StopMusic	-
Privado	void	writeXML	-
Atributos			
Acceso	**Modo**	**Tipo o Clase**	**Nombre**
Público		boolean	advertising
Público		String	googlePlayKey
Público		String	iconSource
Público		String	name
Público		String	namePackage
Público		int	screenOrientation
Público		String	soundSource
Observaciones			

7.3.1.10 PageSetGlobalsProperties

Nombre	**Tipo**	**Descripción**	**Hereda de**
PageSetGlobalsProperties	Pública	Sirve para dar funcionalidad para configurar las propiedades específicas del videojuego.	Page, clase de Windows.Controls de Visual Studio 2010.
Responsabilidades			
Número	**Descripción**		
1	Funcionalidad para configurar las propiedades específicas del videojuego.		
2	Generar el fragmento de código de la instancia del DSL correspondiente.		
Métodos			
Acceso | Modo	**Tipo de Retorno**	**Nombre**	**Parámetros y tipos**
Privado	Void	IloadState	-
Privado	boolean	INextPage	-
Privado	boolean	IPreviousPage	-
Privado	void	IsaveState	-
Privado	void	writeXML	-
Atributos			
Acceso	**Modo**	**Tipo o Clase**	**Nombre**
Observaciones			

7.3.1.11 Scenery

Nombre	**Tipo**	**Descripción**	**Hereda de**
Scenery	Pública	Sirve para dar funcionalidad para crear lo niveles del videojuego.	Page, clase de Windows.Controls de Visual Studio 2010.
Responsabilidades			
Número	**Descripción**		
1	Funcionalidad para crear y definir los niveles.		
2	Generar el fragmento de la instancia del DSL de cada elemento de cada nivel.		
Métodos			
Acceso | Modo	**Tipo de Retorno**	**Nombre**	**Parámetros y tipos**
Privado	void	ILoadState	-
Privado	boolean	INextPage	-
Privado	boolean	IPreviousPage	-
Privado	void	ISaveState	-
Privado	void	saveCurrentLevel	-
Privado	boolean	writeFilesLevel	-
Atributos			
Acceso	**Modo**	**Tipo o Clase**	**Nombre**
Público		String	attemps
Público		Background	back
Público		String	description
Público		Background	front
Público		int	indexLevel
Público		String	listLevels
Público		String	listTops
Público		Background	medium
Público		String	music
Público		int	numberLevels
Público		String	thumbnail
Observaciones			

7.3.1.12 BaseWindow

Nombre	**Tipo**	**Descripción**	**Hereda de**
BaseWindow	Pública	Será la clase que representa la ventana contenedora de la interfaz.	Window, clase de Windows.Controls de Visual Studio 2010.
Responsabilidades			
Número	**Descripción**		
1	Funcionalidad para contener las ventanas de la interfaz, el flujo de pantallas.		
Métodos			
Acceso | Modo	**Tipo de Retorno**	**Nombre**	**Parámetros y tipos**

Privado	void	goToNextPage	-
Privado	void	goToPreviousPage	-
Privado	void	guardarMenuItem	-
Privado	void	openPage	-
Privado	void	salirMenuItem	-
Atributos			
Acceso	**Modo**	**Tipo o Clase**	**Nombre**
Privado		List<string>	listWindowsNames
Privado		List<string>	listWindowsFlow
Observaciones			

7.3.1.13 Analyzer

Nombre	**Tipo**	**Descripción**	**Hereda de**
Analyzer	Publica	Será la clase encargada de analizar el fichero instancia del DSL.	
Responsabilidades			
Número	**Descripción**		
1	Analizar la instancia del DSL.		
2	Copiar los recursos y los niveles a la carpeta destino.		
3	Sustituir los valores obtenidos de la instancia del DSL en la plantilla.		
Métodos			
Acceso | Modo	**Tipo de Retorno**	**Nombre**	**Parámetros y tipos**
Público	Boolean	analyzeDSL	-
Público	Boolean	copyLevelFilesToDestiny	String
Público	Boolean	copyResources	-
Público	Void	copyToDestiny	-
Público	Void	createDestinyStructure	String
Público	void	substitute	-
Atributos			
Acceso	**Modo**	**Tipo o Clase**	**Nombre**
Público		Dictionary	Dicctionary
Público		Reader	Reader
Público		String	rutaImages
Público		String	rutaSounds
Observaciones			

7.3.1.14 Reader

Nombre	**Tipo**	**Descripción**	**Hereda de**
Reader	Pública	Es la clase responsable de chequear las rutas.	
Responsabilidades			
Número	**Descripción**		
1	Se encarga de verificar que la ruta de origen del proyecto, las de los		

	ficheros y la de la plantilla son correctas.		
Métodos			
Acceso | Modo	**Tipo de Retorno**	**Nombre**	**Parámetros y tipos**
Público	Boolean	searchRoutes	
Atributos			
Acceso	**Modo**	**Tipo o Clase**	**Nombre**
Público		String	rutaDestino
Público		String	rutaFicheros
Público		string	rutaOrigen
Observaciones			

7.3.1.15 Model

Nombre	**Tipo**	**Descripción**	**Hereda de**
Model	Pública	Representa el objeto modelo, a partir del cual se crearán los demás objetos del juego.	
Responsabilidades			
Número	**Descripción**		
1	Proporciona funcionalidad para el manejo de modelos		
Métodos			
Acceso | Modo	**Tipo de Retorno**	**Nombre**	**Parámetros y tipos**
Público	Void	drawOnScreen	Canvas
Público	Boolean	isCollided	Float float
Público	boolean	isOurOfScreen	-
Atributos			
Acceso	**Modo**	**Tipo o Clase**	**Nombre**
Público		Int	Gravity_speed
Público		Int	Height
Público		Int	Image
Público		Int	Width
Público		Int	SpriteCollisionBadItem
Público		Int	SpriteCollisionGoodItem
Público		Int	SpriteDefault
Público		Int	SpriteDieItem
Público		Int	SpriteDown
Público		Int	SpriteUp
Público		Int	SpriteLeft
Público		Int	SpriteRight
Público		Int	xSpeed
Público		int	ySpeed
Observaciones			

7.3.1.16 Collision

Nombre	Tipo	Descripción	Hereda de
Collision	Pública	Clase de implementación de modelo que representa la colisión entre dos modelos.	Modelo
Responsabilidades			
Número	**Descripción**		
1	Proporciona funcionalidad para el manejo de las colisiones entre los objetos.		
Métodos			
Acceso | Modo	**Tipo de Retorno**	**Nombre**	**Parámetros y tipos**
Atributos			
Acceso	**Modo**	**Tipo o Clase**	**Nombre**
Público		Int	Down_area
Público		Int	Left_area
Público		Int	Right_area
Público		int	Top_area
Observaciones			

7.3.1.17 Enemy

Nombre	Tipo	Descripción	Hereda de
Enemy	Pública	Clase de implementación de modelo que representa un enemigo.	Modelo
Responsabilidades			
Número	**Descripción**		
1	Proporciona funcionalidad para el manejo de los enemigos del juego.		
Métodos			
Acceso | Modo	**Tipo de Retorno**	**Nombre**	**Parámetros y tipos**
Público	Boolean	isCollided	Float float
Público	Boolean	reachToEnd	-
Público	Void	drawOnScreen	Canvas
Público	Void	Destroy	-
Público	Void	startAnimation	int
Público	Void	updateState	-
Público	void	moveAutomatically	-
Atributos			
Acceso	**Modo**	**Tipo o Clase**	**Nombre**
Público		Int	Points
Público		Int	xSpeedInScreen
Público		Int	ySpeedInScreen

Público		boolean	Give_points
Observaciones			

7.3.1.18 MenuView

Nombre	**Tipo**	**Descripción**	**Hereda de**
MenuView	Pública	Clase que permite desplegar la plantilla del menú principal del juego.	View, de Android.
Responsabilidades			
Número	**Descripción**		
1	Proporciona funcionalidad para el manejo de la pantalla del menú principal.		
Métodos			
Acceso | Modo	**Tipo de Retorno**	**Nombre**	**Parámetros y tipos**
Público	Void	Initialize	Context
Público	Void	onDraw	Canvas
Público	boolean	onTouchEvent	MotionEvent
Atributos			
Acceso	**Modo**	**Tipo o Clase**	**Nombre**
Público		Button	Exit
Público		Button	Options
Público		Button	start
Observaciones			
Cualquier comentario adicional a la clase...			

7.3.1.19 GameView

Nombre	**Tipo**	**Descripción**	**Hereda de**
GameView	Público	Clase que permite desplegar la pantalla de juego.	View de Android
Responsabilidades			
Número	**Descripción**		
1	Proporciona funcionalidad para el manejo de la pantalla de juego.		
Métodos			
Acceso | Modo	**Tipo de Retorno**	**Nombre**	**Parámetros y tipos**
Público	Void	finishLevel	-
Público	Void	glCheckCollisionWithEnemies	-
Público	Void	glMoveElements	-
Público	Void	glUpdateScoreboards	-
Público	Void	startGameOverActivity	-
Público	Void	onDraw	Canvas
Público	Void	initialize	Context
Público	boolean	onTouchEvent	MotionEvent
Atributos			
Acceso	**Modo**	**Tipo o Clase**	**Nombre**
Público		Icon	lifes

Público		Icon	Meters
Público		Icon	Points
Público		ScoreBoard	lifesS
Público		ScoreBoard	MetersS
Público		ScoreBoard	pointsS
Observaciones			

Capítulo 8. Desarrollo de las Pruebas

8.1 Pruebas Unitarias

Las pruebas unitarias realizadas para este proyecto se han llevado a cabo mediante el Visual Studio 2010 y la opción de creación de pruebas unitarias que viene integrada en el entorno.

Para ello se crea un proyecto de pruebas (*Proyect Unit Test*), y se diseña un test para cada método que se desea probar.

Se han realizado las pruebas sobre las clases que mayor funcionalidad contienen, que son las que representan las pantallas de configuración de propiedades del juego, los menús, personajes y escenarios del juego. Y la clase Utilidades, que es la que guarda los recursos utilizados en disco y salva y carga el estado de los elementos de los menús.

A continuación se detalla mediante tablas las pruebas realizadas sobre cada clase y los resultados obtenidos.

8.1.1 Configuración Propiedades Globales

Caso de Prueba 1: Configurar propiedades globales	
Entrada	**Resultado Esperado**
Se introdujo un nombre para el videojuego con tildes, paréntesis y otros caracteres especiales.	El sistema avisa al usuario de que no se pueden introducir nombres que contengan esos caracteres.
	Resultado obtenido
	El sistema avisa al usuario de que no se pueden introducir nombres que contengan esos caracteres.
Entrada	**Resultado Esperado**
Se introdujo un nombre para el paquete con tildes, paréntesis y otros caracteres.	El sistema avisa al usuario de que no se pueden introducir nombres que contengan esos caracteres.
	Resultado obtenido
	Error. El sistema no avisó al usuario de que no se pueden introducir nombres que contengan esos caracteres.
Entrada	**Resultado Esperado**
Se dejaron vacíos todos los campos.	El sistema aviará de qué propiedades no tienen valor.
	Resultado obtenido
	El sistema avisó de qué propiedades no tenían valor asignado.

8.1.2 Configuración propiedades específicas

Caso de Prueba 2: Configurar propiedades específicas	
Entrada	**Resultado Esperado**
Se eligió un tipo de juego.	El sistema asigna a la variable global que guarda el tipo de juego, el tipo de juego seleccionado.
	Resultado obtenido
	El sistema asigna a la variable global que guarda el tipo de juego, el tipo de juego seleccionado.
Entrada	**Resultado Esperado**
Se escogió un valor para marcador.	El sistema asigna a las variables globales que guardan el valor de cada marcador, el valor elegido para cada uno.
	Resultado obtenido
	El sistema asigna a las variables globales que guardan el valor de cada marcador, el valor elegido para cada uno.
Entrada	**Resultado Esperado**
Se escogió que el juego tenga meta y deje escapar a los enemigos no destruidos.	El sistema asigna a las variables globales correspondientes el valor de cada tipo de final.
	Resultado obtenido
	El sistema asigna a las variables globales correspondientes el valor de cada tipo de final.

8.1.3 Pantallas menús

Las pruebas que se ejecutaron para cada método de cada clase correspondiente a cada menú del juego y los resultados que se obtuvieron fueron los mismos, así que solamente se va a mostrar una tabla para mostrar las pruebas realizadas.

Caso de Prueba 3: Pantallas menús	
Entrada	**Resultado Esperado**
Se dieron valores numéricos a aquellas propiedades que son de tipo texto.	El sistema avisa de que contiene datos incorrectos.
	Resultado obtenido
	El sistema avisa de que contiene datos incorrectos.
Entrada	**Resultado Esperado**
Se dieron valores conteniendo solo letras a aquellas propiedades que son de tipo numérico.	El sistema avisa de que contiene datos incorrectos.
	Resultado obtenido
	El sistema avisa de que contiene datos incorrectos.
Entrada	**Resultado Esperado**
Se introdujeron rutas inválidas para las imágenes y los sonidos.	El sistema avisa de que la ruta no existe.
	Resultado obtenido
	El sistema avisa de que la ruta no existe.

8.1.4 Personaje principal

Caso de Prueba 4: Personaje principal	
Entrada	**Resultado Esperado**
Se dieron valores numéricos a aquellas propiedades que son de tipo texto.	El sistema avisa de que contiene datos incorrectos.
	Resultado obtenido
	El sistema avisa de que contiene datos incorrectos.
Entrada	**Resultado Esperado**
Se dieron valores conteniendo solo letras a aquellas propiedades que son de tipo numérico.	El sistema avisa de que contiene datos incorrectos.
	Resultado obtenido
	El sistema avisa de que contiene datos incorrectos.
Entrada	**Resultado Esperado**
Se introdujeron rutas inválidas para las imágenes y los sonidos.	El sistema avisa de que la ruta no existe.
	Resultado obtenido
	El sistema avisa de que la ruta no existe.
Entrada	**Resultado Esperado**
Se creó un personaje con propiedades sin valores.	El sistema no deja crear el personaje.
	Resultado obtenido
	Error. El sistema dejó crear el personaje.

8.1.5 Enemigos

Caso de Prueba 5: Enemigos	
Entrada	**Resultado Esperado**
Se dieron valores numéricos a aquellas propiedades que son de tipo texto.	El sistema avisa de que contiene datos incorrectos.
	Resultado obtenido
	El sistema avisa de que contiene datos incorrectos.
Entrada	**Resultado Esperado**
Se dieron valores conteniendo solo letras a aquellas propiedades que son de tipo numérico.	El sistema avisa de que contiene datos incorrectos.
	Resultado obtenido
	El sistema avisa de que contiene datos incorrectos.
Entrada	**Resultado Esperado**
Se introdujeron rutas inválidas para las imágenes y los sonidos.	El sistema avisa de que la ruta no existe.
	Resultado obtenido
	El sistema avisa de que la ruta no existe.
Entrada	**Resultado Esperado**
Se creó un enemigo con propiedades sin valores.	El sistema no deja crear el enemigo.
	Resultado obtenido
	El sistema no dejó crear el enemigo.

8.1.6 Meta

Caso de Prueba 6: Meta	
Entrada	**Resultado Esperado**
Se dieron valores numéricos a aquellas propiedades que son de tipo texto.	El sistema avisa de que contiene datos incorrectos.
	Resultado obtenido
	El sistema avisa de que contiene datos incorrectos.
Entrada	**Resultado Esperado**
Se dieron valores conteniendo solo letras a aquellas propiedades que son de tipo numérico.	El sistema avisa de que contiene datos incorrectos.
	Resultado obtenido
	El sistema avisa de que contiene datos incorrectos.
Entrada	**Resultado Esperado**
Se introdujeron rutas inválidas para las imágenes y los sonidos.	El sistema avisa de que la ruta no existe.
	Resultado obtenido
	El sistema avisa de que la ruta no existe.
Entrada	**Resultado Esperado**
Se introdujo una meta con propiedades sin valores.	El sistema no deja crear la meta.
	Resultado obtenido
	Error. El sistema dejó crear la meta.

8.1.7 Escenario

Caso de Prueba 7: Escenario	
Entrada	**Resultado Esperado**
Se dieron valores numéricos a aquellas propiedades que son de tipo texto.	El sistema avisa de que contiene datos incorrectos.
	Resultado obtenido
	El sistema avisa de que contiene datos incorrectos.
Entrada	**Resultado Esperado**
Se dieron valores conteniendo solo letras a aquellas propiedades que son de tipo numérico.	El sistema avisa de que contiene datos incorrectos.
	Resultado obtenido
	El sistema avisa de que contiene datos incorrectos.
Entrada	**Resultado Esperado**
Se introdujeron rutas inválidas para las imágenes y los sonidos.	El sistema avisa de que la ruta no existe.
	Resultado obtenido
	El sistema avisa de que la ruta no existe.
Entrada	**Resultado Esperado**
Se introdujo un escenario sin elementos.	El sistema no deja crear el escenario.
	Resultado obtenido
	Error. El sistema dejó crear el escenario.

8.1.8 Utilidades

Caso de Prueba 8: Utilidades	
Entrada	**Resultado Esperado**
Se introdujeron rutas inválidas para salvar las imágenes y los sonidos.	El sistema avisa de que la ruta no existe.
	Resultado obtenido
	El sistema avisa de que la ruta no existe.
Entrada	**Resultado Esperado**
Se introdujeron rutas de imágenes y sonidos ya existentes para salvarlos en disco.	El sistema modifica el recurso por el nuevo.
	Resultado obtenido
	Error. El sistema no modificó el recurso.
Entrada	**Resultado Esperado**
Se crearon controles con datos vacíos para guardarlos.	El sistema avisa de que hay elementos con datos sin contenido.
	Resultado obtenido
	El sistema avisa de que hay elementos con datos sin contenido.

8.2 Pruebas de Integración y del Sistema

8.2.1.1 Caso de uso: Crear nuevo videojuego

8.2.1.1.1 Subcaso de uso: Configurar propiedades globales

Caso de Uso 1: Crear nuevo videojuego. Subcaso: Configurar propiedades globales	
Prueba	**Resultado Esperado**
Introducir un nombre para el videojuego con tildes, paréntesis y otros caracteres especiales.	El sistema avisa al usuario de que no se pueden introducir nombres que contengan esos caracteres.
	Resultado obtenido
	El sistema avisa al usuario de que no se pueden introducir nombres que contengan esos caracteres.
Prueba	**Resultado Esperado**
Introducir un nombre para el videojuego que solo contenga números.	Se asigna ese nombre introducido al nombre del videojuego y al paquete.
	Resultado obtenido
	Se asigna ese nombre introducido al nombre del videojuego y al paquete.
Prueba	**Resultado Esperado**
Introducir un nombre para el videojuego que solo contenga letras.	Se asigna ese nombre introducido al nombre del videojuego y al paquete.
	Resultado obtenido
	Se asigna ese nombre introducido al nombre del videojuego y al paquete.

Prueba	**Resultado Esperado**
Introducir un nombre para el videojuego que contenga letras y números.	Se asigna ese nombre introducido al nombre del videojuego y al paquete.
	Resultado obtenido
	Se asigna ese nombre introducido al nombre del videojuego y al paquete.
Prueba	**Resultado Esperado**
Se intenta modificar el nombre del paquete introduciendo tildes, paréntesis y otros caracteres especiales.	El sistema avisa al usuario de que no se puede introducir un nombre para el paquete que contenga esos caracteres.
	Resultado obtenido
	Error. El sistema no ha avisado. **Solución**: se ha añadido el código necesario para controlar ese caso. **Corregido**. El sistema avisa al usuario de que no se puede introducir un nombre para el paquete que contenga esos caracteres.
Prueba	**Resultado Esperado**
Se intenta modificar el nombre del paquete introduciendo números y letras.	Se asigna ese nombre introducido al nombre del paquete.
	Resultado obtenido
	Se asigna ese nombre introducido al nombre del paquete.
Prueba	**Resultado Esperado**
Se intenta elegir un icono para el videojuego con formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger el icono tiene un filtro, así que solamente mostrará imágenes con esta extensión.
	Resultado obtenido
	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger el icono tiene un filtro, así que solamente mostrará imágenes con esta extensión.
Prueba	**Resultado Esperado**
Se elige un icono para el videojuego con extensión .png.	Se mostrará una vista previa en la pantalla el icono seleccionado. Y se le asignará al videojuego.
	Resultado obtenido
	Se mostró una vista previa en la pantalla el icono seleccionado. Y se le asigna al videojuego.
Prueba	**Resultado Esperado**
Se marca la opción de publicidad y no se añade ningún ID.	El sistema avisará de que se tiene que introducir un ID.
	Resultado obtenido
	El sistema avisó de que se tiene que introducir un ID.
Prueba	**Resultado Esperado**
Se marcar la opción de publicidad y se introduce un ID.	Se asigna ese valor al identificador de la publicidad
	Resultado obtenido
	Se asigna ese valor al identificador de la publicidad
Prueba	**Resultado Esperado**
Se escoge una orientación para el videojuego.	Se asigna esa orientación al videojuego.
	Resultado obtenido
	Se asigna esa orientación al videojuego.
Prueba	**Resultado Esperado**

Se pulsa sobre el botón de reproducir sin haber introducido un sonido.	El sistema no reproducirá ningún sonido.
	Resultado obtenido
	El sistema no reprodujo ningún sonido.
Prueba	**Resultado Esperado**
Se intenta elegir una melodía para los menús del juego con formato .mp4.	El sistema no permite seleccionar archivos de sonido con formato distinto a .mp3. El explorador de carpetas para escoger el icono tiene un filtro, así que solamente mostrará archivos de sonido con esta extensión.
	Resultado obtenido
	Error. Permite elegir archivos con cualquier extensión. **Solución**: se aplicó el filtro al cuadro de diálogo. **Corregido**. El sistema no permite seleccionar archivos de sonido con formato distinto a .mp3
Prueba	**Resultado Esperado**
Se selecciona un archivo de sonido en formato .mp3.	Se asigna ese sonido a la música de fondo de los menús.
	Resultado obtenido
	Se asigna ese sonido a la música de fondo de los menús.
Prueba	**Resultado Esperado**
Se pulsa sobre el botón de reproducir después de elegir una melodía.	El sistema reproducirá ese sonido.
	Resultado obtenido
	El sistema reprodujo ese sonido.
Prueba	**Resultado Esperado**
Se pulsa sobre el botón de pausa después de elegir una melodía.	El sistema pausará la melodía.
	Resultado obtenido
	El sistema pausó la melodía.
Prueba	**Resultado Esperado**
Se pulsa sobre el botón de reproducir después de haber pausado la reproducción.	El sistema reanudará la reproducción.
	Resultado obtenido
	El sistema reanudó la reproducción.
Prueba	**Resultado Esperado**
Se pulsa sobre el botón de parar la melodía.	El sistema parará la reproducción.
	Resultado obtenido
	El sistema paró la reproducción.
Prueba	**Resultado Esperado**
Se pulsa sobre el botón de reproducir después de haber parado la reproducción.	El sistema iniciará la reproducción.
	Resultado obtenido
	El sistema inició la reproducción.
Prueba	**Resultado Esperado**
Se pulsa sobre el botón "Siguiente".	En la ruta donde se eligió que se iba a almacenar el juego, se creará una carpeta con el nombre del proyecto, y unas subcarpetas dentro llamadas "images", "sounds" y "levels"
	Resultado obtenido
	En la ruta donde se eligió que se iba a almacenar el juego, se creó una carpeta con el nombre del proyecto, y unas subcarpetas dentro llamadas "images", "sounds" y "levels"

8.2.1.1.2 Subcaso de uso: Configurar propiedades específicas

Caso de Uso 1: Crear nuevo videojuego. Subcaso: Configurar propiedades específicas	
Prueba	**Resultado Esperado**
Se elige el tipo de juego "Sigue el dedo" haciendo clic sobre la imagen correspondiente.	Este tipo de juego queda seleccionado y el resto de animaciones de los otros tipos de juegos quedan pausadas.
	Resultado obtenido
	Este tipo de juego queda seleccionado y el resto de animaciones de los otros tipos de juegos quedan pausadas.
Prueba	**Resultado Esperado**
Se elige el tipo de juego "JetPack" haciendo clic sobre la imagen correspondiente.	Este tipo de juego queda seleccionado y el resto de animaciones de los otros tipos de juegos quedan pausadas.
	Resultado obtenido
	Este tipo de juego queda seleccionado y el resto de animaciones de los otros tipos de juegos quedan pausadas.
Prueba	**Resultado Esperado**
Se elige el tipo de juego "Carriles" haciendo clic sobre la imagen correspondiente.	Este tipo de juego queda seleccionado y el resto de animaciones de los otros tipos de juegos quedan pausadas.
	Resultado obtenido
	Este tipo de juego queda seleccionado y el resto de animaciones de los otros tipos de juegos quedan pausadas.
Prueba	**Resultado Esperado**
Se escoge un valor para el marcador de vidas. Si o No.	Si es escogió si el sistema registrará que el juego tendrá vidas, si escogió no, registrará que no.
	Resultado obtenido
	Se escogió "Si" y se registró en el sistema.
Prueba	**Resultado Esperado**
Se escoge un valor para el marcador de puntos. Si o No.	Si es escogió si el sistema registrará que el juego tendrá puntos, si escogió no, registrará que no.
	Resultado obtenido
	Se escogió "No" y se registró en el sistema.
Prueba	**Resultado Esperado**
Se escoge un valor para el marcador de metros. Si o No.	Si es escogió si el sistema registrará que el juego tendrá metros, si escogió no, registrará que no.
	Resultado obtenido
	Se escogió "Si" y se registró en el sistema.
Prueba	**Resultado Esperado**
Se escogerá la opción de que el juego tenga meta y que se dejen escapar a los enemigos sin restar vidas.	El sistema registrará que el juego tendrá meta y los enemigos no destruidos que se escapen de la pantalla no restarán vidas al personaje principal.
	Resultado obtenido
	El sistema registró que el juego tendrá meta y los enemigos no destruidos que se escapen de la pantalla no restarán vidas al personaje principal.
Prueba	**Resultado Esperado**
Se escogerá la opción de que el juego no tenga meta y que no se dejen escapar a los enemigos sin restar vidas.	El sistema registrará que el juego no tendrá meta y los enemigos no destruidos que se escapen de la pantalla restarán vidas al personaje principal.
	Resultado obtenido
	El sistema registró que el juego no tendrá meta y los

	enemigos no destruidos que se escapen de la pantalla restarán vidas al personaje principal.
Prueba	**Resultado Esperado**
Se escogerá la opción de que el juego tenga meta y que no se dejen escapar a los enemigos sin restar vidas.	El sistema registrará que el juego tendrá meta y los enemigos no destruidos que se escapen de la pantalla restarán vidas al personaje principal.
	Resultado obtenido
	El sistema registró que el juego tendrá meta y los enemigos no destruidos que se escapen de la pantalla restarán vidas al personaje principal.
Prueba	**Resultado Esperado**
Se escogerá la opción de que el juego no tenga meta y que se dejen escapar a los enemigos sin restar vidas.	El sistema avisará de que se tiene que elegir una de las dos opciones para el final del nivel del juego.
	Resultado obtenido
	El sistema avisó de que se tiene que elegir una de las dos opciones para el final del nivel del juego.

8.2.1.1.3 Subcaso de uso: Configurar pantalla principal

Caso de Uso 1: Crear nuevo videojuego. Subcaso: Configurar pantalla principal	
Prueba	**Resultado Esperado**
Se selecciona el fondo y se intenta modificar su imagen con una en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
	Resultado obtenido
	El sistema no permite seleccionar imágenes con formato distinto a .png.
Prueba	**Resultado Esperado**
Se selecciona una imagen para el fondo en formato .png.	Se mostrará el fondo de la pantalla con la nueva imagen elegida.
	Resultado obtenido
	Se mostró el fondo de la pantalla con la nueva imagen elegida.
Prueba	**Resultado Esperado**
Con la imagen de fondo seleccionada se modifica su posición X y Y.	El fondo de pantalla se desplazará automáticamente a su nueva posición.
	Resultado obtenido
	El fondo de pantalla se desplazó automáticamente a su nueva posición.
Prueba	**Resultado Esperado**
Se selecciona uno de los botones.	El botón seleccionado aparecerá rodeado por un rectángulo de color verde.
	Resultado obtenido
	El botón seleccionado apareció rodeado por un rectángulo de color verde.
Prueba	**Resultado Esperado**
Con un botón seleccionado se modifica su ancho y su	El botón modificará su tamaño automáticamente.
	Resultado obtenido

alto.	**Error**. El botón modificó el ancho con el valor del alto y viceversa. **Solución**: se arregló asignando a cada uno el método que le correspondía. **Corregido**. El botón modificó su tamaño automáticamente.
Prueba	**Resultado Esperado**
Con un botón seleccionado se modifica su posición X e Y.	El botón modificará su posición en la pantalla automáticamente.
	Resultado obtenido
	El botón modificó su posición en la pantalla automáticamente.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se intenta modificar su imagen con una en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
	Resultado obtenido
	El sistema no permitió seleccionar imágenes con formato distinto a .png.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen para el fondo en formato .png.	Se mostrará el botón con la nueva imagen elegida.
	Resultado obtenido
	Se mostró el botón con la nueva imagen elegida.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen de menor tamaño que la actual.	El botón cambiará su imagen y su tamaño por los de la nueva imagen.
	Resultado obtenido
	El botón cambió su imagen y su tamaño por los de la nueva imagen.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen de mayor tamaño que la actual.	El botón cambiará su imagen y su tamaño por los de la nueva imagen.
	Resultado obtenido
	El botón cambió su imagen y su tamaño por los de la nueva imagen.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se intenta elegir un sonido para que se reproduzca cuando se pulse el botón en el videojuego con formato .mp4.	El sistema no permite seleccionar archivos de sonido con formato distinto a .mp3. El explorador de carpetas para escoger lo tiene un filtro, así que solamente mostrará archivos de sonido con esta extensión.
	Resultado obtenido
	El sistema no permitió seleccionar archivos de sonido con formato distinto a .mp3.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona un sonido para que se reproduzca cuando se pulse el botón en el videojuego en formato .mp3.	Se asigna ese sonido al botón, para que se reproduzca cuando se pulse.
	Resultado obtenido
	Se asignó ese sonido al botón, para que se reproduzca cuando se pulse.

Prueba	**Resultado Esperado**
Con un botón seleccionado, se marca la opción de ocultar.	Ese botón se mostrará con una transparencia muy alta, pero sin llegar a desaparecer por si se quiere des ocultar. Este botón no aparecerá en el videojuego.
	Resultado obtenido
	Ese botón se mostró con una transparencia muy alta, pero sin llegar a desaparecer por si se quiere des ocultar. Este botón no apareció en el videojuego.
Prueba	**Resultado Esperado**
Con un botón seleccionado y que se ha ocultado previamente, se desmarca la opción de ocultar.	Se vuelve a mostrar el botón como estaba originalmente. El botón aparecerá en el juego.
	Resultado obtenido
	Se vuelve a mostrar el botón como estaba originalmente. El botón apareció en el juego.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se desplaza este por la pantalla.	El botón se moverá mientras se tenga pulsado el botón izquierdo del ratón. Cuando se deje de pulsar el botón dejará de moverse y quedará en esa posición. En los campos de texto de Ancho y Alto se actualizará el valor de sus coordenadas mientras se esté desplazando
	Resultado obtenido
	El botón se movió mientras se tuvo pulsado el botón izquierdo del ratón. Cuando se dejó de pulsar el botón dejó de moverse y quedó en esa posición. En los campos de texto de Ancho y Alto se actualizó el valor de sus coordenadas mientras se estuvo desplazando

8.2.1.1.4 Subcaso de uso: Configurar pantalla opciones

Caso de Uso 1: Crear nuevo videojuego. Subcaso: Configurar pantalla opciones	
Prueba	**Resultado Esperado**
Se selecciona el fondo y se intenta modificar su imagen con una en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
	Resultado obtenido
	El sistema no permitió seleccionar imágenes con formato distinto a .png.
Prueba	**Resultado Esperado**
Se selecciona una imagen para el fondo en formato .png.	Se mostrará el fondo de la pantalla con la nueva imagen elegida.
	Resultado obtenido
	Se mostró el fondo de la pantalla con la nueva imagen elegida.
Prueba	**Resultado Esperado**
Con la imagen de fondo seleccionada se modifica su posición X y Y.	El fondo de pantalla se desplazará automáticamente a su nueva posición.
	Resultado obtenido
	El fondo de pantalla se desplazó automáticamente a su

	nueva posición.
Prueba	**Resultado Esperado**
Se selecciona uno de los botones.	El botón seleccionado aparecerá rodeado por un rectángulo de color verde.
	Resultado obtenido
	El botón seleccionado apareció rodeado por un rectángulo de color verde.
Prueba	**Resultado Esperado**
Con un botón seleccionado se modifica su ancho y su alto.	El botón modificará su tamaño automáticamente.
	Resultado obtenido
	El botón modificó su tamaño automáticamente.
Prueba	**Resultado Esperado**
Con un botón seleccionado se modifica su posición X e Y.	El botón modificará su posición en la pantalla automáticamente.
	Resultado obtenido
	El botón modificó su posición en la pantalla automáticamente.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se intenta modificar su imagen con una en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
	Resultado obtenido
	El sistema no permite seleccionar imágenes con formato distinto a .png
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen para el fondo en formato .png.	Se mostrará el botón con la nueva imagen elegida.
	Resultado obtenido
	Se mostró el botón con la nueva imagen elegida.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen de menor tamaño que la actual.	El botón cambiará su imagen y su tamaño por los de la nueva imagen.
	Resultado obtenido
	El botón cambió su imagen y su tamaño por los de la nueva imagen.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen de mayor tamaño que la actual.	El botón cambiará su imagen y su tamaño por los de la nueva imagen.
	Resultado obtenido
	El botón cambió su imagen y su tamaño por los de la nueva imagen.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se intenta elegir un sonido para que se reproduzca cuando se pulse el botón en el videojuego con formato .mp4.	El sistema no permite seleccionar archivos de sonido con formato distinto a .mp3. El explorador de carpetas para escoger lo tiene un filtro, así que solamente mostrará archivos de sonido con esta extensión.
	Resultado obtenido
	El sistema no permitió seleccionar archivos de sonido con formato distinto a .mp3

Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona un sonido para que se reproduzca cuando se pulse el botón en el videojuego en formato .mp3.	Se asigna ese sonido al botón, para que se reproduzca cuando se pulse.
	Resultado obtenido
	Se asignó ese sonido al botón, para que se reproduzca cuando se pulse.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se marca la opción de ocultar.	Ese botón se mostrará con una transparencia muy alta, pero sin llegar a desaparecer por si se quiere des ocultar. Este botón no aparecerá en el videojuego.
	Resultado obtenido
	Ese botón se mostró con una transparencia muy alta, pero sin llegar a desaparecer por si se quiere des ocultar. Este botón no apareció en el videojuego.
Prueba	**Resultado Esperado**
Con un botón seleccionado y que se ha ocultado previamente, se desmarca la opción de ocultar.	Se vuelve a mostrar el botón como estaba originalmente. El botón aparecerá en el juego.
	Resultado obtenido
	Se vuelve a mostrar el botón como estaba originalmente. El botón apareció en el juego.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se desplaza este por la pantalla.	El botón se moverá mientras se tenga pulsado el botón izquierdo del ratón. Cuando se deje de pulsar el botón dejará de moverse y quedará en esa posición. En los campos de texto de Ancho y Alto se actualizará el valor de sus coordenadas mientras se esté desplazando.
	Resultado obtenido
	El botón se movió mientras se tuvo pulsado el botón izquierdo del ratón. Cuando se dejó de pulsar el botón dejó de moverse y quedó en esa posición. En los campos de texto de Ancho y Alto se actualizó el valor de sus coordenadas mientras se estuvo desplazando.

8.2.1.1.5 Subcaso de uso: Configurar pantalla selección de nivel

Caso de Uso 1: Crear nuevo videojuego. Subcaso: Configurar pantalla selección de nivel	
Prueba	**Resultado Esperado**
Se selecciona el fondo y se intenta modificar su imagen con una en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
	Resultado obtenido
	El sistema no permite seleccionar imágenes con formato distinto a .png.
Prueba	**Resultado Esperado**
Se selecciona una imagen para el fondo en formato .png.	Se mostrará el fondo de la pantalla con la nueva imagen elegida.
	Resultado obtenido
	Se mostró el fondo de la pantalla con la nueva imagen

	elegida.
Prueba	**Resultado Esperado**
Con la imagen de fondo seleccionada se modifica su posición X y Y.	El fondo de pantalla se desplazará automáticamente a su nueva posición.
	Resultado obtenido
	El fondo de pantalla se desplazó automáticamente a su nueva posición.
Prueba	**Resultado Esperado**
Se selecciona uno de los botones.	El botón seleccionado aparecerá rodeado por un rectángulo de color verde.
	Resultado obtenido
	El botón seleccionado apareció rodeado por un rectángulo de color verde.
Prueba	**Resultado Esperado**
Con un botón seleccionado se modifica su ancho y su alto.	El botón modificará su tamaño automáticamente.
	Resultado obtenido
	El botón modificó su tamaño automáticamente.
Prueba	**Resultado Esperado**
Con un botón seleccionado se modifica su posición X e Y.	El botón modificará su posición en la pantalla automáticamente.
	Resultado obtenido
	El botón modificó su posición en la pantalla automáticamente.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se intenta modificar su imagen con una en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
	Resultado obtenido
	El sistema no permite seleccionar imágenes con formato distinto a .png.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen para el fondo en formato .png.	Se mostrará el botón con la nueva imagen elegida.
	Resultado obtenido
	Se mostró el botón con la nueva imagen elegida.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen de menor tamaño que la actual.	El botón cambiará su imagen y su tamaño por los de la nueva imagen.
	Resultado obtenido
	El botón cambió su imagen y su tamaño por los de la nueva imagen.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen de mayor tamaño que la actual.	El botón cambiará su imagen y su tamaño por los de la nueva imagen.
	Resultado obtenido
	El botón cambió su imagen y su tamaño por los de la nueva imagen.
Prueba	**Resultado Esperado**
Con un botón seleccionado,	El sistema no permite seleccionar archivos de sonido con

se intenta elegir un sonido para que se reproduzca cuando se pulse el botón en el videojuego con formato .mp4.	formato distinto a .mp3. El explorador de carpetas para escoger lo tiene un filtro, así que solamente mostrará archivos de sonido con esta extensión.
	Resultado obtenido
	El sistema no permite seleccionar archivos de sonido con formato distinto a .mp3.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona un sonido para que se reproduzca cuando se pulse el botón en el videojuego en formato .mp3.	Se asigna ese sonido al botón, para que se reproduzca cuando se pulse.
	Resultado obtenido
	Se asignó ese sonido al botón, para que se reproduzca cuando se pulse.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se marca la opción de ocultar.	Ese botón se mostrará con una transparencia muy alta, pero sin llegar a desaparecer por si se quiere des ocultar. Este botón no aparecerá en el videojuego.
	Resultado obtenido
	Ese botón se mostró con una transparencia muy alta, pero sin llegar a desaparecer por si se quiere des ocultar. Este botón no apareció en el videojuego.
Prueba	**Resultado Esperado**
Con un botón seleccionado y que se ha ocultado previamente, se desmarca la opción de ocultar.	Se vuelve a mostrar el botón como estaba originalmente. El botón aparecerá en el juego.
	Resultado obtenido
	Se vuelve a mostrar el botón como estaba originalmente. El botón apareció en el juego
Prueba	**Resultado Esperado**
Con un botón seleccionado, se desplaza este por la pantalla.	El botón se moverá mientras se tenga pulsado el botón izquierdo del ratón. Cuando se deje de pulsar el botón dejará de moverse y quedará en esa posición. En los campos de texto de Ancho y Alto se actualizará el valor de sus coordenadas mientras se esté desplazando
	Resultado obtenido
	El botón se movió mientras se tuvo pulsado el botón izquierdo del ratón. Cuando se dejó de pulsar el botón dejó de moverse y quedó en esa posición. En los campos de texto de Ancho y Alto se actualizó el valor de sus coordenadas mientras se estuvo desplazando
Prueba	**Resultado Esperado**
Se selecciona la imagen de miniatura del nivel y se intenta modificar su imagen.	El sistema inhabilita el botón de añadir imagen, ya que la imagen de la miniatura del nivel solamente se podrá elegir cuando se defina el nivel.
	Resultado obtenido
	El sistema inhabilitó el botón de añadir imagen.

8.2.1.1.6 Subcaso de uso: Configurar pantalla carga

Caso de Uso 1: Crear nuevo videojuego. Subcaso: Configurar pantalla carga	
Prueba	**Resultado Esperado**
Se selecciona el fondo y se intenta modificar su imagen con una en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
	Resultado obtenido
	El sistema no permitió seleccionar imágenes con formato distinto a .png
Prueba	**Resultado Esperado**
Se selecciona una imagen para el fondo en formato .png.	Se mostrará el fondo de la pantalla con la nueva imagen elegida.
	Resultado obtenido
	Se mostró el fondo de la pantalla con la nueva imagen elegida.
Prueba	**Resultado Esperado**
Con la imagen de fondo seleccionada se modifica su posición X y Y.	El fondo de pantalla se desplazará automáticamente a su nueva posición.
	Resultado obtenido
	El fondo de pantalla se desplazó automáticamente a su nueva posición.

8.2.1.1.7 Subcaso de uso: Configurar pantalla pausa juego

Caso de Uso 1: Crear nuevo videojuego. Subcaso: Configurar pantalla juego	
Prueba	**Resultado Esperado**
Se selecciona el fondo y se intenta modificar su imagen con una en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
	Resultado obtenido
	El sistema no permitió seleccionar imágenes con formato distinto a .png.
Prueba	**Resultado Esperado**
Se selecciona una imagen para el fondo en formato .png.	Se mostrará el fondo de la pantalla con la nueva imagen elegida.
	Resultado obtenido
	Se mostró el fondo de la pantalla con la nueva imagen elegida.
Prueba	**Resultado Esperado**
Con la imagen de fondo seleccionada se modifica su posición X y Y.	El fondo de pantalla se desplazará automáticamente a su nueva posición.
	Resultado obtenido
	El fondo de pantalla se desplazó automáticamente a su nueva posición.
Prueba	**Resultado Esperado**
Se selecciona uno de los	El botón seleccionado aparecerá rodeado por un rectángulo

botones.	de color verde.
	Resultado obtenido
	El botón seleccionado apareció rodeado por un rectángulo de color verde.
Prueba	**Resultado Esperado**
Con un botón seleccionado se modifica su ancho y su alto.	El botón modificará su tamaño automáticamente.
	Resultado obtenido
	El botón modificó su tamaño automáticamente.
Prueba	**Resultado Esperado**
Con un botón seleccionado se modifica su posición X e Y.	El botón modificará su posición en la pantalla automáticamente.
	Resultado obtenido
	El botón modificó su posición en la pantalla automáticamente.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se intenta modificar su imagen con una en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
	Resultado obtenido
	El sistema no permitió seleccionar imágenes con formato distinto a .png.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen para el fondo en formato .png.	Se mostrará el botón con la nueva imagen elegida.
	Resultado obtenido
	Se mostró el botón con la nueva imagen elegida.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen de menor tamaño que la actual.	El botón cambiará su imagen y su tamaño por los de la nueva imagen.
	Resultado obtenido
	El botón cambió su imagen y su tamaño por los de la nueva imagen.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen de mayor tamaño que la actual.	El botón cambiará su imagen y su tamaño por los de la nueva imagen.
	Resultado obtenido
	El botón cambió su imagen y su tamaño por los de la nueva imagen.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se intenta elegir un sonido para que se reproduzca cuando se pulse el botón en el videojuego con formato .mp4.	El sistema no permite seleccionar archivos de sonido con formato distinto a .mp3. El explorador de carpetas para escoger lo tiene un filtro, así que solamente mostrará archivos de sonido con esta extensión.
	Resultado obtenido
	El sistema no permitió seleccionar archivos de sonido con formato distinto a .mp3.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona un sonido para	Se asigna ese sonido al botón, para que se reproduzca cuando se pulse.

que se reproduzca cuando se pulse el botón en el videojuego en formato .mp3.	**Resultado obtenido**
	Se asignó ese sonido al botón, para que se reproduzca cuando se pulse.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se marca la opción de ocultar.	Ese botón se mostrará con una transparencia muy alta, pero sin llegar a desaparecer por si se quiere des ocultar. Este botón no aparecerá en el videojuego.
	Resultado obtenido
	Ese botón se mostró con una transparencia muy alta, pero sin llegar a desaparecer por si se quiere des ocultar. Este botón no apareció en el videojuego
Prueba	**Resultado Esperado**
Con un botón seleccionado y que se ha ocultado previamente, se desmarca la opción de ocultar.	Se vuelve a mostrar el botón como estaba originalmente. El botón aparecerá en el juego.
	Resultado obtenido
	Se vuelve a mostrar el botón como estaba originalmente. El botón apareció en el juego.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se desplaza este por la pantalla.	El botón se moverá mientras se tenga pulsado el botón izquierdo del ratón. Cuando se deje de pulsar el botón dejará de moverse y quedará en esa posición. En los campos de texto de Ancho y Alto se actualizará el valor de sus coordenadas mientras se esté desplazando
	Resultado obtenido
	El botón se movió mientras se tuvo pulsado el botón izquierdo del ratón. Cuando se dejó de pulsar el botón dejó de moverse y quedó en esa posición. En los campos de texto de Ancho y Alto se actualizó el valor de sus coordenadas mientras se estuvo desplazando.

8.2.1.1.8 Subcaso de uso: Configurar pantalla juego

Caso de Uso 1: Crear nuevo videojuego. Subcaso: Configurar pantalla juego	
Prueba	**Resultado Esperado**
Se selecciona el fondo y se intenta modificar su imagen con una en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
	Resultado obtenido
	El sistema no permitió seleccionar imágenes con formato distinto a .png
Prueba	**Resultado Esperado**
Se selecciona una imagen para el fondo en formato .png.	Se mostrará el fondo de la pantalla con la nueva imagen elegida.
	Resultado obtenido
	Se mostró el fondo de la pantalla con la nueva imagen elegida.
Prueba	**Resultado Esperado**
Con la imagen de fondo	El fondo de pantalla se desplazará automáticamente a su

seleccionada se modifica su posición X y Y.	nueva posición.
	Resultado obtenido
	El fondo de pantalla se desplazó automáticamente a su nueva posición.
Prueba	**Resultado Esperado**
Se selecciona uno de los elementos de la barra de estado.	El elemento seleccionado aparecerá rodeado por un rectángulo de color verde.
	Resultado obtenido
	El elemento seleccionado apareció rodeado por un rectángulo de color verde.
Prueba	**Resultado Esperado**
Con un elemento seleccionado se modifica su ancho y su alto.	El elemento modificará su tamaño automáticamente.
	Resultado obtenido
	El elemento modificó su tamaño automáticamente.
Prueba	**Resultado Esperado**
Con un elemento seleccionado se modifica su posición X e Y.	El elemento modificará su posición en la pantalla automáticamente.
	Resultado obtenido
	El elemento modificó su posición en la pantalla automáticamente.
Prueba	**Resultado Esperado**
Con un elemento seleccionado, se intenta modificar su imagen con una en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión. Si se trata de un marcador, el botón de añadir imagen se deshabilitará al pulsar sobre el marcador.
	Resultado obtenido
	El sistema no permitió seleccionar imágenes con formato distinto a .png.
Prueba	**Resultado Esperado**
Con un elemento seleccionado, se selecciona una imagen para el fondo en formato .png.	Se mostrará el elemento con la nueva imagen elegida. Si se trata de un marcador, el botón de añadir imagen se deshabilitará al pulsar sobre el marcador.
	Resultado obtenido
	Se mostró el elemento con la nueva imagen elegida. **Error**. Se seleccionó un marcador y no se deshabilitó el botón de añadir imagen. **Solución**: se modificó el comportamiento del marcador. **Corregido**. Se seleccionó un marcador y se deshabilitó el botón de añadir imagen.
Prueba	**Resultado Esperado**
Con un elemento seleccionado, se selecciona una imagen de menor tamaño que la actual.	El elemento cambiará su imagen y su tamaño por los de la nueva imagen. Si se trata de un marcador, el botón de añadir imagen se deshabilitará al pulsar sobre el marcador.
	Resultado obtenido
	Se mostró el elemento con la nueva imagen elegida. Se seleccionó un marcador y se deshabilitó el botón de añadir imagen.

Prueba	**Resultado Esperado**
Con un elemento seleccionado, se selecciona una imagen de mayor tamaño que la actual.	El elemento cambiará su imagen y su tamaño por los de la nueva imagen. Si se trata de un marcador, el botón de añadir imagen se deshabilitará al pulsar sobre el marcador.
	Resultado obtenido
	Se mostró el elemento con la nueva imagen elegida. Se seleccionó un marcador y se deshabilitó el botón de añadir imagen.
Prueba	**Resultado Esperado**
Con un elemento seleccionado, se intenta elegir un sonido para que se reproduzca cuando se pulse el botón en el videojuego con formato .mp4.	El sistema no permite seleccionar archivos de sonido con formato distinto a .mp3. El explorador de carpetas para escoger lo tiene un filtro, así que solamente mostrará archivos de sonido con esta extensión.
	Resultado obtenido
	El sistema no permitió seleccionar archivos de sonido con formato distinto a .mp3.
Prueba	**Resultado Esperado**
Con un elemento seleccionado, se selecciona un sonido para que se reproduzca cuando se pulse el botón en el videojuego en formato .mp3.	Se asigna ese sonido al elemento, para que se reproduzca cuando se pulse.
	Resultado obtenido
	Se asignó ese sonido al elemento.
Prueba	**Resultado Esperado**
Con un elemento seleccionado, se marca la opción de ocultar.	Ese elemento se mostrará con una transparencia muy alta, pero sin llegar a desaparecer por si se quiere des ocultar. Este botón no aparecerá en el videojuego.
	Resultado obtenido
	Ese elemento se mostró con una transparencia muy alta, pero sin llegar a desaparecer por si se quiere des ocultar. Este botón no apareció en el videojuego.
Prueba	**Resultado Esperado**
Con un elemento seleccionado y que se ha ocultado previamente, se desmarca la opción de ocultar.	Se vuelve a mostrar el elemento como estaba originalmente. El botón aparecerá en el juego.
	Resultado obtenido
	Se vuelve a mostrar el elemento como estaba originalmente. El botón apareció en el juego.
Prueba	**Resultado Esperado**
Con un elemento seleccionado, se desplaza este por la pantalla.	El elemento se moverá mientras se tenga pulsado el botón izquierdo del ratón. Cuando se deje de pulsar el botón dejará de moverse y quedará en esa posición. En los campos de texto de Ancho y Alto se actualizará el valor de sus coordenadas mientras se esté desplazando
	Resultado obtenido
	El elemento se movió mientras se tuvo pulsado el botón izquierdo del ratón. Cuando se dejó de pulsar el botón dejó de moverse y quedó en esa posición. En los campos de texto de Ancho y Alto se actualizó el valor de sus coordenadas mientras se estuvo desplazando

8.2.1.1.9 Subcaso de uso: Configurar pantalla derrota nivel

Caso de Uso 1: Crear nuevo videojuego. Subcaso: Configurar pantalla derrota nivel	
Prueba	**Resultado Esperado**
Se selecciona el fondo y se intenta modificar su imagen con una en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
	Resultado obtenido
	El sistema no permitió seleccionar imágenes con formato distinto a .png.
Prueba	**Resultado Esperado**
Se selecciona una imagen para el fondo en formato .png.	Se mostrará el fondo de la pantalla con la nueva imagen elegida.
	Resultado obtenido
	Se mostró el fondo de la pantalla con la nueva imagen elegida.
Prueba	**Resultado Esperado**
Con la imagen de fondo seleccionada se modifica su posición X e Y.	El fondo de pantalla se desplazará automáticamente a su nueva posición.
	Resultado obtenido
	El fondo de pantalla se desplazó automáticamente a su nueva posición.
Prueba	**Resultado Esperado**
Se selecciona uno de los botones.	El botón seleccionado aparecerá rodeado por un rectángulo de color verde.
	Resultado obtenido
	El botón seleccionado apareció rodeado por un rectángulo de color verde.
Prueba	**Resultado Esperado**
Con un botón seleccionado se modifica su ancho y su alto.	El botón modificará su tamaño automáticamente.
	Resultado obtenido
	El botón modificó su tamaño automáticamente.
Prueba	**Resultado Esperado**
Con un botón seleccionado se modifica su posición X e Y.	El botón modificará su posición en la pantalla automáticamente.
	Resultado obtenido
	El botón modificó su posición en la pantalla automáticamente.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se intenta modificar su imagen con una en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
	Resultado obtenido
	El sistema no permitió seleccionar imágenes con formato distinto a .png.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen	Se mostrará el botón con la nueva imagen elegida.
	Resultado obtenido

para el fondo en formato .png.	Se mostró el botón con la nueva imagen elegida.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen de menor tamaño que la actual.	El botón cambiará su imagen y su tamaño por los de la nueva imagen.
	Resultado obtenido
	El botón cambió su imagen y su tamaño por los de la nueva imagen.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen de mayor tamaño que la actual.	El botón cambiará su imagen y su tamaño por los de la nueva imagen.
	Resultado obtenido
	El botón cambió su imagen y su tamaño por los de la nueva imagen.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se intenta elegir un sonido para que se reproduzca cuando se pulse el botón en el videojuego con formato .mp4.	El sistema no permite seleccionar archivos de sonido con formato distinto a .mp3. El explorador de carpetas para escoger lo tiene un filtro, así que solamente mostrará archivos de sonido con esta extensión.
	Resultado obtenido
	El sistema no permitió seleccionar archivos de sonido con formato distinto a .mp3.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona un sonido para que se reproduzca cuando se pulse el botón en el videojuego en formato .mp3.	Se asigna ese sonido al botón, para que se reproduzca cuando se pulse.
	Resultado obtenido
	Se asignó ese sonido al botón.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se marca la opción de ocultar.	Ese botón se mostrará con una transparencia muy alta, pero sin llegar a desaparecer por si se quiere des ocultar. Este botón no aparecerá en el videojuego.
	Resultado obtenido
	Ese botón se mostró con una transparencia muy alta, pero sin llegar a desaparecer por si se quiere des ocultar. Este botón no apareció en el videojuego.
Prueba	**Resultado Esperado**
Con un botón seleccionado y que se ha ocultado previamente, se desmarca la opción de ocultar.	Se vuelve a mostrar el botón como estaba originalmente. El botón aparecerá en el juego.
	Resultado obtenido
	Se vuelve a mostrar el botón como estaba originalmente. El botón apareció en el juego.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se desplaza este por la pantalla.	El botón se moverá mientras se tenga pulsado el botón izquierdo del ratón. Cuando se deje de pulsar el botón dejará de moverse y quedará en esa posición. En los campos de texto de Ancho y Alto se actualizará el valor de sus coordenadas mientras se esté desplazando
	Resultado obtenido
	El botón se movió mientras se tuvo pulsado el botón

	izquierdo del ratón. Cuando se dejó de pulsar el botón dejó de moverse y quedó en esa posición. En los campos de texto de Ancho y Alto se actualizó el valor de sus coordenadas mientras se estuvo desplazando

8.2.1.1.10 Subcaso de uso: Configurar pantalla victoria nivel

Caso de Uso 1: Crear nuevo videojuego. Subcaso: Configurar pantalla victoria nivel	
Prueba	**Resultado Esperado**
Se selecciona el fondo y se intenta modificar su imagen con una en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
	Resultado obtenido
	El sistema no permitió seleccionar imágenes con formato distinto a .png.
Prueba	**Resultado Esperado**
Se selecciona una imagen para el fondo en formato .png.	Se mostrará el fondo de la pantalla con la nueva imagen elegida.
	Resultado obtenido
	Se mostró el fondo de la pantalla con la nueva imagen elegida.
Prueba	**Resultado Esperado**
Con la imagen de fondo seleccionada se modifica su posición X e Y.	El fondo de pantalla se desplazará automáticamente a su nueva posición.
	Resultado obtenido
	El fondo de pantalla se desplazó automáticamente a su nueva posición.
Prueba	**Resultado Esperado**
Se selecciona uno de los botones.	El botón seleccionado aparecerá rodeado por un rectángulo de color verde.
	Resultado obtenido
	El botón seleccionado apareció rodeado por un rectángulo de color verde.
Prueba	**Resultado Esperado**
Con un botón seleccionado se modifica su ancho y su alto.	El botón modificará su tamaño automáticamente.
	Resultado obtenido
	El botón modificó su tamaño automáticamente.
Prueba	**Resultado Esperado**
Con un botón seleccionado se modifica su posición X e Y.	El botón modificará su posición en la pantalla automáticamente.
	Resultado obtenido
	El botón modificó su posición en la pantalla automáticamente.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se intenta modificar su imagen con una en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.

	Resultado obtenido
	El sistema no permitió seleccionar imágenes con formato distinto a .png.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen para el fondo en formato .png.	Se mostrará el botón con la nueva imagen elegida.
	Resultado obtenido
	Se mostró el botón con la nueva imagen elegida.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen de menor tamaño que la actual.	El botón cambiará su imagen y su tamaño por los de la nueva imagen.
	Resultado obtenido
	El botón cambió su imagen y su tamaño por los de la nueva imagen.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen de mayor tamaño que la actual.	El botón cambiará su imagen y su tamaño por los de la nueva imagen.
	Resultado obtenido
	El botón cambió su imagen y su tamaño por los de la nueva imagen.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se intenta elegir un sonido para que se reproduzca cuando se pulse el botón en el videojuego con formato .mp4.	El sistema no permite seleccionar archivos de sonido con formato distinto a .mp3. El explorador de carpetas para escoger lo tiene un filtro, así que solamente mostrará archivos de sonido con esta extensión.
	Resultado obtenido
	El sistema no permitió seleccionar archivos de sonido con formato distinto a .mp3.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona un sonido para que se reproduzca cuando se pulse el botón en el videojuego en formato .mp3.	Se asigna ese sonido al botón, para que se reproduzca cuando se pulse.
	Resultado obtenido
	Se asignó ese sonido al botón.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se marca la opción de ocultar.	Ese botón se mostrará con una transparencia muy alta, pero sin llegar a desaparecer por si se quiere des ocultar. Este botón no aparecerá en el videojuego.
	Resultado obtenido
	Ese botón se mostró con una transparencia muy alta, pero sin llegar a desaparecer por si se quiere des ocultar. Este botón no apareció en el videojuego.
Prueba	**Resultado Esperado**
Con un botón seleccionado y que se ha ocultado previamente, se desmarca la opción de ocultar.	Se vuelve a mostrar el botón como estaba originalmente. El botón aparecerá en el juego.
	Resultado obtenido
	Se volvió a mostrar el botón como estaba originalmente. El botón apareció en el juego.
Prueba	**Resultado Esperado**
Con un botón seleccionado,	El botón se moverá mientras se tenga pulsado el botón

se desplaza este por la pantalla.	izquierdo del ratón. Cuando se deje de pulsar el botón dejará de moverse y quedará en esa posición. En los campos de texto de Ancho y Alto se actualizará el valor de sus coordenadas mientras se esté desplazando
	Resultado obtenido
	El botón se movió mientras se tuvo pulsado el botón izquierdo del ratón. Cuando se dejó de pulsar el botón dejó de moverse y quedó en esa posición. En los campos de texto de Ancho y Alto se actualizó el valor de sus coordenadas mientras se estuvo desplazando

8.2.1.1.11 Subcaso de uso: Configurar pantalla fin juego

Caso de Uso 1: Crear nuevo videojuego. Subcaso: Configurar pantalla fin de juego	
Prueba	**Resultado Esperado**
Se selecciona el fondo y se intenta modificar su imagen con una en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
	Resultado obtenido
	El sistema no permitió seleccionar imágenes con formato distinto a .png.
Prueba	**Resultado Esperado**
Se selecciona una imagen para el fondo en formato .png.	Se mostrará el fondo de la pantalla con la nueva imagen elegida.
	Resultado obtenido
	Se mostró el fondo de la pantalla con la nueva imagen elegida.
Prueba	**Resultado Esperado**
Con la imagen de fondo seleccionada se modifica su posición X e Y.	El fondo de pantalla se desplazará automáticamente a su nueva posición.
	Resultado obtenido
	El fondo de pantalla se desplazó automáticamente a su nueva posición.
Prueba	**Resultado Esperado**
Se selecciona uno de los botones.	El botón seleccionado aparecerá rodeado por un rectángulo de color verde.
	Resultado obtenido
	El botón seleccionado apareció rodeado por un rectángulo de color verde.
Prueba	**Resultado Esperado**
Con un botón seleccionado se modifica su ancho y su alto.	El botón modificará su tamaño automáticamente.
	Resultado obtenido
	El botón modificó su tamaño automáticamente.
Prueba	**Resultado Esperado**
Con un botón seleccionado se modifica su posición X e Y.	El botón modificará su posición en la pantalla automáticamente.
	Resultado obtenido
	El botón modificó su posición en la pantalla

	automáticamente.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se intenta modificar su imagen con una en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
	Resultado obtenido
	El sistema no permite seleccionar imágenes con formato distinto a .png.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen para el fondo en formato .png.	Se mostrará el botón con la nueva imagen elegida.
	Resultado obtenido
	Se mostró el botón con la nueva imagen elegida.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen de menor tamaño que la actual.	El botón cambiará su imagen y su tamaño por los de la nueva imagen.
	Resultado obtenido
	El botón cambió su imagen y su tamaño por los de la nueva imagen.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona una imagen de mayor tamaño que la actual.	El botón cambiará su imagen y su tamaño por los de la nueva imagen.
	Resultado obtenido
	El botón cambió su imagen y su tamaño por los de la nueva imagen.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se intenta elegir un sonido para que se reproduzca cuando se pulse el botón en el videojuego con formato .mp4.	El sistema no permite seleccionar archivos de sonido con formato distinto a .mp3. El explorador de carpetas para escoger lo tiene un filtro, así que solamente mostrará archivos de sonido con esta extensión.
	Resultado obtenido
	El sistema no permite seleccionar archivos de sonido con formato distinto a .mp3.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se selecciona un sonido para que se reproduzca cuando se pulse el botón en el videojuego en formato .mp3.	Se asigna ese sonido al botón, para que se reproduzca cuando se pulse.
	Resultado obtenido
	Se asignó ese sonido al botón.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se marca la opción de ocultar.	Ese botón se mostrará con una transparencia muy alta, pero sin llegar a desaparecer por si se quiere des ocultar. Este botón no aparecerá en el videojuego.
	Resultado obtenido
	Ese botón se mostró con una transparencia muy alta, pero sin llegar a desaparecer por si se quiere des ocultar. Este botón no apareció en el videojuego.
Prueba	**Resultado Esperado**
Con un botón seleccionado y	Se vuelve a mostrar el botón como estaba originalmente. El

que se ha ocultado previamente, se desmarca la opción de ocultar.	botón aparecerá en el juego.
	Resultado obtenido
	Se vuelve a mostrar el botón como estaba originalmente. El botón apareció en el juego.
Prueba	**Resultado Esperado**
Con un botón seleccionado, se desplaza este por la pantalla.	El botón se moverá mientras se tenga pulsado el botón izquierdo del ratón. Cuando se deje de pulsar el botón dejará de moverse y quedará en esa posición. En los campos de texto de Ancho y Alto se actualizará el valor de sus coordenadas mientras se esté desplazando
	Resultado obtenido
	El botón se movió mientras se tuvo pulsado el botón izquierdo del ratón. Cuando se dejó de pulsar el botón dejó de moverse y quedó en esa posición. En los campos de texto de Ancho y Alto se actualizó el valor de sus coordenadas mientras se estuvo desplazando

8.2.1.1.12 Subcaso de uso: Crear personaje principal

Caso de Uso 1: Crear nuevo videojuego. Subcaso: Crear personaje principal	
Prueba	**Resultado Esperado**
Se pulsa sobre el botón "Añadir a biblioteca" sin haber creado ningún personaje.	El botón estará deshabilitado y no se podrá añadir.
	Resultado obtenido
	El botón está deshabilitado y no se pudo añadir.
Prueba	**Resultado Esperado**
Se intenta añadir una imagen a cada animación en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
	Resultado obtenido
	El sistema no permite seleccionar imágenes con formato distinto a .png.
Prueba	**Resultado Esperado**
Se selecciona una imagen para cada animación para el personaje en formato .png.	Se mostrará en pantalla la imagen elegida y en la vista previa se mostrará la ejecución de la animación
	Resultado obtenido
	Se mostró en pantalla la imagen elegida y en la vista previa se mostró la ejecución de la animación
Prueba	**Resultado Esperado**
Se modifica el número de frames por uno mayor al número de animaciones que tiene la imagen.	En la vista previa se mostrará la ejecución de la animación y se verá que no es correcta porque no se ha introducido el número de frames adecuado.
	Resultado obtenido
	En la vista previa se mostró la ejecución de la animación y se vio que no es correcta porque no se ha introducido el número de frames adecuado.
Prueba	**Resultado Esperado**
Se modifica el número de	En la vista previa se mostrará la ejecución de la animación y

frames por uno menos al número de animaciones que tiene la imagen.	se verá que no es correcta porque no se ha introducido el número de frames adecuado.
	Resultado obtenido
	En la vista previa se mostró la ejecución de la animación y se vio que no es correcta porque no se ha introducido el número de frames adecuado.
Prueba	**Resultado Esperado**
Se modifica el número de frames por el número de animaciones que tiene la imagen.	En la vista previa se mostrará la ejecución de la animación y se verá que es correcta.
	Resultado obtenido
	En la vista previa se mostró la ejecución de la animación y se vio que es correcta.
Prueba	**Resultado Esperado**
Se modificarán las propiedades en las que se tenga que introducir un valor manualmente, introduciendo letras en lugar de números.	El sistema avisará de que se han introducido valores erróneos.
	Resultado obtenido
	El sistema avisó de que se han introducido valores erróneos.
Prueba	**Resultado Esperado**
Se modifica la velocidad del personaje mediante el desplegable. Se prueba con valores negativos, positivos y el 0.	Se ilumina la flecha correspondiente a la velocidad elegida. Al modificar la velocidad se ilumina automáticamente la flecha que corresponde.
	Resultado obtenido
	Se ilumina la flecha correspondiente a la velocidad elegida. Al modificar la velocidad se ilumina automáticamente la flecha que corresponde.
Prueba	**Resultado Esperado**
Se intenta asignar un sonido al personaje que no está en formato .mp3.	El sistema no permite seleccionar archivos de sonido con formato distinto a .mp3. El explorador de carpetas para escoger lo tiene un filtro, así que solamente mostrará archivos de sonido con esta extensión.
	Resultado obtenido
	El sistema no permite seleccionar archivos de sonido con formato distinto a .mp3.
Prueba	**Resultado Esperado**
Se selecciona un archivo de sonido .mp3 para cada sonido del personaje.	Los sonidos se cargan y almacenan correctamente.
	Resultado obtenido
	Los sonidos se cargan y almacenan correctamente.
Prueba	**Resultado Esperado**
Se pulsa sobre el botón "Borrar".	Si aún no se ha añadido a la biblioteca el personaje principal creado, no se podrá borrar. Si ya se ha añadido se borrará correctamente.
	Resultado obtenido
	No se ha añadió a la biblioteca el personaje creado, no se pudo borrar. Ya se ha añadió se borró correctamente.
Prueba	**Resultado Esperado**
Se pulsa sobre el botón "Añadir a biblioteca" con un personaje creado.	El personaje se almacenará correctamente y se añadirá a la lista que hay en la parte superior de la pantalla.

	Resultado obtenido
	El personaje se almacenó correctamente y se añadió a la lista que hay en la parte superior de la pantalla.
Prueba	**Resultado Esperado**
Se crea otro personaje principal y se pulsa sobre "Añadir a biblioteca".	Se almacenará y se añadirá a lista que se encuentra situada en la parte superior de la pantalla.
	Resultado obtenido
	Se almacenó y se añadió a lista que se encuentra situada en la parte superior de la pantalla.
Prueba	**Resultado Esperado**
Se hace clic sobre uno de los personajes creados.	Se cargarán en pantalla sus animaciones y el valor de sus propiedades.
	Resultado obtenido
	Se cargaron en pantalla sus animaciones y el valor de sus propiedades.
Prueba	**Resultado Esperado**
Con un personaje seleccionado, se modifica alguna de sus propiedades y alguna de sus animaciones, y se hace clic en "Actualizar".	Se guardarán los cambios realizados y se mostrarán en pantalla.
	Resultado obtenido
	Se guardaron los cambios realizados y se mostraron en pantalla.

8.2.1.1.13 Subcaso de uso: Crear enemigos

Caso de Uso 1: Crear nuevo videojuego. Subcaso: Crear enemigos	
Prueba	**Resultado Esperado**
Se pulsa sobre el botón "Añadir a biblioteca" sin haber creado ningún enemigo.	El botón estará deshabilitado y no se podrá añadir.
	Resultado obtenido
	El botón está deshabilitado y no se pudo añadir.
Prueba	**Resultado Esperado**
Se intenta añadir una imagen a cada animación en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
	Resultado obtenido
	El sistema no permite seleccionar imágenes con formato distinto a .png
Prueba	**Resultado Esperado**
Se selecciona una imagen para cada animación del enemigo en formato .png.	Se mostrará en pantalla la imagen elegida y en la vista previa se mostrará la ejecución de la animación
	Resultado obtenido
	Se mostró en pantalla la imagen elegida y en la vista previa se mostró la ejecución de la animación
Prueba	**Resultado Esperado**
Se modifica el número de frames por uno mayor al número de animaciones que	En la vista previa se mostrará la ejecución de la animación y se verá que no es correcta porque no se ha introducido el número de frames adecuado.

tiene la imagen.	**Resultado obtenido**
	En la vista previa se mostró la ejecución de la animación y se vio que no es correcta porque no se ha introducido el número de frames adecuado.
Prueba	**Resultado Esperado**
Se modifica el número de frames por uno menos al número de animaciones que tiene la imagen.	En la vista previa se mostrará la ejecución de la animación y se verá que no es correcta porque no se ha introducido el número de frames adecuado.
	Resultado obtenido
	En la vista previa se mostró la ejecución de la animación y se vio que no es correcta porque no se ha introducido el número de frames adecuado.
Prueba	**Resultado Esperado**
Se modifica el número de frames por el número de animaciones que tiene la imagen.	En la vista previa se mostrará la ejecución de la animación y se verá que es correcta.
	Resultado obtenido
	En la vista previa se mostró la ejecución de la animación y se vio que es correcta.
Prueba	**Resultado Esperado**
Se modificarán las propiedades en las que se tenga que introducir un valor manualmente, introduciendo letras en lugar de números.	El sistema avisará de que se han introducido valores erróneos.
	Resultado obtenido
	El sistema avisó de que se han introducido valores erróneos.
Prueba	**Resultado Esperado**
Se modifica la velocidad del enemigo mediante el desplegable. Se prueba con valores negativos, positivos y el 0.	Se ilumina la flecha correspondiente a la velocidad elegida. Al modificar la velocidad se ilumina automáticamente la flecha que corresponde.
	Resultado obtenido
	Se ilumina la flecha correspondiente a la velocidad elegida. Al modificar la velocidad se ilumina automáticamente la flecha que corresponde.
Prueba	**Resultado Esperado**
Se intenta asignar un sonido al enemigo que no está en formato .mp3.	El sistema no permite seleccionar archivos de sonido con formato distinto a .mp3. El explorador de carpetas para escoger lo tiene un filtro, así que solamente mostrará archivos de sonido con esta extensión.
	Resultado obtenido
	El sistema no permite seleccionar archivos de sonido con formato distinto a .mp3.
Prueba	**Resultado Esperado**
Se selecciona un archivo de sonido .mp3 para cada sonido del enemigo.	Los sonidos se cargan y almacenan correctamente.
	Resultado obtenido
	Los sonidos se cargan y almacenan correctamente.
Prueba	**Resultado Esperado**
Se pulsa sobre el botón "Borrar".	Si aún no se ha añadido a la biblioteca el enemigo creado, no se podrá borrar. Si ya se ha añadido se borrará correctamente.
	Resultado obtenido
	No se ha añadió a la biblioteca el enemigo creado, no se pudo

	borrar. Ya se ha añadió se borró correctamente.
Prueba	**Resultado Esperado**
Se pulsa sobre el botón "Añadir a biblioteca" con un enemigo creado.	El enemigo se almacenará correctamente y se añadirá a la lista que hay en la parte superior de la pantalla.
	Resultado obtenido
	El enemigo se almacenó correctamente y se añadió a la lista que hay en la parte superior de la pantalla.
Prueba	**Resultado Esperado**
Se crea otro enemigo principal y se pulsa sobre "Añadir a biblioteca".	Se almacenará y se añadirá a lista que se encuentra situada en la parte superior de la pantalla.
	Resultado obtenido
	Se almacenó y se añadió a lista que se encuentra situada en la parte superior de la pantalla.
Prueba	**Resultado Esperado**
Se hace clic sobre uno de los enemigos creados.	Se cargarán en pantalla sus animaciones y el valor de sus propiedades.
	Resultado obtenido
	Se cargaron en pantalla sus animaciones y el valor de sus propiedades.
Prueba	**Resultado Esperado**
Con un enemigo seleccionado, se modifica alguna de sus propiedades y alguna de sus animaciones, y se hace clic en "Actualizar".	Se guardarán los cambios realizados y se mostrarán en pantalla.
	Resultado obtenido
	Se guardaron los cambios realizados y se mostraron en pantalla.

8.2.1.1.14 Subcaso de uso: Crear meta

Caso de Uso 1: Crear nuevo videojuego. Subcaso: Crear meta	
Prueba	**Resultado Esperado**
Se pulsa sobre el botón "Añadir a biblioteca" sin haber creado ninguna meta.	El botón estará deshabilitado y no se podrá añadir.
	Resultado obtenido
	El botón está deshabilitado y no se pudo añadir.
Prueba	**Resultado Esperado**
Se intenta añadir una imagen a la animación en formato .jpg.	El sistema no permite seleccionar imágenes con formato distinto a .png. El explorador de carpetas para escoger la imagen tiene un filtro, así que solamente mostrará imágenes con esta extensión.
	Resultado obtenido
	El sistema no permite seleccionar imágenes con formato distinto a .png
Prueba	**Resultado Esperado**
Se selecciona una imagen para la animación de la meta en formato .png.	Se mostrará en pantalla la imagen elegida y en la vista previa se mostrará la ejecución de la animación
	Resultado obtenido
	Se mostró en pantalla la imagen elegida y en la vista previa se

	mostró la ejecución de la animación
Prueba	**Resultado Esperado**
Se modificarán las propiedades en las que se tenga que introducir un valor manualmente, introduciendo letras en lugar de números.	El sistema avisará de que se han introducido valores erróneos.
	Resultado obtenido
	El sistema avisó de que se han introducido valores erróneos.
Prueba	**Resultado Esperado**
Se intenta asignar un sonido a la meta que no está en formato .mp3.	El sistema no permite seleccionar archivos de sonido con formato distinto a .mp3. El explorador de carpetas para escoger lo tiene un filtro, así que solamente mostrará archivos de sonido con esta extensión.
	Resultado obtenido
	El sistema no permitió seleccionar archivos de sonido con formato distinto a .mp3.
Prueba	**Resultado Esperado**
Se selecciona un archivo de sonido .mp3 para el sonido de la meta.	Los sonidos se cargan y almacenan correctamente.
	Resultado obtenido
	Los sonidos se cargan y almacenan correctamente.
Prueba	**Resultado Esperado**
Se pulsa sobre el botón "Borrar".	Si aún no se ha añadido a la biblioteca la meta creada, no se podrá borrar. Si ya se ha añadido se borrará correctamente.
	Resultado obtenido
	No se ha añadido a la biblioteca la meta creada, no se pudo borrar. Ya se ha añadido se borró correctamente.
Prueba	**Resultado Esperado**
Se pulsa sobre el botón "Añadir a biblioteca" con una meta creada.	La meta se almacenará correctamente y se añadirá a la lista que hay en la parte superior de la pantalla.
	Resultado obtenido
	La meta se almacenó correctamente y se añadió a la lista que hay en la parte superior de la pantalla.
Prueba	**Resultado Esperado**
Se crea otra meta y se pulsa sobre "Añadir a biblioteca".	Se almacenará y se añadirá a lista que se encuentra situada en la parte superior de la pantalla.
	Resultado obtenido
	Se almacenó y se añadió a lista que se encuentra situada en la parte superior de la pantalla.
Prueba	**Resultado Esperado**
Se hace clic sobre una de las metas creadas.	Se cargarán en pantalla sus animaciones y el valor de sus propiedades.
	Resultado obtenido
	Se cargaron en pantalla sus animaciones y el valor de sus propiedades.
Prueba	**Resultado Esperado**
Con una meta seleccionada, se modifica alguna de sus	Se guardarán los cambios realizados y se mostrarán en pantalla.

propiedades y la imagen de la animación, y se hace clic en "Actualizar".	**Resultado obtenido**
	Se guardaron los cambios realizados y se mostraron en pantalla.

8.2.1.1.15 Subcaso de uso: Crear nivel

Caso de Uso 1: Crear nuevo videojuego. Subcaso: Crear nivel	
Prueba	**Resultado Esperado**
Se crea un escenario vacío.	No se puede crear el escenario y el sistema avisa que es necesario añadir un personaje principal y una meta.
	Resultado obtenido
	No se pudo crear el escenario y el sistema avisó que es necesario añadir un personaje principal y una meta.
Prueba	**Resultado Esperado**
Se añaden elementos a un nivel.	Los elementos quedan añadidos en el escenario.
	Resultado obtenido
	Los elementos quedan añadidos en el escenario.
Prueba	**Resultado Esperado**
Se borra un elemento del nivel.	El elemento se elimina del escenario.
	Resultado obtenido
	El elemento se elimina del escenario.
Prueba	**Resultado Esperado**
Se configuran las propiedades del nivel.	Se asignan los valores introducidos al nivel.
	Resultado obtenido
	Se asignan los valores introducidos al nivel.
Prueba	**Resultado Esperado**
Se añade una imagen de descripción y una miniatura del nivel.	Se añaden esas imágenes al nivel.
	Resultado obtenido
	Se añaden esas imágenes al nivel.
Prueba	**Resultado Esperado**
Se modifican los carriles del nivel.	Se modifica la posición Y de los carriles.
	Resultado obtenido
	Se modifica la posición Y de los carriles.
Prueba	**Resultado Esperado**
Se añade un fondo al nivel.	Se muestra la imagen en el escenario.
	Resultado obtenido
	Se muestra la imagen en el escenario.
Prueba	**Resultado Esperado**
Se añade un fondo multicapa al nivel.	Se muestran las imágenes en el escenario.
	Resultado obtenido
	Se muestran las imágenes en el escenario.
Prueba	**Resultado Esperado**
Se prueba a aumentar el escenario por la derecha y por la izquierda.	El escenario crece por la derecha e izquierda respectivamente.
	Resultado obtenido
	El escenario crece por la derecha e izquierda respectivamente.
Prueba	**Resultado Esperado**

Se guardan los cambios realizados al nivel pulsando sobre "Guardar nivel".	Se guardan los cambios en el nivel.
	Resultado obtenido
	Se guardan los cambios en el nivel.
Prueba	**Resultado Esperado**
Se añade un nuevo nivel.	Si no se han guardado los cambios introducidos en el nivel anterior, el sistema mostrará un aviso informando de que es necesario guardar los cambios. Si se han guardado, se creará el nuevo el nivel con un número siguiente al anterior y se limpiará el escenario.
	Resultado obtenido
	No se han guardado los cambios introducidos en el nivel anterior y el sistema mostró un aviso informando de que es necesario guardar los cambios. Se han guardado y se creó el nuevo el nivel con un número siguiente al anterior y se limpió el escenario.
Prueba	**Resultado Esperado**
Se hace clic sobre alguno de los niveles ya creados.	Se cargará el escenario y las propiedades de ese nivel.
	Resultado obtenido
	Se cargó el escenario y las propiedades de ese nivel.

8.2.1.2 Caso de uso: Guardar estado videojuego

Caso de Uso 2: Guardar estado videojuego	
Prueba	**Resultado Esperado**
Se navega entre pantallas modificando alguna de sus propiedades o bien dejando los valores por defecto.	En la carpeta donde se está almacenando el juego se va creando un archivo .dat por cada pantalla que se configura y se pulsa sobre "Siguiente". Si ya existe se actualiza.
	Resultado obtenido
	Se guardó un juego nuevo se crearon los archivos .dat de las pantallas configuradas. Se guardó un juego cargado y actualizaron los ficheros.
Prueba	**Resultado Esperado**
Se pulsa sobre la opción "Guardar" del menú.	En la carpeta donde se está almacenando el juego se va creando un archivo .dat por cada pantalla en la que se pulsó sobre "Guardar". Si ya existe se actualiza.
	Resultado obtenido
	Se creó el archivo de cada pantalla que se guardó.
Prueba	**Resultado Esperado**
Se recorren todas las pantallas de la interfaz.	Deberá existir un archivo .dat por cada pantalla de la interfaz.
	Resultado obtenido
	Había un archivo por cada una de las pantallas del interfaz.

8.2.1.3 Caso de uso: Cargar estado videojuego

Caso de Uso 3: Cargar estado videojuego	
Prueba	**Resultado Esperado**
En la pantalla inicial, se elige un directorio que contiene un juego ya creado con la interfaz.	Se cargará el videojuego correctamente. Se cargará el estado de todas sus pantallas.
	Resultado obtenido
	Se cargó correctamente.
Prueba	**Resultado Esperado**
En la interfaz, se pulsa "Anterior" después de haber pulsado "Siguiente" en cada pantalla.	Se muestra el último estado almacenado de cada pantalla.
	Resultado obtenido
	Se mostró el último estado almacenado de cada pantalla.
Prueba	**Resultado Esperado**
Se carga un videojuego con errores, como por ejemplo, faltan recursos.	El sistema mostrará un error avisando del fallo producido.
	Resultado obtenido
	Error. El sistema no mostró ningún aviso y dejó de funcionar. **Solución**: se incorporó código para controlar este caso. **Corregido**. El sistema mostró un aviso.

8.2.1.4 Caso de uso: Exportar videojuego

Caso de Uso 4: Exportar videojuego	
Prueba	**Resultado Esperado**
Se finaliza el proceso de crear videojuego y se llega a la pantalla se selección de ruta de despliegue.	Se muestra la pantalla de selección de ruta de despliegue.
	Resultado obtenido
	Se muestra la pantalla de selección de ruta de despliegue.
Prueba	**Resultado Esperado**
Se escoge un directorio donde almacenar el proyecto del videojuego.	Se abre un cuadro de diálogo y se escoge un directorio.
	Resultado obtenido
	Se abrió un cuadro de diálogo y se escogió un directorio.
Prueba	**Resultado Esperado**
Se pulsa sobre el botón "Generar videojuego".	Se mostrará un mensaje de que se está generando el videojuego y cuando termine el proceso, el sistema avisará de que se generó correctamente.
	Resultado obtenido
	Se mostró un mensaje de que se está generando el videojuego y cuando terminó el proceso, el sistema avisó de que se generó correctamente.
Prueba	**Resultado Esperado**
Se va al directorio donde se indicó que se iba a almacenar el proyecto.	Existe una carpeta llamada "Android_touch_with_character" que contiene el proyecto generado.
	Resultado obtenido
	Existe una carpeta llamada "Android_touch_with_character" que contiene el proyecto generado.
Prueba	**Resultado Esperado**
En ese directorio se abre la	Cada una de ellas contiene los ficheros de imágenes, de

carpeta "drawable","sounds" y "levels".	sonidos y de los niveles respectivamente.
	Resultado obtenido
	Cada una de ellas contiene los ficheros de imágenes, de sonidos y de los niveles respectivamente.
Prueba	**Resultado Esperado**
Se importa el proyecto generado al entorno de desarrollo Eclipse.	Al importarlo no da errores de ningún tipo.
	Resultado obtenido
	Error. Dio errores, había etiquetas sin sustituir por sus valores. **Solución**: Se corrigió el nombre de la etiqueta en la plantilla. **Corregido**. Al importarlo no dio errores de ningún tipo.
Prueba	**Resultado Esperado**
Se compila el proyecto.	Se compila correctamente y se obtiene la apk lista para instalar.
	Resultado obtenido
	Se compiló correctamente y se obtuvo la apk lista para instalar.

8.2.1.5 Caso de uso: Crear/Editar instancia DSL

Caso de Uso 5: Crear/Editar instancia DSL	
Prueba	**Resultado Esperado**
Crear archivo.	Se crea correctamente.
	Resultado obtenido
	Se crea correctamente.
Prueba	**Resultado Esperado**
Modificar archivo.	Se modifica correctamente.
	Resultado obtenido
	Se modifica correctamente.
Prueba	**Resultado Esperado**
Se crear con fallos de formato.	Al abrirlo con un navegador se mostrarán los errores.
	Resultado obtenido
	Al abrirlo con un navegador se mostraron los errores.
Prueba	**Resultado Esperado**
Se guarda con extensión .txt.	No es correcto, deberá ser .xml.
	Resultado obtenido
	El analizador no encuentra el fichero porque no tiene extensión .xml
Prueba	**Resultado Esperado**
Se guarda con extensión .xml	Es correcto. Queda almacenado en disco.
	Resultado obtenido
	Es correcto. Queda almacenado en disco.

8.2.1.6 Caso de uso: Crear/Editar Plantilla

Caso de Uso 6: Crear/Editar Plantilla	
Prueba	**Resultado Esperado**
Se crea una plantilla nueva.	La plantilla creada sigue la estructura de carpetas correcta.
	Resultado obtenido
	La plantilla creada sigue la estructura de carpetas correcta.
Prueba	**Resultado Esperado**
Se sustituyen las partes variables con datos correctos.	La plantilla se convertirá en un proyecto, funcionará correctamente. No dará errores ni de compilación ni de ejecución.
	Resultado obtenido
	La plantilla se convirtió en un proyecto, funcionó correctamente. No dio errores ni de compilación ni de ejecución.
Prueba	**Resultado Esperado**
Se sustituyen las partes variables con datos incorrectos y se borrarán algunos recursos.	El proyecto generado será incorrecto no compilará. Se avisará de los fallos que contiene.
	Resultado obtenido
	El proyecto generado es incorrecto no compiló. Se avisó de los fallos que contiene.
Prueba	**Resultado Esperado**
Se modifica una plantilla ya existente.	La plantilla modificada sigue la estructura de carpeta correcta.
	Resultado obtenido
	La plantilla modificada sigue la estructura de carpeta correcta.
Prueba	**Resultado Esperado**
Se sustituyen las partes variables con datos correctos.	La plantilla se convertirá en un proyecto, funcionará correctamente. No dará errores ni de compilación ni de ejecución.
	Resultado obtenido
	La plantilla se convirtió en un proyecto, funcionó correctamente. No dio errores ni de compilación ni de ejecución.
Prueba	**Resultado Esperado**
Se sustituyen las partes variables con datos incorrectos y se borrarán algunos recursos.	El proyecto generado será incorrecto no compilará. Se avisará de los fallos que contiene.
	Resultado obtenido
	El proyecto generado es incorrecto no compila. Se avisó de los fallos que contiene.
Prueba	**Resultado Esperado**
Se compila el proyecto creado o modificado, que se obtiene después de sustituir todas las partes variables por valores correctos, y se ejecuta.	Se mostrará un juego que funciona correctamente.
	Resultado obtenido
	Se mostró un juego que funciona correctamente.

8.2.1.7 Caso de uso: Jugar videojuego

Caso de Uso 7: Jugar videojuego	
Prueba	**Resultado Esperado**
Se abre la aplicación del videojuego en un dispositivo móvil.	El juego funciona correctamente y se puede jugar.
	Resultado obtenido
	El juego funciona correctamente y se puede jugar.
Prueba	**Resultado Esperado**
Se abre la aplicación del videojuego en el emulador del Eclipse.	El juego funciona correctamente y se puede jugar.
	Resultado obtenido
	El juego funciona correctamente y se puede jugar.
Prueba	**Resultado Esperado**
Se prueba a jugar todos los niveles.	Todos funcionan correctamente.
	Resultado obtenido
	Todos funcionan correctamente.
Prueba	**Resultado Esperado**
Se pierde un nivel.	Se pierde y se muestra la pantalla de derrota.
	Resultado obtenido
	Se pierde y se muestra la pantalla de derrota.
Prueba	**Resultado Esperado**
Se gana un nivel.	Se gana y se muestra la pantalla de victoria.
	Resultado obtenido
	Se gana y se muestra la pantalla de victoria.
Prueba	**Resultado Esperado**
Se terminan todos los niveles con éxito.	Se termina el videojuego y se muestra la pantalla de fin.
	Resultado obtenido
	Se termina el videojuego y se muestra la pantalla de fin.

8.3 Pruebas de Usabilidad

Las pruebas de usabilidad para este proyecto tienen un gran impacto en el grado de cumplimiento de los objetivos marcados al principio de este documento, ya que se pretende que la herramienta desarrollada sirva para que cualquier persona pueda crear sus propios videojuegos. Por lo tanto, los resultados obtenidos de estas pruebas serán tenidos en cuenta a la hora de realizar las conclusiones y para las futuras ampliaciones de este proyecto.

Para realizar estas pruebas se han utilizado 3 individuos de cada perfil, así que, en total se han realizado las pruebas con 12 usuarios.

8.3.1 Resultados de las Pruebas de Usabilidad

Para mostrar el resultado de las pruebas de usabilidad realizadas en el sistema de manera que sea fácil su estudio y la obtención de conclusiones a partir de ellos, se ha decidido mostrar de forma tabular. No obstante, en el CD se encuentran recogidos los resultados completos de estas pruebas.

8.3.1.1 Preguntas de Carácter General

A continuación se incluyen cuatros tablas que muestran los resultados obtenidos en las pruebas de usabilidad para las preguntas de carácter general.

8.3.1.1.1 Perfil usuario 1

Nº pregunta	Opción 1	Opción 2	Opción 3	Opción 4	Opción 5	Opción 6
1	100%	0%	0%	0%	---	---
2	100%	0%	0%	0%	---	---
3	66%	33%	0%	---	---	---
4	33%	33%	33%	0%	---	---
5	33%	66%	0%	---	---	---
6	100%	100%	50%	0%	0%	0%
7	100%	0%	---	---	---	---
8	0%	33%	66%	---	---	---
9	0%	0%	0%	100%	33%	0%
10	33%	66%	0%	0%	0%	---

Ilustración 8.1. Preguntas carácter general perfil usuario 1

8.3.1.1.2 Perfil usuario 2

Nº pregunta	Opción 1	Opción 2	Opción 3	Opción 4	Opción 5	Opción 6
1	66%	33%	0%	0%	---	---
2	100%	0%	0%	0%	---	---
3	100%	0%	0%	---	---	---
4	33%	66%	0%	0%	---	---
5	0%	66%	33%	---	---	---
6	33%	66%	0%	33%	0%	0%
7	100%	0%	---	---	---	---
8	0%	33%	66%	---	---	---
9	0%	0%	0%	100%	0%	0%
10	33%	66%	0%	0%	0%	---

Ilustración 8.2. Preguntas carácter general perfil usuario 2

8.3.1.1.3 Perfil usuario 3

Nº pregunta	Opción 1	Opción 2	Opción 3	Opción 4	Opción 5	Opción 6
1	0%	100%	0%	0%	---	---
2	33%	0%	66%	0%	---	---
3	0%	66%	33%	---	---	---
4	100%	0%	0%	0%	---	---
5	0%	0%	100%	---	---	---
6	33%	66%	0%	0%	0%	0%
7	100%	0%	---	---	---	---
8	0%	0%	100%	---	---	---
9	0%	0%	0%	100%	0%	0%
10	0%	100%	0%	0%	0%	---

Ilustración 8.3. Preguntas carácter general perfil usuario 3

8.3.1.1.4 Perfil usuario 4

Nº pregunta	Opción 1	Opción 2	Opción 3	Opción 4	Opción 5	Opción 6
1	0%	100%	0%	0%	---	---
2	33%	0%	66%	0%	---	---
3	0%	66%	33%	---	---	---
4	100%	0%	0%	0%	---	---
5	0%	0%	100%	---	---	---
6	33%	66%	0%	0%	0%	0%
7	100%	0%	---	---	---	---
8	0%	0%	100%	---	---	---
9	0%	0%	0%	100%	0%	0%
10	0%	100%	0%	0%	0%	---

Ilustración 8.4. Preguntas carácter general perfil usuario 4

8.3.1.2 Preguntas Cortas sobre la Aplicación y Observaciones

En este apartado se incluyen las tablas que hacen referencia a las preguntas realizadas a los usuarios una vez había terminado el proceso de crear un videojuego.

8.3.1.2.1 Perfil usuario 1

Nº pregunta	Si	No	A veces	Nº pregunta	Si	No	A veces
1	100%	0%	0%	29	100%	0%	0%
2	100%	0%	0%	30	100%	0%	0%
3	100%	0%	0%	31	100%	0%	0%
4	0%	100%	0%	32	100%	0%	0%
5	100%	0%	0%	33	100%	0%	0%
6	100%	0%	0%	34	0%	100%	0%
7	100%	0%	0%	35	100%	0%	0%
8	100%	0%	0%	36	100%	0%	0%
9	100%	0%	0%	37	100%	0%	0%
10	100%	0%	0%	38	100%	0%	0%
11	0%	100%	0%	39	100%	0%	0%
12	100%	0%	0%	40	100%	0%	0%
13	0%	100%	0%	41	100%	0%	0%
14	100%	0%	0%	42	100%	0%	0%
15	100%	0%	0%	43	100%	0%	0%
16	100%	0%	0%	44	100%	0%	0%
17	0%	100%	0%	45	100%	0%	0%
18	0%	100%	0%	46	100%	0%	0%
19	0%	100%	0%	47	100%	0%	0%
20	100%	0%	0%	48	100%	0%	0%
21	100%	0%	0%	49	100%	0%	0%

22	100%	0%	0%	**50**	100%	0%	0%
23	100%	0%	0%	**51**	100%	0%	0%
24	100%	0%	0%	**52**	100%	0%	0%
25	100%	0%	0%	**53**	100%	0%	0%
26	0%	100%	0%	**54**	100%	0%	0%
27	100%	0%	0%	**55**	100%	0%	0%
28	100%	0%	0%				0%

Ilustración 8.5. Preguntas cortas perfil usuario 1

8.3.1.2.2 Perfil usuario 2

Nº pregunta	Si	No	A veces	Nº pregunta	Si	No	A veces
1	100%	0%	0%	**29**	100%	0%	0%
2	100%	0%	0%	**30**	100%	0%	0%
3	100%	0%	0%	**31**	100%	0%	0%
4	0%	100%	0%	**32**	100%	0%	0%
5	100%	0%	0%	**33**	100%	0%	0%
6	100%	0%	0%	**34**	0%	100%	0%
7	100%	0%	0%	**35**	100%	0%	0%
8	100%	0%	0%	**36**	100%	0%	0%
9	100%	0%	0%	**37**	100%	0%	0%
10	100%	0%	0%	**38**	100%	0%	0%
11	0%	100%	0%	**39**	100%	0%	0%
12	100%	0%	0%	**40**	100%	0%	0%
13	0%	100%	0%	**41**	100%	0%	0%
14	100%	0%	0%	**42**	100%	0%	0%
15	100%	0%	0%	**43**	100%	0%	0%
16	100%	0%	0%	**44**	100%	0%	0%
17	0%	100%	0%	**45**	100%	0%	0%
18	0%	100%	0%	**46**	100%	0%	0%
19	0%	100%	0%	**47**	100%	0%	0%
20	100%	0%	0%	**48**	100%	0%	0%
21	100%	0%	0%	**49**	100%	0%	0%
22	100%	0%	0%	**50**	100%	0%	0%
23	100%	0%	0%	**51**	100%	0%	0%
24	100%	0%	0%	**52**	100%	0%	0%
25	100%	0%	0%	**53**	100%	0%	0%
26	0%	100%	0%	**54**	100%	0%	0%
27	100%	0%	0%	**55**	100%	0%	0%
28	100%	0%	0%				0%

Ilustración 8.6. Preguntas cortas perfil usuario 2

8.3.1.2.3 Perfil usuario 3

Nº pregunta	Si	No	A veces	Nº pregunta	Si	No	A veces
1	100%	0%	0%	29	100%	0%	0%
2	100%	0%	0%	30	100%	0%	0%
3	100%	0%	0%	31	100%	0%	0%
4	0%	100%	0%	32	100%	0%	0%
5	100%	0%	0%	33	100%	0%	0%
6	100%	0%	0%	34	0%	100%	0%
7	100%	0%	0%	35	100%	0%	0%
8	100%	0%	0%	36	100%	0%	0%
9	100%	0%	0%	37	100%	0%	0%
10	100%	0%	0%	38	100%	0%	0%
11	0%	100%	0%	39	100%	0%	0%
12	100%	0%	0%	40	100%	0%	0%
13	0%	100%	0%	41	100%	0%	0%
14	100%	0%	0%	42	100%	0%	0%
15	100%	0%	0%	43	100%	0%	0%
16	100%	0%	0%	44	100%	0%	0%
17	0%	100%	0%	45	100%	0%	0%
18	0%	100%	0%	46	100%	0%	0%
19	0%	100%	0%	47	100%	0%	0%
20	100%	0%	0%	48	100%	0%	0%
21	100%	0%	0%	49	100%	0%	0%
22	100%	0%	0%	50	100%	0%	0%
23	100%	0%	0%	51	100%	0%	0%
24	100%	0%	0%	52	100%	0%	0%
25	100%	0%	0%	53	100%	0%	0%
26	0%	100%	0%	54	100%	0%	0%
27	100%	0%	0%	55	100%	0%	0%
28	100%	0%	0%				0%

Ilustración 8.7. Preguntas cortas perfil usuario 3

8.3.1.2.4 Perfil usuario 4

Nº pregunta	Si	No	A veces	Nº pregunta	Si	No	A veces
1	100%	0%	0%	29	100%	0%	0%
2	100%	0%	0%	30	100%	0%	0%
3	100%	0%	0%	31	100%	0%	0%
4	0%	100%	0%	32	100%	0%	0%
5	100%	0%	0%	33	100%	0%	0%
6	100%	0%	0%	34	0%	100%	0%
7	100%	0%	0%	35	100%	0%	0%
8	100%	0%	0%	36	100%	0%	0%
9	100%	0%	0%	37	100%	0%	0%
10	100%	0%	0%	38	100%	0%	0%
11	0%	100%	0%	39	100%	0%	0%
12	66%	33%	0%	40	100%	0%	0%
13	0%	100%	0%	41	66%	33%	0%
14	100%	0%	0%	42	100%	0%	0%
15	100%	0%	0%	43	100%	0%	0%
16	100%	0%	0%	44	100%	0%	0%
17	0%	100%	0%	45	100%	0%	0%
18	0%	100%	0%	46	100%	0%	0%
19	66%	33%	0%	47	100%	0%	0%
20	100%	0%	0%	48	100%	0%	0%
21	100%	0%	0%	49	100%	0%	0%
22	100%	0%	0%	50	100%	0%	0%
23	100%	0%	0%	51	100%	0%	0%
24	100%	0%	0%	52	100%	0%	0%
25	100%	0%	0%	53	100%	0%	0%
26	0%	100%	0%	54	100%	0%	0%
27	100%	0%	0%	55	100%	0%	0%
28	100%	0%	0%				0%

Ilustración 8.8. Preguntas cortas perfil usuario 4

8.3.2 Análisis de los resultados obtenidos

Una vez realizadas las pruebas de usabilidad de la herramienta que se ha desarrollado como trabajo fin de máster, se han estudiado los resultados obtenidos en las mismas para analizar los puntos conflictivos del sistema e intentar solucionarles en futuras ampliaciones o mejoras.

En general, para ningún usuario de los diferentes perfiles supuso alguna dificultad crear un videojuego con la herramienta. Todos los usuarios obtuvieron el juego que esperaban y pudieron desplegarlo sin problemas en sus dispositivos móviles.

El proceso se crear los niveles fue quizás la parte más compleja del desarrollo para todos los usuarios, fue en la parte en la que invirtieron más tiempo, pero finalmente todos consiguieron crear los niveles que quisieron.

Al ofrecer al usuario los menús de los videojuegos ya configurados con imágenes por defecto, restó dificultad al proceso. Aunque alguno de ellos utilizó la biblioteca de imágenes que se le proporcionaba durante la prueba, y completaron el proceso de forma satisfactoria.

A todos los usuarios también se les proporcionó el manual de usuario, que fue consultado por el 50% de los sujetos, y este les ayudó a resolver las dudas que tenían. Fue consultado en su mayoría en el momento de definición de los niveles. Por tanto, para futuras ampliaciones se va a contemplar el reducir el número de acciones a realizar dentro del editor de niveles.

Capítulo 9. Manuales del Sistema

9.1 Manual de Instalación

9.1.1 Requisitos

Para poder realizar la instalación de la herramienta es necesario tener instalado ***.NET Framework v4.0*** o superior.

9.1.2 Pasos a seguir

En la carpeta de archivos ejecutables hay dos archivos:

- **.NetFramework4.0.exe:** que es el ejecutable para instalar la versión 4.0 del framework de .NET, necesario para poder ejecutar la herramienta. Solo es preciso ejecutarlo si se tiene una versión inferior del .NET Framework.
- **InstallSetup.msi:** es el ejecutable que realizará la instalación de la aplicación.

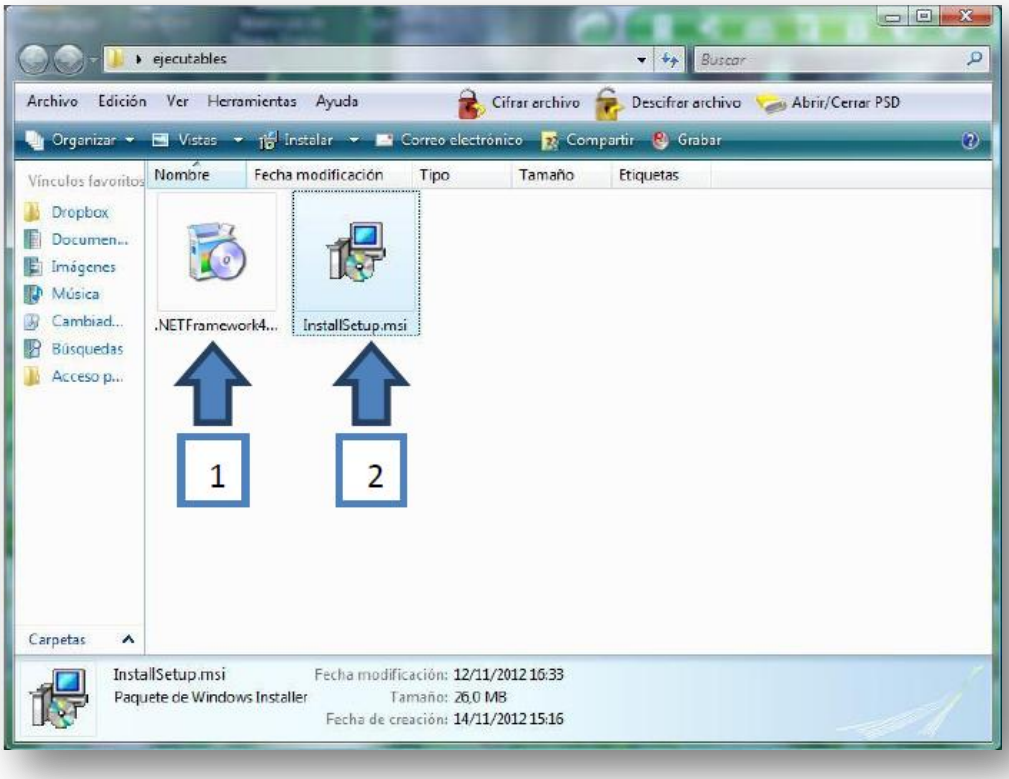


Ilustración 9.1. Archivos de instalación

PASO 1. Haciendo doble clic sobre ".NETFramework4.0.exe" se abrirá un asistente para la instalación o actualización del framework.

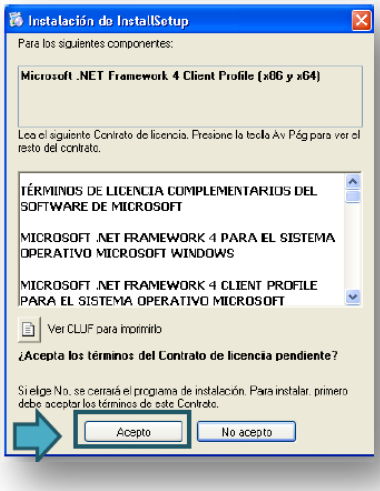


Ilustración 9.2. Instalación .NETFramework 4.0

Para comenzar la instalación se deberá pulsar sobre "*Acepto*". En caso de hacer clic en "*No Acepto*" se cancelará la instalación.

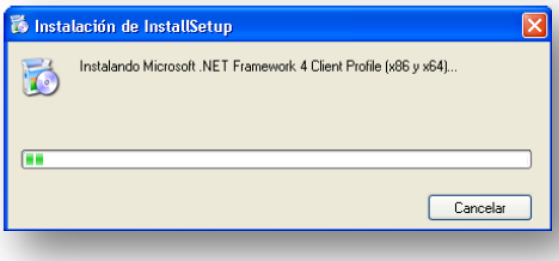


Ilustración 9.3. Instalando .NETFramework 4.0

Al finalizar será necesario reiniciar el sistema.

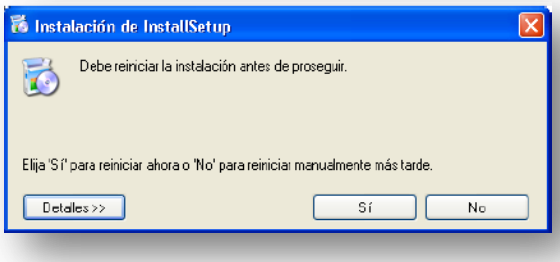


Ilustración 9.4. Instalación .NETFramework 4.0 finalizada

PASO 2. Al hacer doble clic sobre el archivo ejecutable "*InstallSetup.msi*" comenzará el proceso de instalación de la herramienta. Se iniciará un asistente que guiará al usuario en el proceso de la instalación.

Ilustración 9.5. Instalar herramienta

En el primer paso de la instalación se deberá escoger la ruta donde instalar la aplicación. Esa carpeta o directorio deberá tener permisos de escritura para poder instalarlo. En caso de que el PC donde se vaya a instalar tenga más de una cuenta de usuario, se podrá escoger la opción de instalarlo solo para el usuario actual o para todos los usuarios existentes. Por defecto viene marcada la opción "*Para todos los usuarios*".

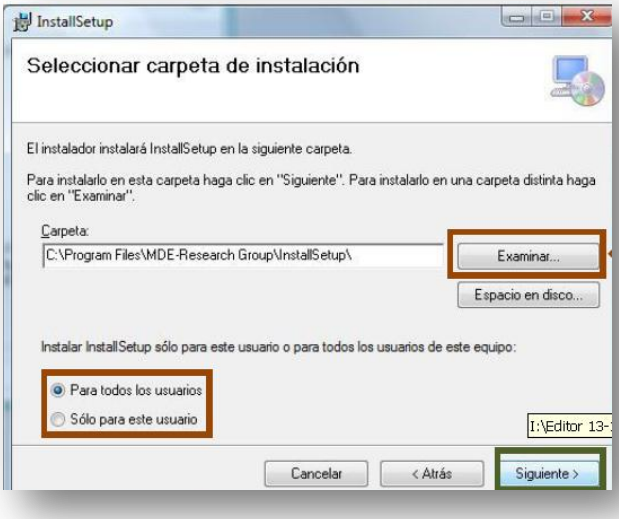


Ilustración 9.6. Seleccionar carpeta instalación

Si se desea modificar la ruta que viene por defecto para instalar la herramienta, se deberá hacer clic sobre "*Examinar*" y se abrirá una ventana emergente donde se podrá escoger otra ruta distinta. Para el ejemplo que se está mostrando se ha escogido la carpeta temporal de Windows "*Temp*".

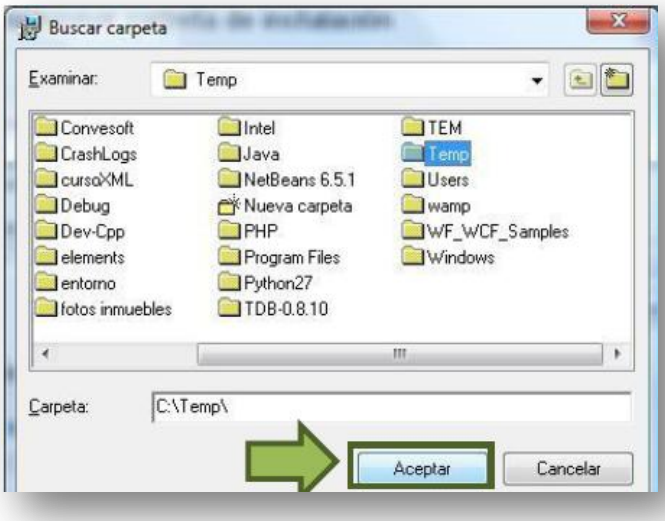


Ilustración 9.7. Carpeta Temp

Una vez escogida la nueva ruta se pulsará "Aceptar" y se cerrará esa ventana emergente y se dará a "Siguiente" y comenzará el proceso de instalación. Cuando este proceso termine se creará un acceso directo a la herramienta en el escritorio.

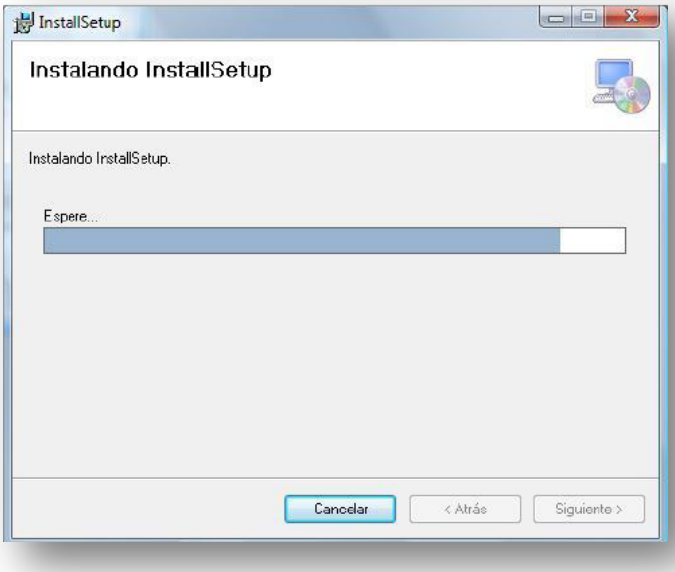


Ilustración 9.8. Instalando

9.2 Manual de Ejecución

Al finalizar la instalación se creará automáticamente un acceso directo en el escritorio del PC. Haciendo doble clic sobre él se abrirá la aplicación.

Ilustración 9.9. Icono aplicación

Otra forma de ejecutarla es ir al directorio donde se indicó que se realizaría la instalación y hacer doble clic sobre el ejecutable "Gade4all.exe"(1). En ese directorio se encontrarán también las carpetas con los recursos (2) necesarios para la ejecución de Gade4all.

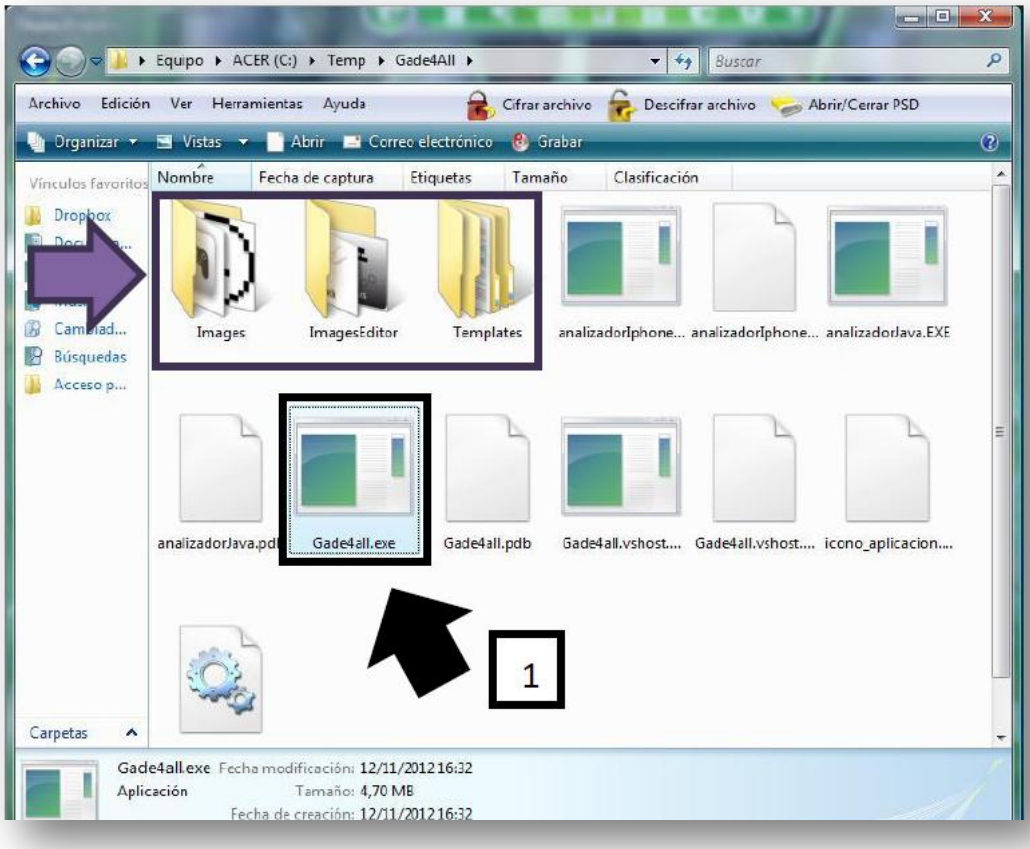


Ilustración 9.10. Ejecutar aplicación

9.3 Manual de Usuario

9.3.1 Requisitos

Para poder realizar la instalación de la herramienta es necesario:

- .NET Framework v4.0 o superior.
- Que la carpeta donde se van a almacenar los juegos creados no tenga restringidos los permisos de escritura.

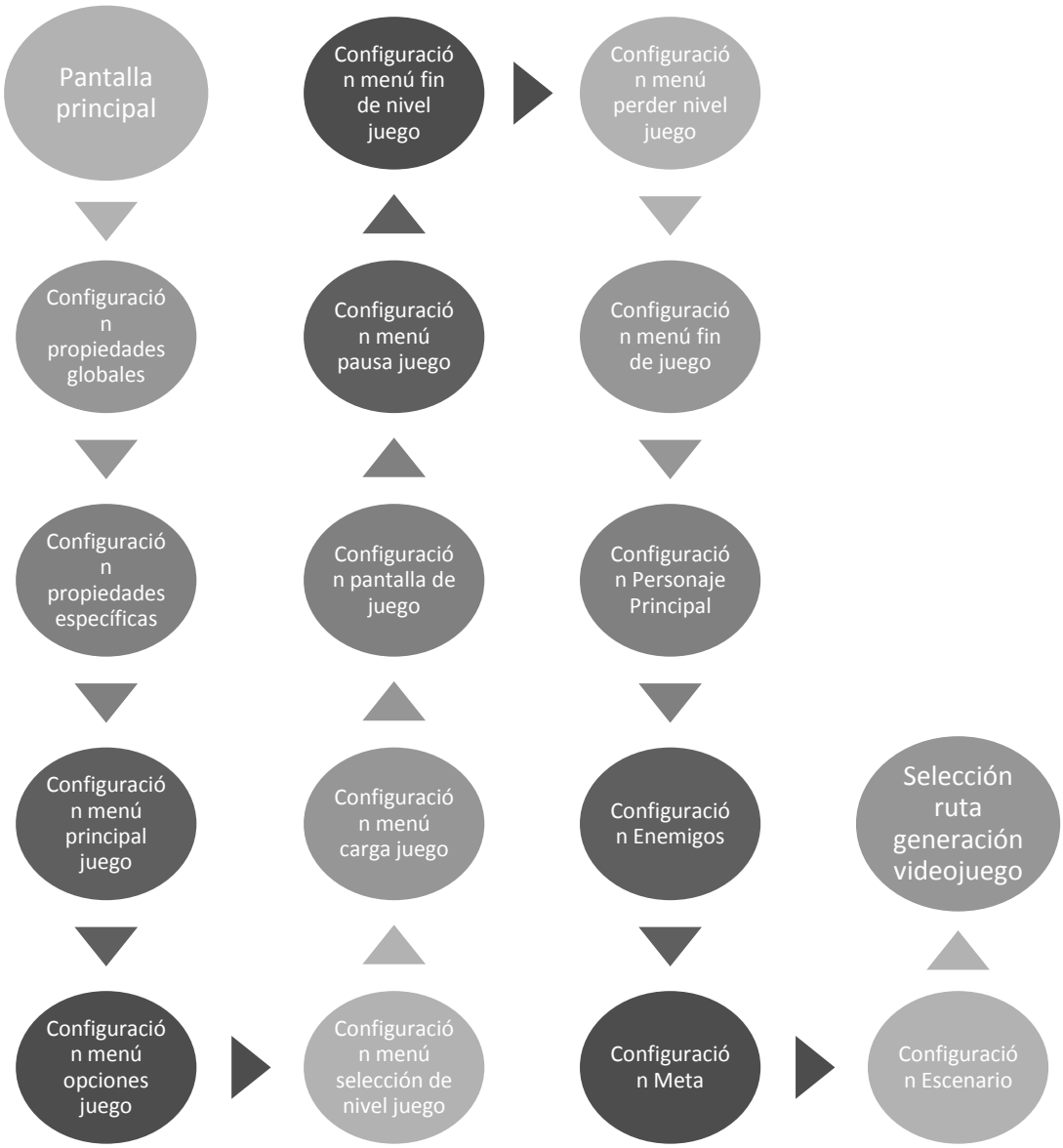
Una vez finalizado el proceso de instalación de la aplicación correctamente, para poder ejecutar y probar los juegos creados con la herramienta es necesario tener instalado el siguiente software:

- Para la plataforma Android:
  - Entorno de desarrollo Eclipse (http://www.eclipse.org)
  - Android Software Development Kit (Android SDK) instalado en Eclipse
- Para la plataforma iOs:
  - Entorno de desarrollo Xcode (http://developer.apple.com/xcode/)

9.3.2 Tutorial de creación de un videojuego con la herramienta Gade4all

A continuación se explican de forma detallada los pasos a seguir para la creación de un videojuego con la aplicación Gade4all. Se analizará el funcionamiento de cada pantalla paso a paso.

En el gráfico siguiente se muestra la estructura de las pantallas de la herramienta.



9.3.2.1 Pantalla inicial

Al ejecutar la aplicación se abrirá la siguiente pantalla:

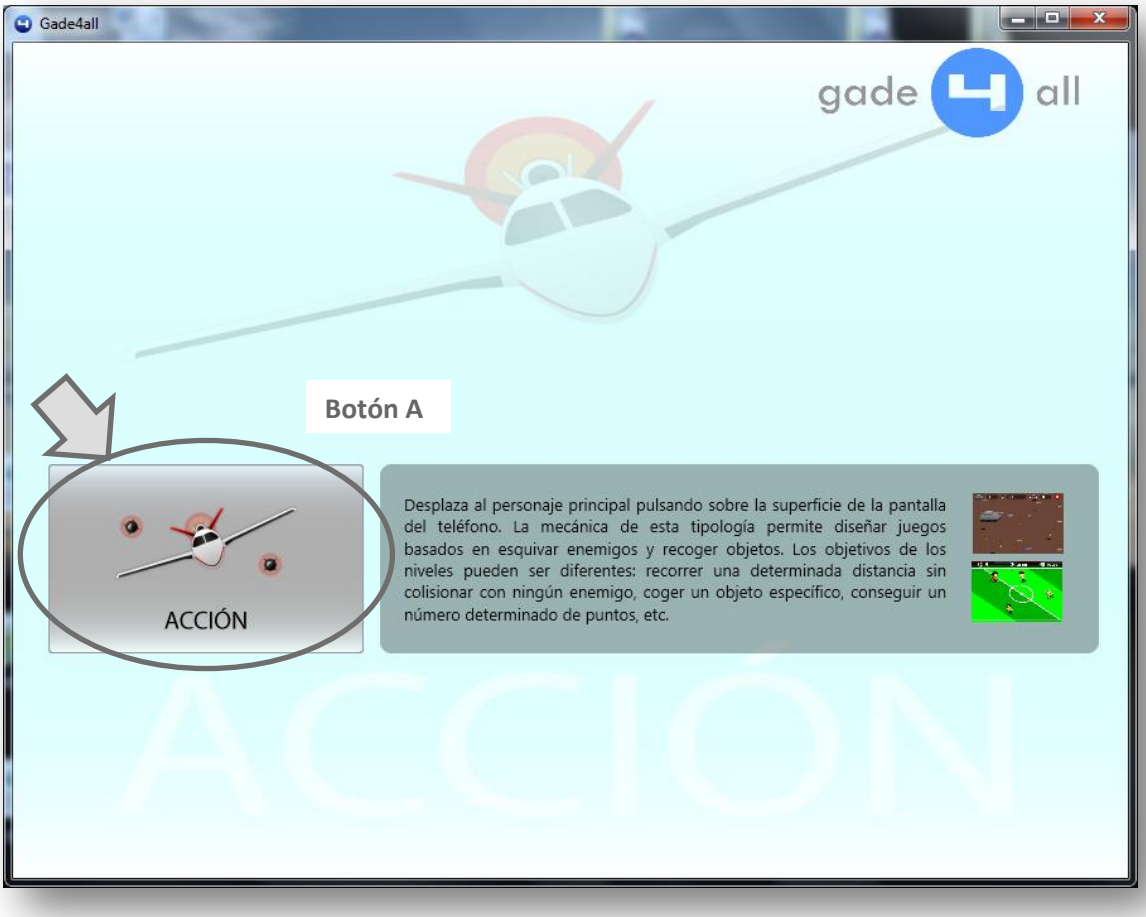


Ilustración 9.11. Pantalla inicial

Para comenzar con el proceso de creación del videojuego desde cero bastará con hacer clic sobre el botón A. A la derecha de este botón se encuentra una breve descripción de la tipología del juego. En este caso la opción elegida es juegos de tipo Touch con personaje principal. Al hacer clic se abrirá un cuadro de diálogo en el que se escogerá la ruta donde se almacenarán los juegos que se vayan a crear. Si se desea cargar algún juego creado anteriormente, bastaría con seleccionar en ese cuadro de diálogo, el directorio que contiene el juego, y este se cargaría automáticamente. Si se carga un juego ya creado, se tiene que cargar uno de la misma tipología que se ha seleccionado.

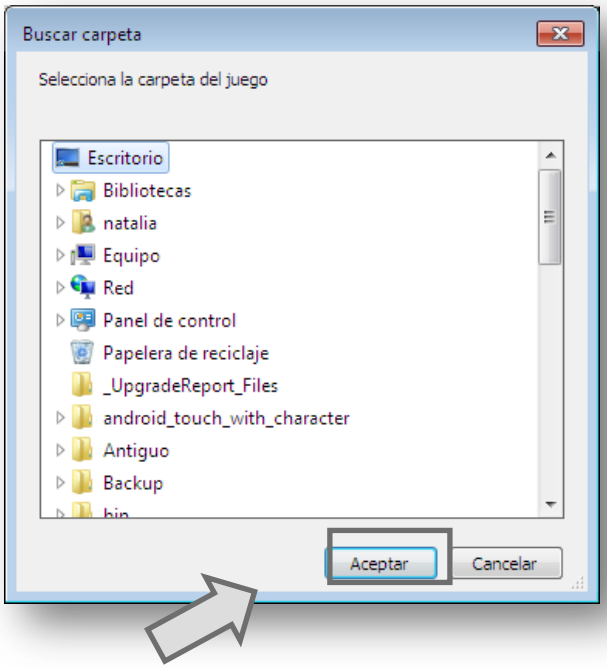


Ilustración 9.12. Selección Workspace

Una vez escogido el directorio, hacer clic sobre el botón "*Aceptar*".

A continuación se va a explicar el funcionamiento de la herramienta, así como la forma de proceder en cada pantalla. Esta explicación sirve tanto para crear un juego desde cero, como para modificar uno cargado.

9.3.2.2 Menú de la aplicación

Todas las pantallas de configuración, se abren dentro de una ventana contenedora, y esta tiene un menú en la parte superior (A) y una barra (B) en la que se muestra un botón por cada pantalla que el usuario ya ha visitado y que son también accesos directos a cada pantalla. Si se hace clic sobre uno de esos botones se cargará la pantalla correspondiente.

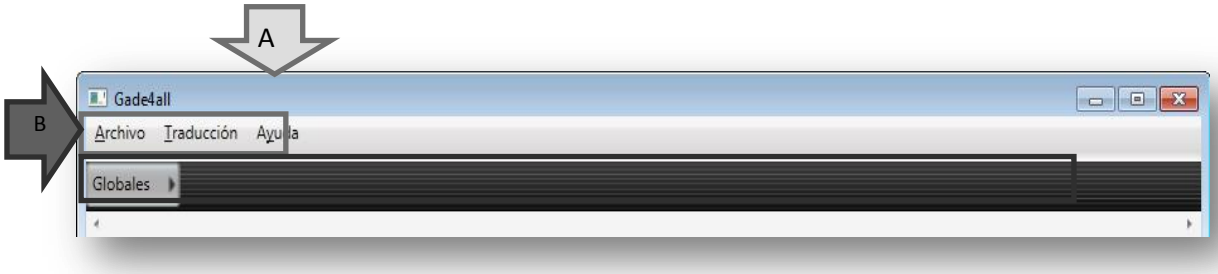


Ilustración 9.13. Menú de la aplicación

Este sería el aspecto de la barra de botones (B) cuando se ha visitado más de una pantalla:

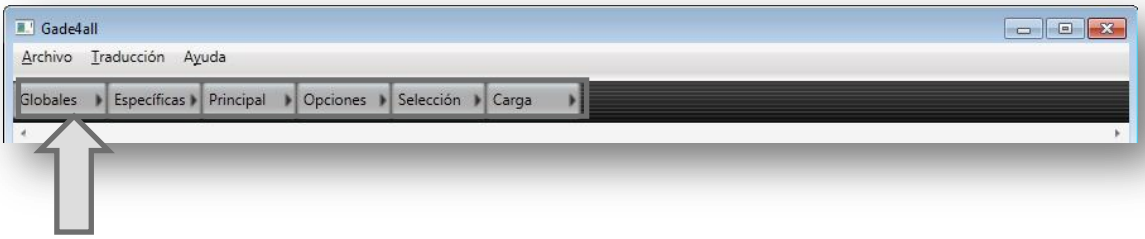


Ilustración 9.14. Barra de accesos directos

En el menú de la aplicación (A) hay dos elementos: "*Archivo*" y "*Traducción*". Al hacer clic sobre "*Archivo*" se despliega el siguiente menú:

Ilustración 9.15. Menú Archivo

En el cual hay disponibles dos opciones:

- **Cerrar:** Se cerrará la aplicación.
- **Guardar:** Se guardará el estado de la pantalla en la que se encuentre el usuario en ese momento.

El elemento "Traducción" aún no tiene implementada funcionalidad, pero su objetivo será poder seleccionar un idioma y modificar así el de la herramienta.

En la parte inferior de la ventana contenedora de la aplicación hay dos botones "*Atrás*" y "*Siguiente*" que permitirán al usuario navegar por las pantallas de la aplicación.

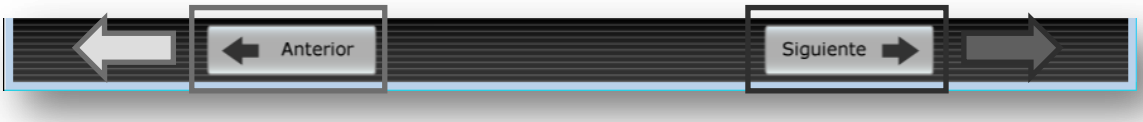


Ilustración 9.16. Menú navegación

Al hacer clic en "*Atrás*" se volverá a la pantalla anterior y se cargará el último estado de esa pantalla. En cambio, si se pulsa "*Siguiente*", se abrirá la siguiente pantalla y se guardará el estado de la página en la que se encontraba en el momento de pulsar el botón.

9.3.2.3 Pantalla configuración inicial, propiedades globales del juego

En esta pantalla se pueden configurar las propiedades globales del juego. Se va a analizar cada una de ellas.

Ilustración 9.17. Configuración propiedades globales

- **Paso 1:** Introducir el nombre del juego. Este campo es obligatorio.
- **Paso 2:** Elegir el icono de la aplicación. Al pulsar sobre el botón "*Elegir Icono*" se abrirá una ventana emergente para elegir el archivo en formato .png que se utilizará como icono. Las dimensiones de esta imagen serán de 57 x 57 píxeles. Una vez elegido el icono se muestra una miniatura del mismo. Por defecto viene cargado un icono genérico.
- **Paso 3:** El nombre del paquete de la aplicación. Se rellena automáticamente con el mismo nombre que el juego. Se puede modificar pero no puede contener espacios en blanco, y tiene que empezar por la cadena "com.". Por defecto es "com.nombredeljuego".
- **Paso 4:** En caso de que el juego a crear vaya a tener publicidad, se debe marcar la casilla "*Tiene publicidad*" y rellenar el identificador de la publicidad para desplegar en Android.
- **Paso 5:** Escoger la orientación del juego. Marcar una de las dos opciones. La orientación por defecto es "*Horizontal*".
- **Paso 6:** Sonido de los menús. Haciendo clic en "*Elegir sonido*", se mostrará una ventana emergente para escoger el archivo de audio en formato .mp3, que será la melodía o sonido

que se reproducirá en los distintos menús del juego. No es un campo obligatorio. Dispone de tres botones (Stop, Play y Pause) para poder reproducir el sonido que se acaba de introducir.

Para pasar a la siguiente pantalla, se pulsaría sobre el botón "*Siguiente*".

9.3.2.4 *Ejemplo pantalla Propiedades Globales configurada*

Ilustración 9.18. Ejemplo pantalla propiedades globales

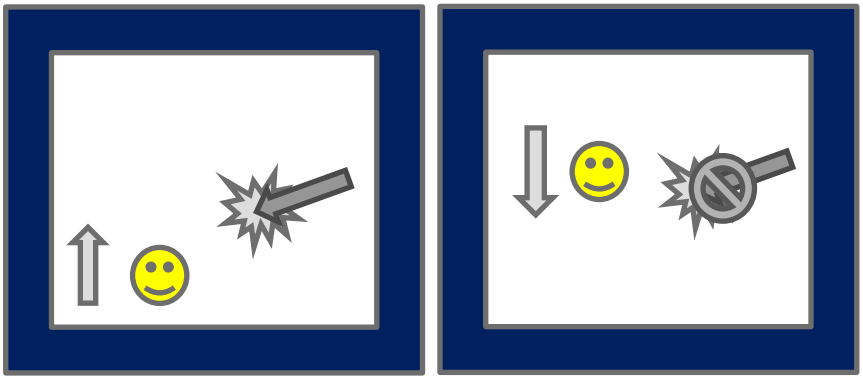
9.3.2.5 Pantalla configuración de propiedades especificas

En esta pantalla se van a configurar las propiedades específicas del juego, es decir, aquellas que van a determinar que sea de un tipo de juego u otro.

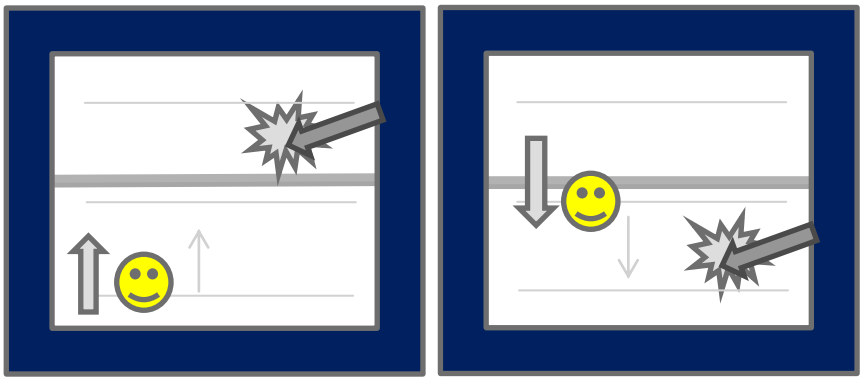
Ilustración 9.19. Configuración propiedades específicas

Los pasos a seguir para configurar esta pantalla serían los siguientes:

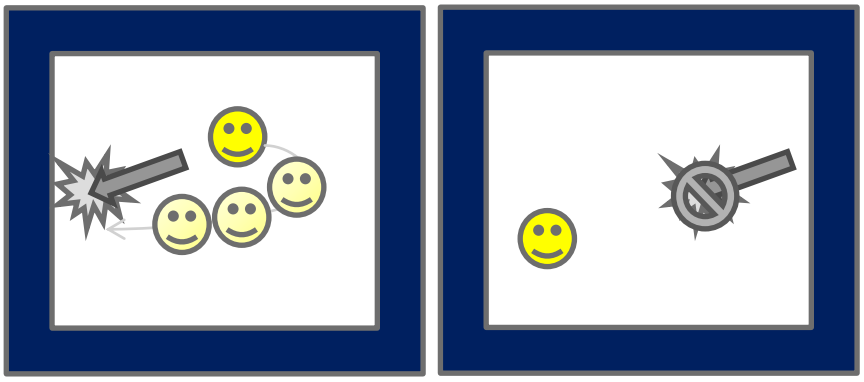
- **Paso 1:** Escoger el tipo de juego. Por defecto aparece marcado el tipo de juego es "JetPack".
    - **JetPack:** En este tipo de juego, el personaje principal se desplazará hacia arriba mientras el usuario mantenga pulsado el dedo sobre la pantalla del dispositivo móvil. En el momento que deje de tocarla, el personaje bajará hasta la parte inferior de la pantalla.



- **Carriles:** El personaje principal se desplazará por carriles. Cada vez que el usuario toque la mitad superior de la pantalla el personaje subirá un nivel, así hasta llegar hasta el último carril. En caso de tocar la mitad inferior, este bajará un nivel por cada toque que se realice.



- **Sigue al dedo:** El personaje principal seguirá recorrido que haga el dedo del usuario por la pantalla. En el momento que el dedo deje de tocar la pantalla, el personaje principal dejará de moverse.



- **Paso 2:** Elegir qué elementos se van a tener en cuenta en las puntuaciones del juego. Si se marca "*Si*" será que sí contiene ese marcador, si se elige "*No*" no contendrá ese tipo de marcador. Por defecto, el juego tendrá todos los marcadores.
  - **Metros:** Para contar los metros que recorre el personaje principal del juego.
  - **Puntos:** Para contabilizar los puntos que se hacen en la partida.
  - **Vidas:** Para mostrar las vidas de las que dispone el personaje principal.

- **Paso 3:** Escoger si el juego tendrá elemento de fin de nivel, es decir, si el personaje principal tiene que chocar con un elemento de fin, una meta por ejemplo, para poder superar el nivel; o también si aquellos enemigos que no sean destruidos, al salirse por algún lado de la pantalla, le restan vidas al personaje principal. Para escoger cualquiera de las dos opciones se marca la casilla correspondiente.

Si se pulsase el botón "*Atrás*" se volvería a la pantalla anterior de configuración de propiedades globales y aparecerían los datos tal y como los dejó configurados el usuario anteriormente. Al hacer clic en "*Siguiente*" se va a la siguiente pantalla.

9.3.2.6 Ejemplo pantalla propiedades específicas configurada

Ilustración 9.20. Ejemplo pantalla propiedades específicas

9.3.2.7 Pantallas configuración pantallas de los menús del juego

Las siguientes pantallas son para configurar los diferentes menús del juego. Todas estas pantallas tienen una estructura común, que se va a explicar a continuación de forma detallada.

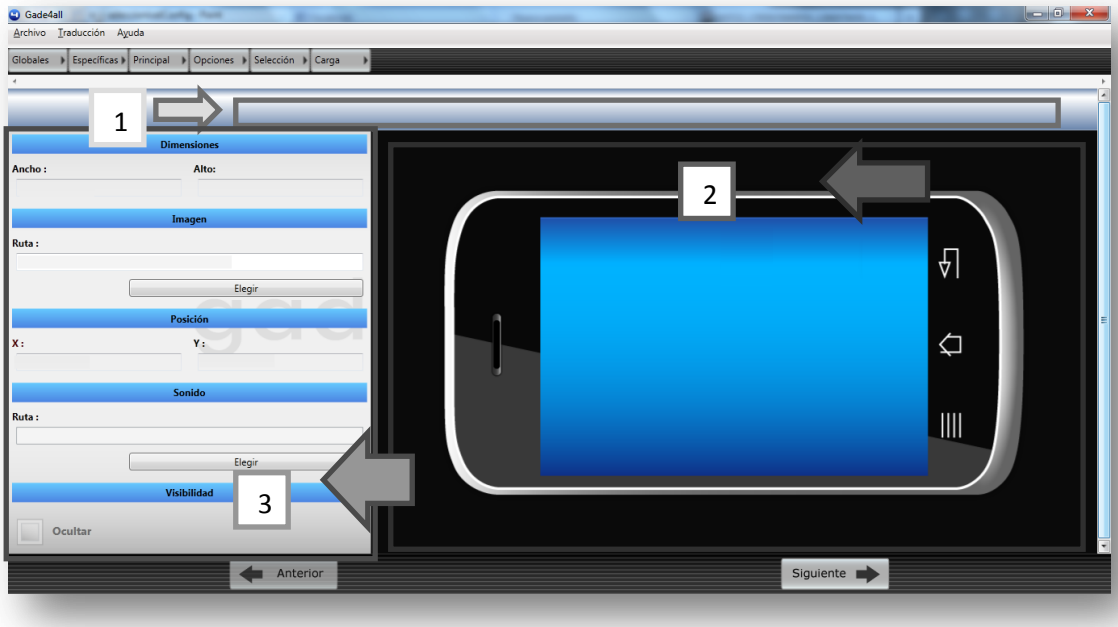


Ilustración 9.21. Estructura pantallas menús

Estas pantallas de configuración se dividen en las siguientes cuatro secciones:

- **Sección 1:** En esta parte aparecerá el nombre de la pantalla actual. Pantalla principal, Pantalla selección de nivel, etc. Por ejemplo:

Ilustración 9.22. Nombre de la pantalla

- **Sección 2:** Pizarra. En esta sección de la pantalla, aparecerían cargados los elementos (fondo principal y botones) que tendría cada menú. Se pueden configurar cada uno de esos elementos, modificando las propiedades que aparecen en la sección **3**. Al hacer clic sobre alguno de los elementos, en la sección 3 se cargarían automáticamente los datos de dicho elemento. El contenido de esta sección es diferente en cada pantalla, ya que cada menú tiene unos botones diferentes. Al elemento fondo no se le puede asignar texto, con lo que es recomendable escoger una imagen con texto integrado.
- **Sección 3:** En esta parte de la pantalla están los diferentes parámetros configurables de cada elemento de los menús. Se va a explicar cada parámetro con detalle:

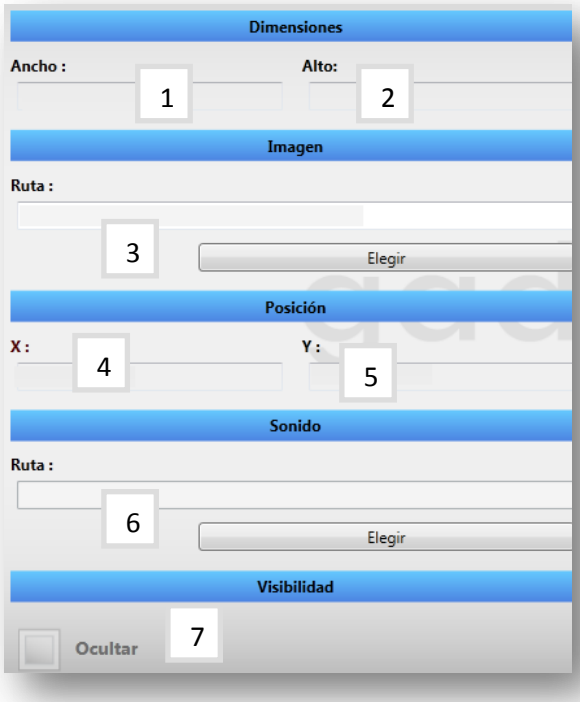


Ilustración 9.23. Parámetros configurables de los elementos de los menús del juego

- **1 Ancho:** En este cuadro de texto se introducirá el ancho del elemento. Tiene que ser un número entero. Al hacer clic sobre algún elemento este se rellenará automáticamente con el ancho del mismo. Al modificar su valor, el elemento seleccionado modificará automáticamente su ancho por el del nuevo valor introducido. Se mide en píxeles.
- **2 Alto:** En este cuadro de texto se introducirá el alto del elemento. Tiene que ser un número entero. Al hacer clic sobre algún elemento este se rellenará automáticamente con el alto del mismo. Al modificar su valor, el elemento seleccionado modificará automáticamente su alto por el del nuevo valor introducido. Se mide en píxeles.
- **3 Imagen:** Si se desea modificar la imagen de un elemento, se tendría que pulsar el botón "*Elegir*" y se abriría un cuadro de diálogo en el que se escogería la nueva imagen en formato .png. Al elegir la imagen, en el cuadro de texto se cargaría la ruta de dicha imagen en disco. Inicialmente aparece la ruta de la imagen por defecto que tenga el elemento.
- **4 Posición Eje X:** En este cuadro de texto se introducirá la posición en el eje X del elemento. Tiene que ser un número entero. Al hacer clic sobre algún elemento este se rellenará automáticamente con la posición X del mismo. Al modificar su valor, el elemento seleccionado modificará automáticamente su posición a la nueva coordenada del eje X introducida.
- **5 Posición Eje Y:** En este cuadro de texto se introducirá la posición en el eje Y del elemento. Tiene que ser un número entero. Al hacer clic sobre algún elemento este se rellenará automáticamente con la posición Y del mismo. Al

modificar su valor, el elemento seleccionado modificará automáticamente su posición a la nueva coordenada del eje Y introducida.

- **6 Sonido:** Esta propiedad no es aplicable a los fondos. Es el sonido que se desea que tengan los botones al ser pulsados. Si se pulsa en "Elegir", se abrirá un cuadro de diálogo para seleccionar el sonido en formato .mp3. No es obligatorio rellenar este campo.
- **7 Visibilidad:** Si se desea ocultar alguno de los elementos del menú basta con seleccionar el elemento y marcar la casilla "Ocultar", de esta forma ese elemento no aparecerá en esa pantalla del juego generado.

Un ejemplo de pantalla con los valores configurables de cada elemento de los menús sería la siguiente:

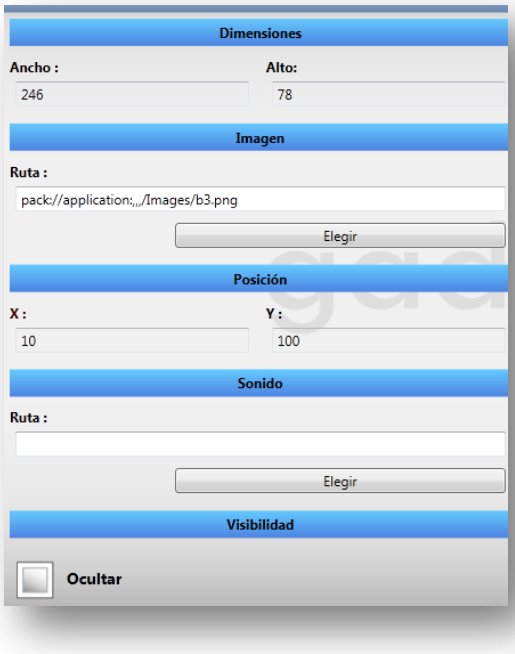


Ilustración 9.24. Ejemplo de parámetros configurables de los elementos de los menús del juego

Al darle al botón "*Siguiente*" en alguna de las pantallas de los menús, se guardará el estado de la pantalla actual, se abrirá la siguiente página y se generará el fragmento de la instancia del DSL correspondiente a la pantalla en la que se esté.

9.3.2.8 Pantalla Principal

La primera pantalla que se va a configurar es la del menú principal de juego. Este menú estará compuesto de una imagen de fondo y de tres botones: "*Jugar*" (para pasar a la pantalla de selección de nivel), "*Opciones*" (para ir a pantalla de las opciones del juego), y "*Salir*" (para salir del juego). De cada uno de estos 4 elementos se podrán configurar los valores que se han visto en el punto anterior.

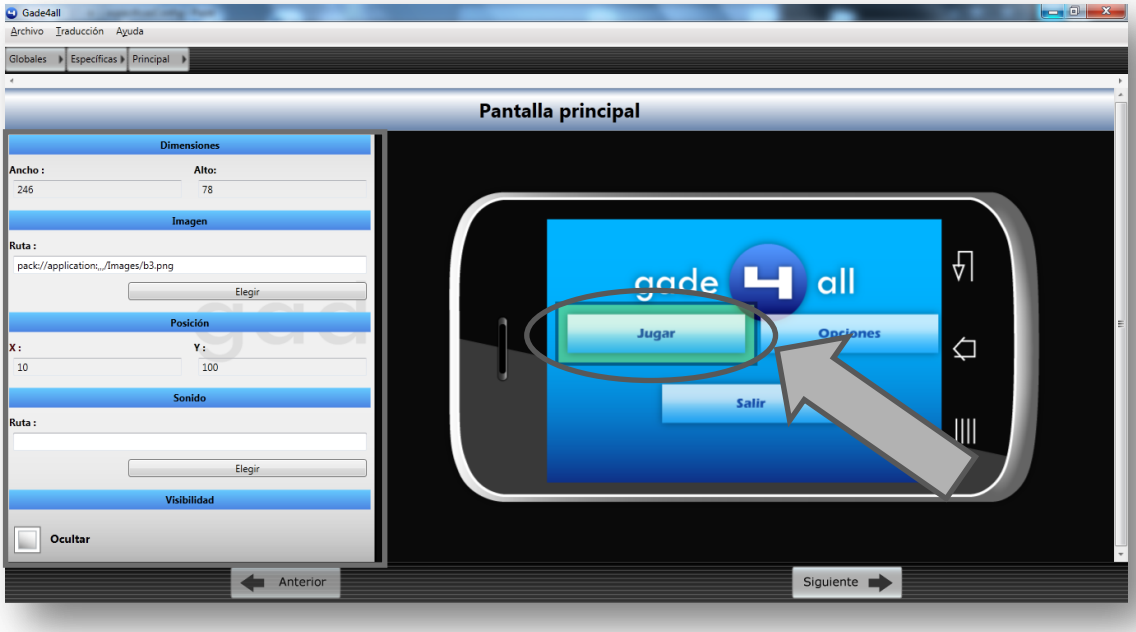


Ilustración 9.25. Configuración pantalla principal del juego

En la parte derecha de la pantalla, la de configuración de parámetros, inicialmente aparecen cargados los valores correspondientes al botón de "*Jugar*".

Para modificar la configuración de alguno de los elementos de la pantalla bastaría con hacer clic sobre uno de ellos y modificar sus valores, que se cargan automáticamente en la parte derecha al hacer clic sobre él. Al seleccionar un elemento, este aparecerá rodeado por un rectángulo de color verde, para saber en todo momento qué elemento se tiene seleccionado.

Al modificar alguno de los valores, este se actualiza automáticamente, para que el usuario pueda ver el resultado y volverlo a modificarlo si lo desea.

Como se ya se ha comentado, a la imagen de fondo no se le puede asignar texto, por lo que es recomendable escoger una imagen con texto integrado, bien para incluir un logo o para indicar qué pantalla es. Esto es igual para todas las pantallas.

Ahora se va a modificar la imagen de los botones "*Opciones*" y "*Salir*".

Ilustración 9.26. Configuración botones pantalla menú principal

Si se desea colocar los botones en otra posición, se puede modificar a mano el valor de sus coordenadas en la parte derecha de la pantalla o bien haciendo clic sobre él y arrastrándolo por la pantalla hasta la posición que se desee.

Un ejemplo de una pantalla del menú principal del juego configurada sería como la siguiente:

Ilustración 9.27. Ejemplo pantalla principal configurada

En el resto de pantallas, para configurar sus elementos la forma de proceder es la misma que en esta pantalla, por lo que solamente se mostrará un ejemplo de como quedaría cada pantalla con los elementos configurados por el usuario de la herramienta.

9.3.2.9 Pantalla Opciones

En esta pantalla se configurará la pantalla correspondiente al menú de opciones del juego, desde el que se podrá desactivar (1) o activar (2) la música del mismo.

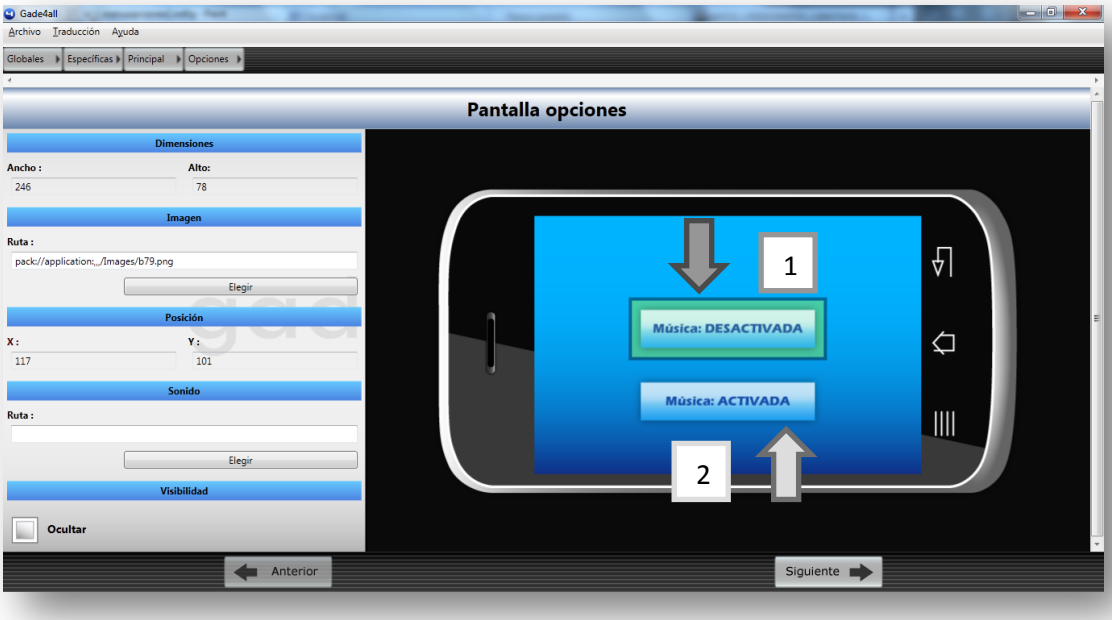


Ilustración 9.28. Configuración pantalla de opciones del juego

Configurada por el usuario quedaría así:

Ilustración 9.29. Pantalla opciones juego configurada

9.3.2.10 Pantalla Selección de nivel

En esta pantalla se configurará el menú de selección de nivel del juego, en la que se podrá escoger el nivel del juego que se desee. Contendrá la miniatura del nivel (1), en la que irán las diferentes miniaturas correspondientes a cada nivel, un botón "Jugar" (2) y unos botones (3) que permitirán navegar por las miniaturas de las niveles para posteriormente seleccionar uno.

Ilustración 9.30. Configuración pantalla selección de nivel del juego

De la miniatura del nivel solamente es necesario configurar la posición en el eje X y eje Y y sus dimensiones, ya que la imagen se elegirá cuando se configuren los niveles del juego más adelante.

Configurada por el usuario quedaría así:

Ilustración 9.31. Pantalla selección de nivel juego configurada

9.3.2.11 Pantalla Loading

En esta pantalla se configura la pantalla de carga del juego, que aparece mientras se carga el nivel después de seleccionarlo en la pantalla anterior. En esta pantalla solamente se puede configurar la imagen de fondo. No tiene botones.

Ilustración 9.32. Configuración pantalla de carga del juego

Un ejemplo de la pantalla configurada por el usuario sería:

Ilustración 9.33. Pantalla Loading configurada

9.3.2.12 Pantalla Juego

Aquí se podrá configurar la pantalla donde se juega. Contiene cinco elementos:

- Imagen de fondo para la barra de estado donde se pondrán los marcadores (1)
- Icono de las vidas y posición del marcador de vidas (2)
- Icono de puntos y posición del marcador (3)
- Icono de metros recorridos y posición del marcador (4)
- Botón de pausa (5)

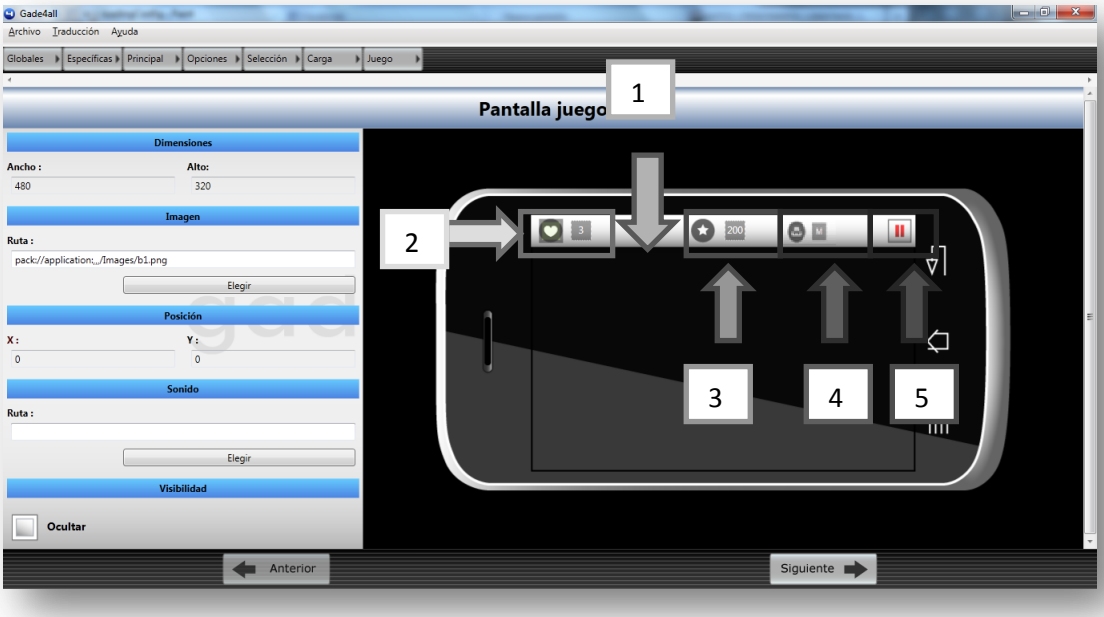


Ilustración 9.34. Configuración pantalla juego

Ilustración 9.35. Icono y marcador

La imagen de fondo que se añade tiene que ser de la siguiente forma:

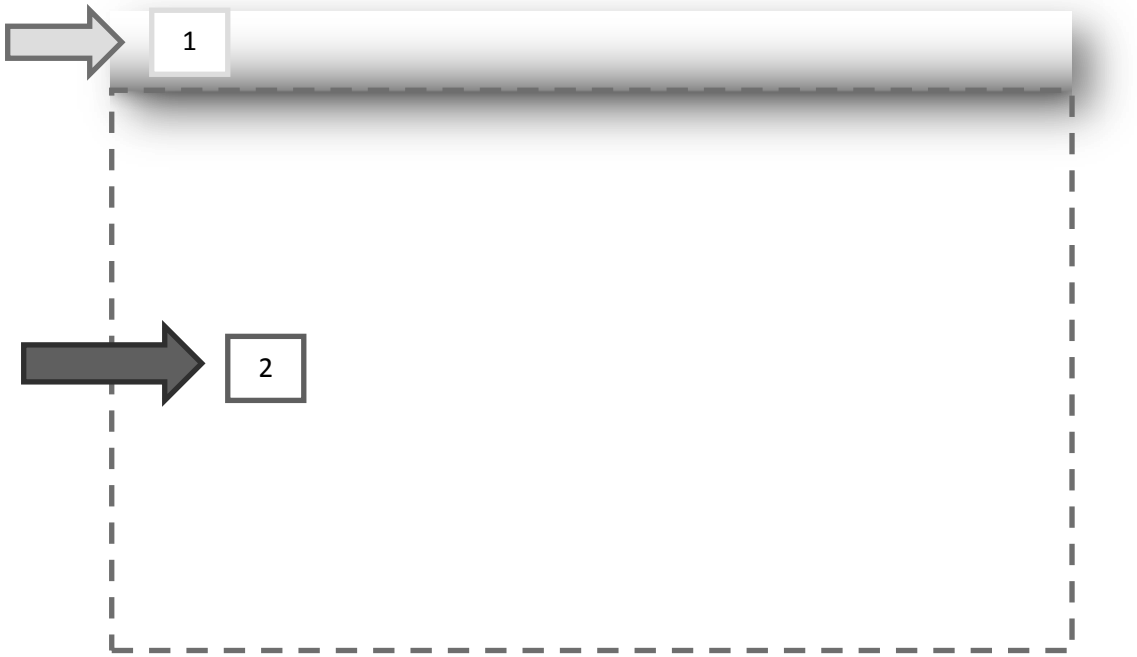


Ilustración 9.36. Imagen barra de estado de pantalla de juego

Si la orientación escogida es horizontal la imagen deberá tener un ancho de 480 y un alto de 320 píxeles. Si, en cambio, se escogió vertical tendrá que ser de 320x480 píxeles.

La imagen estará dividida en dos zonas:

- La zona 1 es donde se situarán los marcadores y tiene que tener fondo.
- La zona 2 tiene que ser transparente, ya que si tiene fondo no se verán los elementos que se hayan colocado en el nivel, pues esta imagen se pinta la última de todas.

En cuanto a la configuración de los iconos (1) y la posición de los marcadores (2), de los primeros se pueden configurar todos los parámetros, y del segundo solamente es necesario configurar la posición X e Y, el texto que contienen, la inicial del marcador al que se corresponde, solamente es orientativo, ya que luego en el juego ese marcador solamente contendrá un número entero, sin imágenes ni texto adicional.

En esta pantalla están los marcadores de iconos de vidas, puntos y metros, pero solamente aparecerán cargados aquellos que el usuario eligió en la pantalla de configuración de propiedades específicas del juego, es decir, si se eligió que el juego tenga puntos y vidas, solamente se podrán configurar esos dos, el de metros no aparecería.

Se puede ocultar solamente el icono y dejar el marcador o viceversa.

Configurándolos quedaría así:

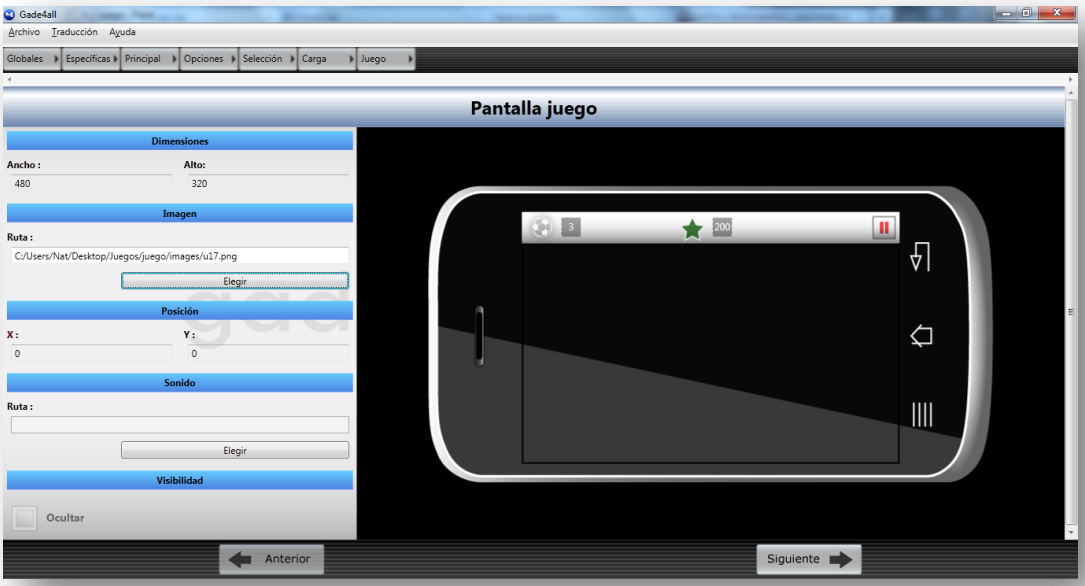


Ilustración 9.37. Pantalla juego configurada

9.3.2.13 Pantalla Pausa

En esta pantalla se configura la pantalla de pausa del juego, que se muestra cuando se aprieta el botón de pausa de la pantalla del juego. Está compuesta por un fondo y dos botones: "*Reanudar*" (1) y "*Salir*" (2). Con el primero se reanudaría el juego y con el segundo se saldría del nivel y se iría al menú principal.

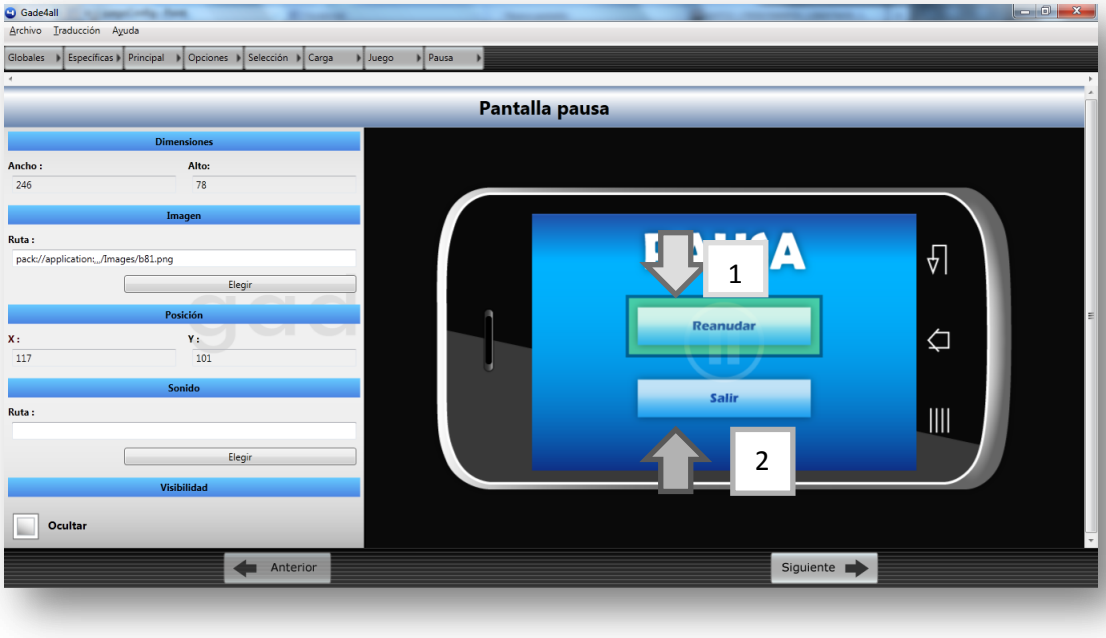


Ilustración 9.38. Configuración pantalla pausa del juego

Configurada por el usuario tendría un aspecto como este:

Ilustración 9.39. Pantalla pausa configurada

9.3.2.14 Pantalla Fin de nivel

En esta pantalla se puede configurar la pantalla de victoria, la que se muestra cuando se supera un nivel con éxito. Contiene una imagen de fondo, dos botones "*Repetir nivel*" (1) y "*Siguiente nivel*" (2) y un marcador (3) donde se mostrarán los puntos obtenidos en la partida.

Ilustración 9.40. Pantalla fin de nivel con éxito

Del marcador solamente se deberá configurar la posición X e Y y las dimensiones, la palabra "Puntos" no se reflejará en el juego, simplemente sirve para indicar donde aparecerán los puntos. La imagen de fondo del marcador tampoco aparecerá, solamente sirve para orientar al usuario sobre el tamaño que tendrá el marcador. Es conveniente no ponerle unas dimensiones muy pequeñas porque si se obtienen puntos de varias cifras al terminar el nivel podrían no entrar dentro del marcador y aparecerían cortados.

Configurada quedaría con este aspecto:

Ilustración 9.41. Pantalla fin de nivel configurada

9.3.2.15 Pantalla Derrota

Esta pantalla es similar a la anterior, Fin de nivel, la única diferencia reside en que en este caso en vez de tener un botón de "*Siguiente nivel*" hay uno de "*Seleccionar nivel*", que llevaría al menú de selección de nivel, ya que al perder no se puede pasar al siguiente nivel. Esta pantalla se mostrará cuando se pierda un nivel.

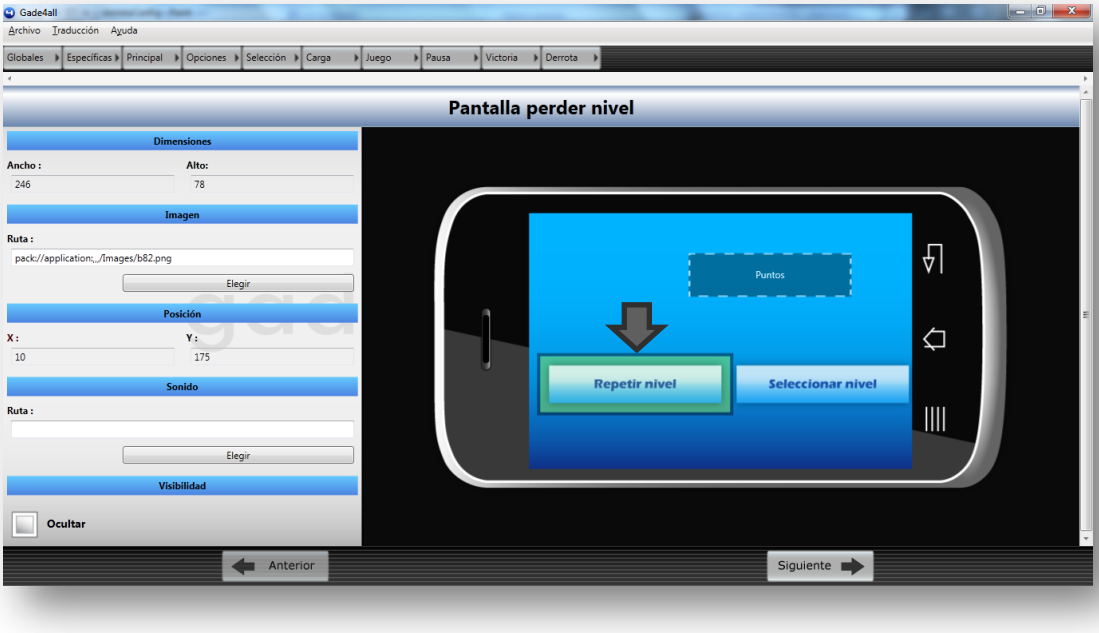


Ilustración 9.42. Configuración pantalla derrota

Una vez configurada tendría este aspecto:

Ilustración 9.43. Pantalla derrota configurada

9.3.2.16 Pantalla Fin de juego

La pantalla de fin de juego es el último menú del juego a configurar. En ella se podrá configurar el menú de fin de juego, que se mostrará cuando se hayan superado todos los niveles del juego. Contendrá una imagen de fondo y botón "*Volver al menú*" (1) que lleva al menú principal del juego.

Ilustración 9.44. Configuración pantalla fin de juego

Configurada quedaría así:

Ilustración 9.45. Pantalla fin de juego configurada

Al terminar de editar este menú de fin de juego, pulsando sobre el botón "*Siguiente*", se pasará a la parte de configuración de los niveles del juego y se generará la instancia del DSL correspondiente a las propiedades del juego y a la configuración de los diferentes menús, que se almacenará en la carpeta del juego. El fichero se llama *dsl.xml* y tiene el siguiente aspecto:

Ilustración 9.46. Instancia del DSL generada

9.3.2.17 Pantalla Personaje principal

Una vez configurados los diferentes menús del juego, el siguiente paso es la configuración del personaje principal del juego.

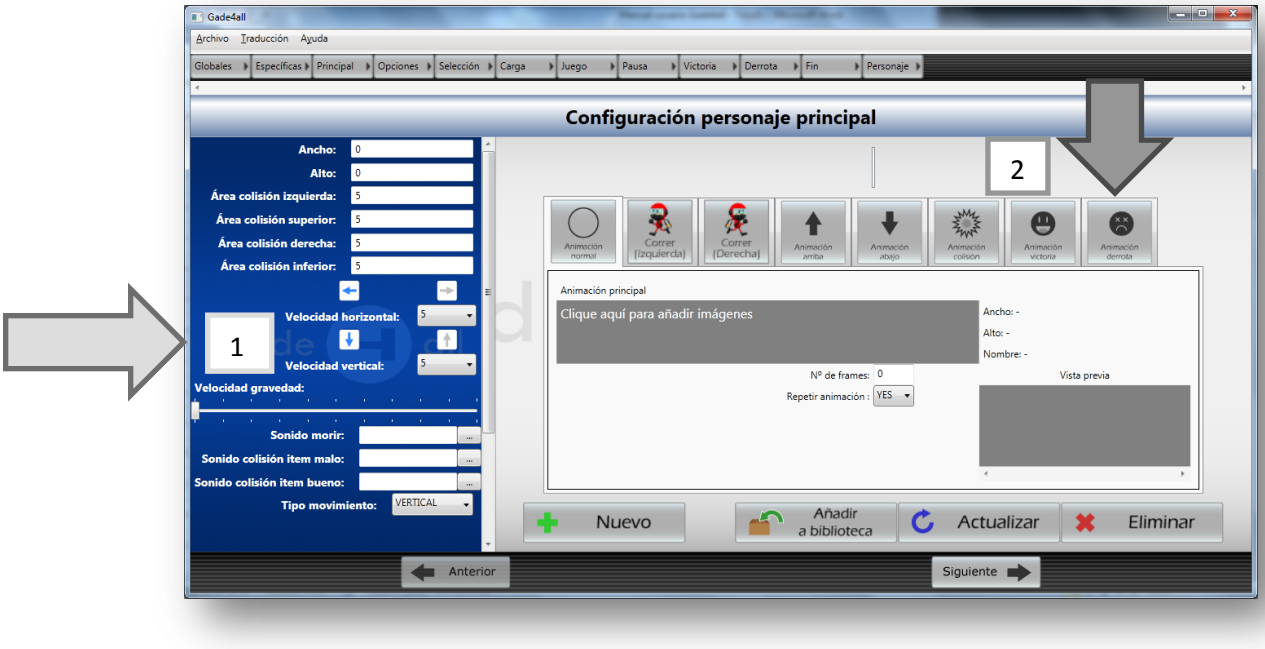


Ilustración 9.47. Configuración personaje principal del juego

Esta pantalla está dividida en dos partes, la primera se corresponde con la configuración de las propiedades del personaje (1) y en la segunda se asignan las imágenes de las animaciones del personaje (2). Se va a explicar el funcionamiento de cada una por separado.

En las propiedades del personaje vienen cargados por defecto unos valores, que el usuario deberá modificar en función del personaje que quiera crear. Al introducir las imágenes de las animaciones el ancho, el alto y el área de colisión se rellenan automáticamente, aun así se pueden modificar.

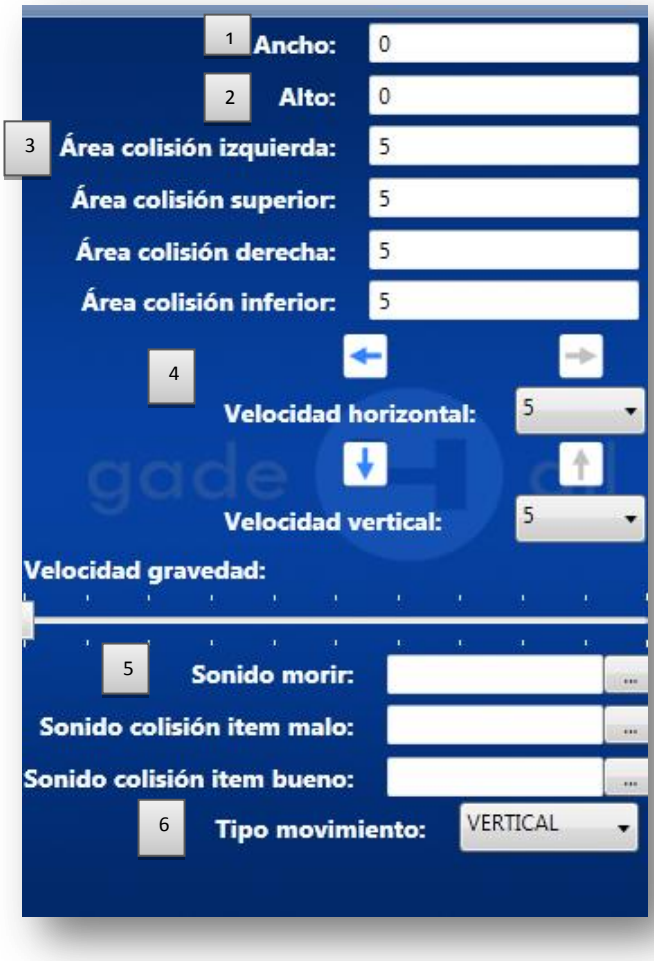


Ilustración 9.48. Configuración propiedades personaje principal

- **Paso 1:** El ancho que tendrá el personaje, debe ser un número entero positivo.
- **Paso 2:** El alto del personaje, debe ser un número entero positivo.
- **Paso 3:** Área de colisión. Este área se utilizar para saber si el personaje ha colisionado con algún enemigo o bien al ser tocado por el usuario en la pantalla. Define una región cuadrada del personaje, que será la que colisiona con otros elementos. Cuanto mayor sea el área más fácil será el juego, porque será más sencillo realizar la colisión; y por tanto a menor colisión más difícil será que el personaje colisione con algún objeto. Se tienen que definir los cuatro lados de la región de colisión. Todos ellos deben ser un valor entero positivo.

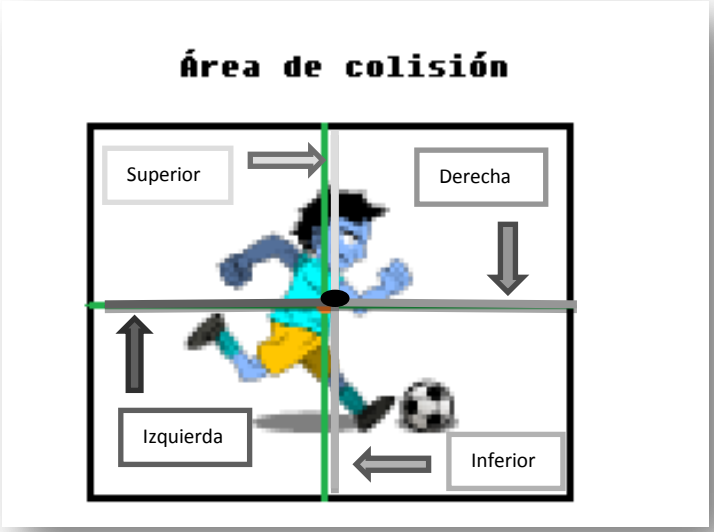


Ilustración 9.49. Área de colisión

Ilustración 9.50. Tamaños áreas de colisión

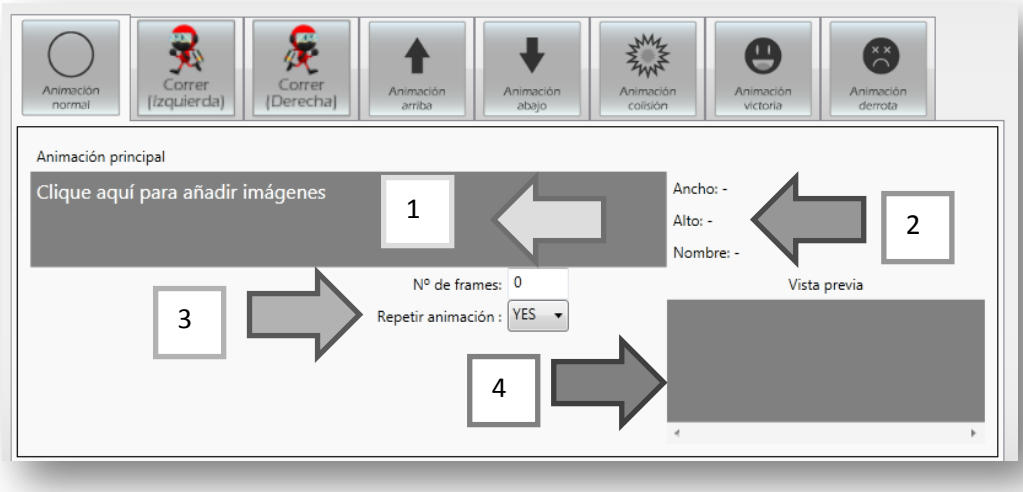
- **Paso 4:** Velocidades. El personaje tiene tres velocidades asociadas, en el eje X (movimiento horizontal), en el eje Y (movimiento vertical) y gravedad (velocidad con la que el objeto cae). Si se quiere que el personaje solamente se mueva en el eje Y se pone a 0 la velocidad en el eje X y viceversa. Si se da valor distinto de 0 a las dos, el personaje se podrá mover en todas direcciones. Las flechas indican en qué dirección se desplazará el personaje con la velocidad que se haya introducido. Si se pone la velocidad vertical y horizontal con valor 0, el personaje no se moverá, estará fijo en la pantalla.

- **Paso 5:** Sonidos. Cuando el personaje choca con un enemigo y pierde vidas (ítem malo) o gana puntos (ítem bueno), o es destruido se reproduce un sonido, y se deben configurar en este punto. Al hacer clic sobre el botón correspondiente a cada sonido se abrirá un cuadro de diálogo para seleccionar el archivo de audio en formato mp3 correspondiente.
- **Paso 6:** Tipo movimiento. El personaje se puede mover en vertical o en horizontal, para ello se debe seleccionar un tipo de movimiento u otro.

En la parte de selección de imágenes para las distintas animaciones del personaje se debe seleccionar una imagen para cada animación:

- Principal
- Izquierda
- Derecha
- Arriba
- Abajo
- Colisión
- Victoria
- Derrota

Es obligatorio seleccionar la animación principal, si algunas de las demás no se rellena se le asignará la imagen de la animación principal. La forma de proceder es la siguiente para todas las animaciones:



- **Paso 1:** Hacer clic sobre la caja gris (1), y se abrirá un cuadro de diálogo donde se escogerá la imagen de la animación. Extensión del archivo *.png*.
- **Paso 2:** Al introducir la imagen el ancho, el alto y el nombre del recurso de imagen se rellenará automáticamente. (2)
- **Paso 3:** Número de frames. Número de cuadros en los que se divide la imagen para formar la animación. Loop: si la animación se ejecutará continuamente o solamente una vez. (3)

Ilustración 9.51. Frames

Está imagen está formada por seis frames, es decir, la animación está formada por seis imágenes distintas.

- **Paso 4:** Una vez definidos los frames de la animación se podrá visualizar la ejecución de la animación en este cuadro. (4).

Ilustración 9.52. Pantalla personaje principal configurada

Se podrán crear tantos personajes principales como se desee, pudiendo así, por ejemplo, diseñar un videojuego en el que el personaje principal sea diferente en cada nivel. Para guardar el personaje recién creado bastaría con pulsar sobre el botón "Añadir a biblioteca" y se añadiría a la lista. Con cada personaje creado se pueden realizar varias acciones:

- Actualizar
- Añadir a biblioteca

- Eliminar
- Crear uno nuevo

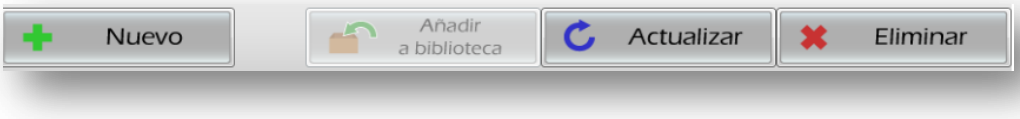


Ilustración 9.53. Opciones de la pantalla de creación de personaje principal

Ilustración 9.54. Personaje principal añadido a la lista

Si se pulsa el botón "*Siguiente*" sin haber añadido el personaje a la biblioteca no se almacenará y se perderán los cambios realizados en ese personaje.

9.3.2.18 Pantalla Enemigos

El siguiente paso a realizar es la creación y configuración de los enemigos. El funcionamiento de esta pantalla es el mismo que en la de configuración de personaje principal.

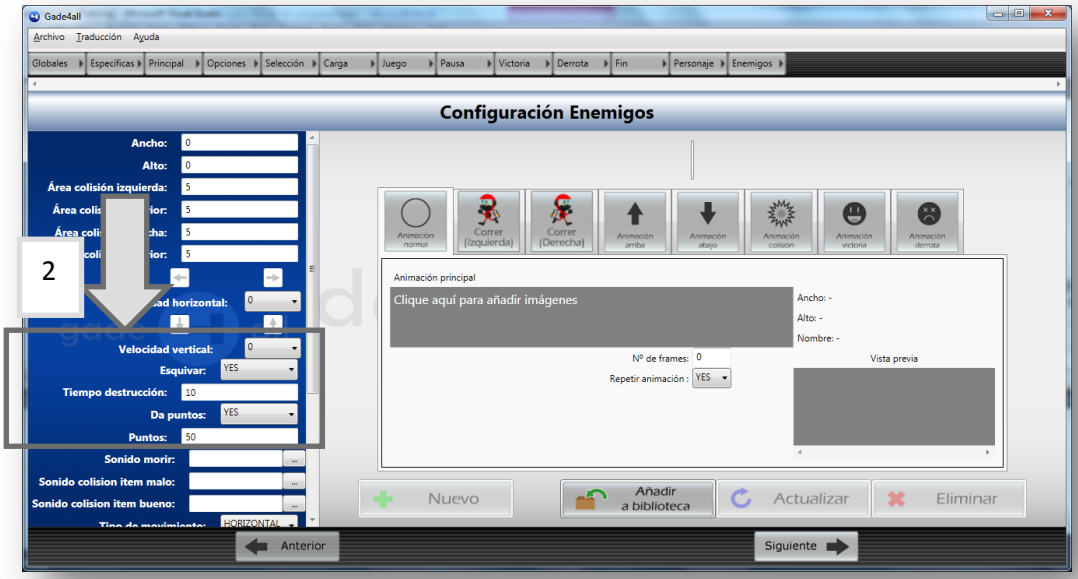


Ilustración 9.55. Configuración Enemigo

La parte correspondiente a la selección de los recursos de las animaciones es idéntica que en la pantalla anterior. También se pueden crear tantos como se deseen.

En cuanto a las propiedades del enemigo, hay cuatro nuevas respecto al personaje principal:

- **Esquivar**: Indica si el enemigo debe ser esquivado o el objetivo es colisionar con él.
- **Tiempo destrucción**: El tiempo que se está ejecutando la animación de morir.
- **Da puntos**: Indica si al chocar con el enemigo se logran puntos o no.
- **Puntos**: En caso de dar puntos, se tiene que indicar cuántos puntos da al chocar con él.

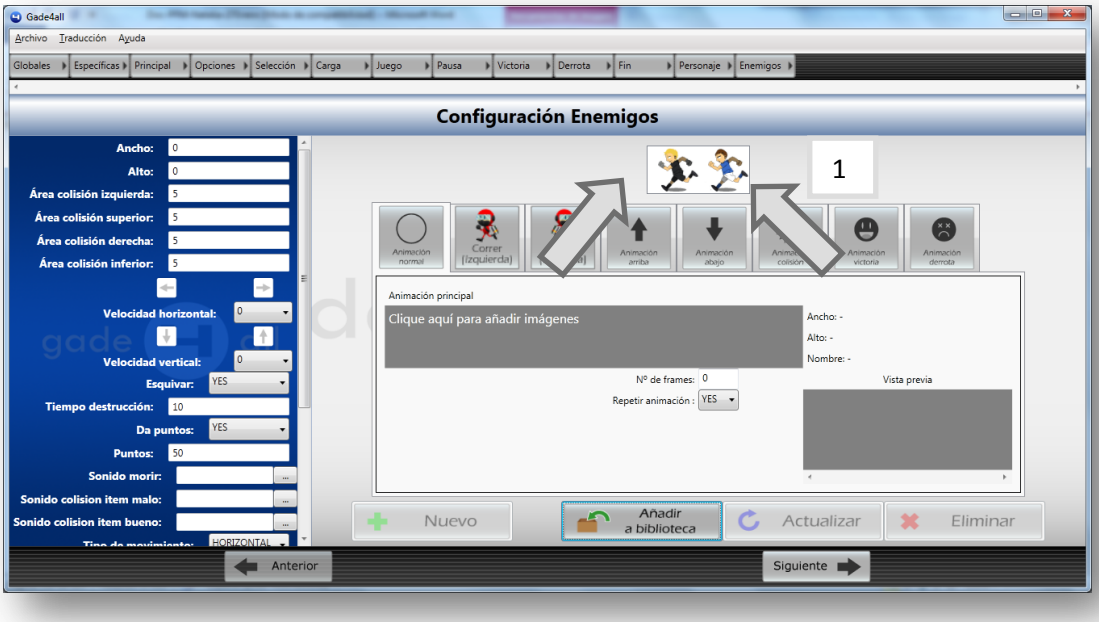


Ilustración 9.56. Pantalla enemigo configurada

En este caso se han creado dos enemigos, tal y como se muestra en la lista de miniaturas (1).

9.3.2.19 Pantalla Meta

Si en la pantalla de configuración de propiedades específicas del juego se seleccionó que el juego tendrá meta se deberá configurar en esta pantalla, si no se escogió esa opción bastará con pulsar sobre el botón "*Siguiente*" para continuar el proceso de creación del juego.

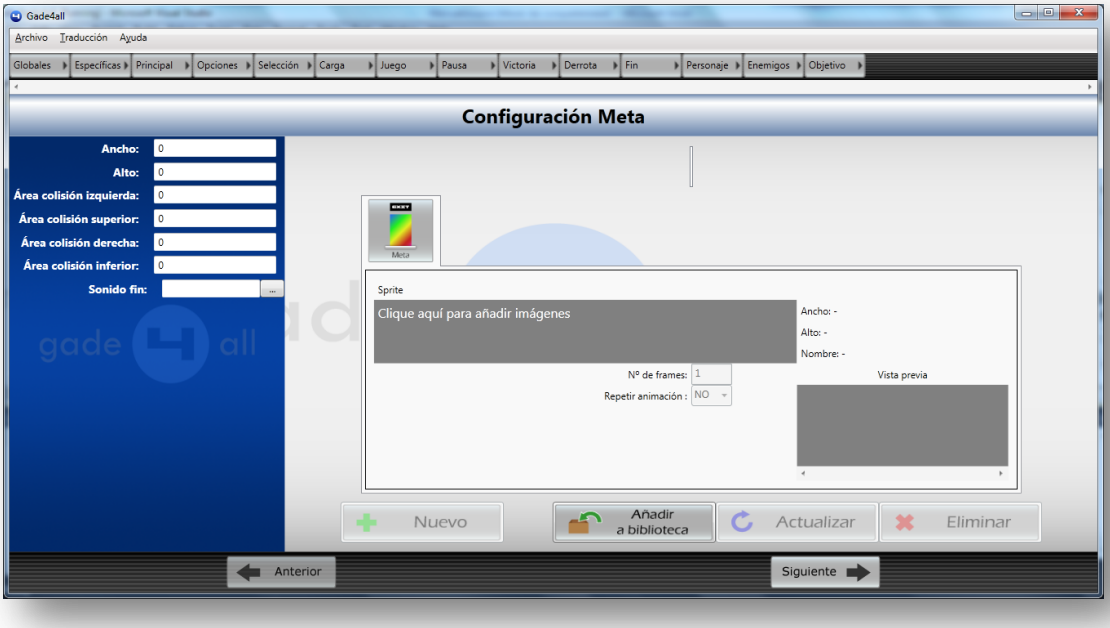


Ilustración 9.57. Configuración meta

En el caso de la configuración de la meta solamente se requiere una animación, y está tendrá solamente un frame y no se repetirá, se ejecutará solamente una vez. Se podrán crear varios.

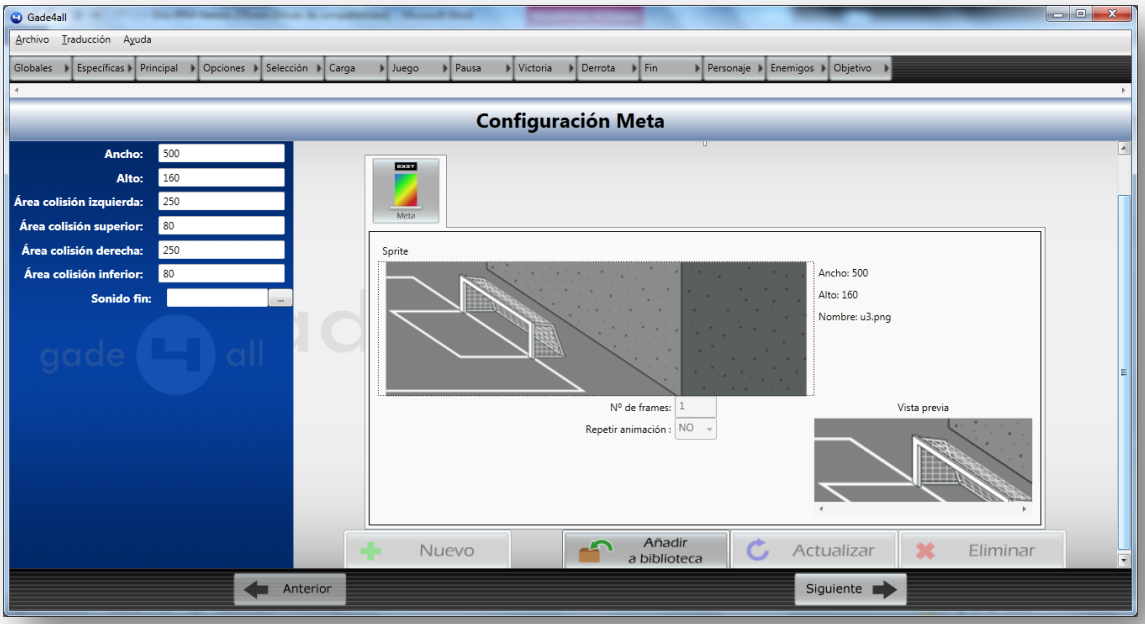


Ilustración 9.58. Pantalla meta configurada

9.3.2.20 Pantalla Escenario

En esta pantalla se realizará la configuración de los diferentes niveles del juego.

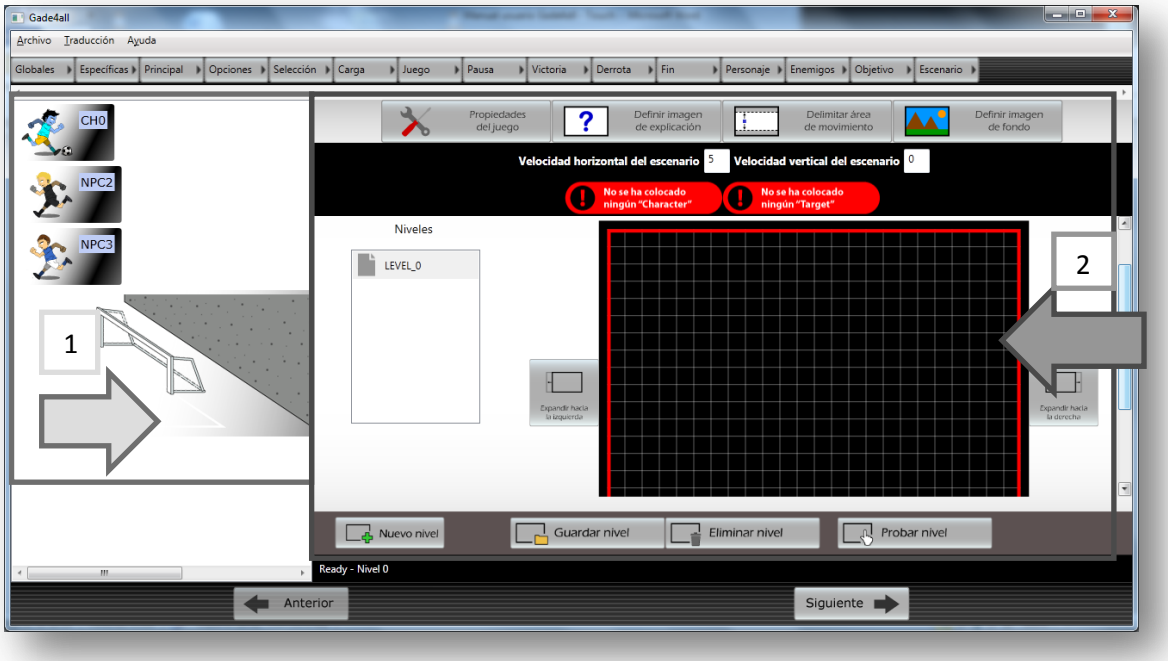


Ilustración 9.59. Configuración niveles del juego

Se divide en dos áreas. La primera (1) contiene los objetos creados en las pantallas anteriores, necesarios para la creación del nivel: Personaje principal, enemigos y meta. Para llevarlos al escenario bastaría con hacer clic sobre uno de ellos y luego hacer clic sobre el escenario para dibujarlo.

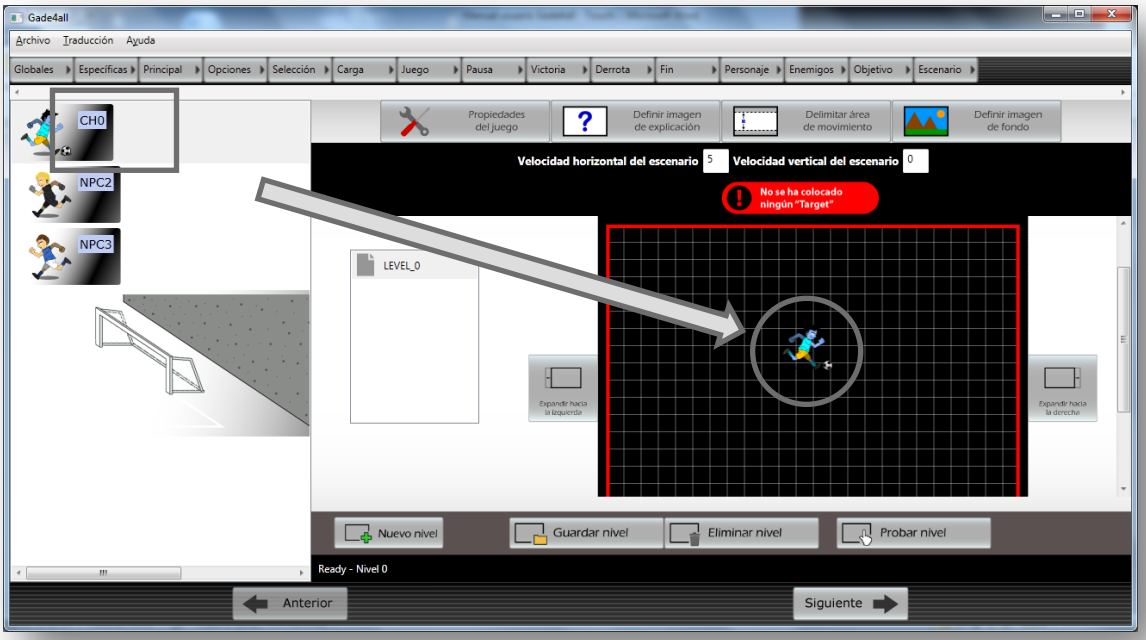


Ilustración 9.60. Dibujar elemento en escenario

El segundo área (2), es donde se van a crear los niveles, para ello hay que seguir varios pasos, los cuales se detallan a continuación.

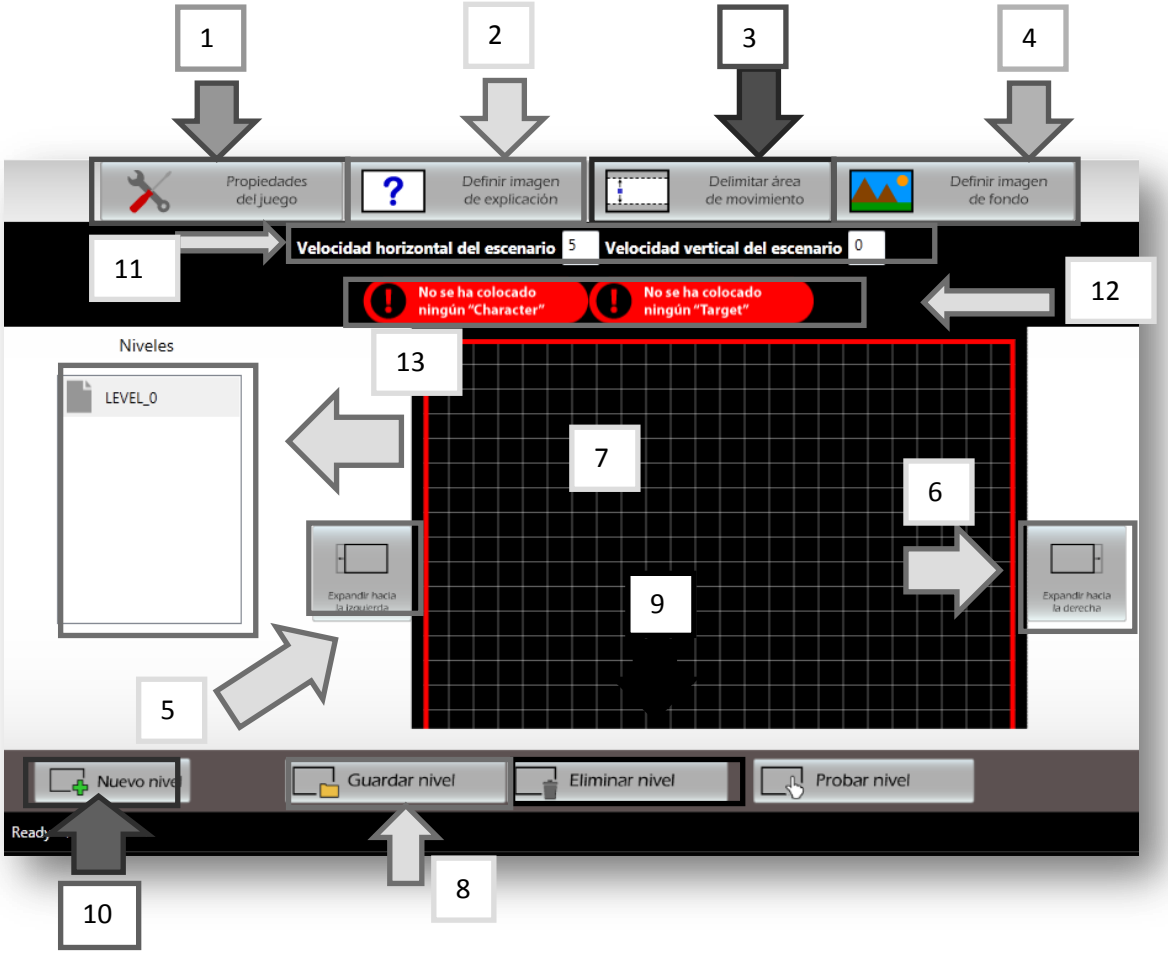


Ilustración 9.61. Pasos creación nivel

El primer paso (1) consiste en configurar las propiedades del nivel. Algunas propiedades tienen un valor por defecto.

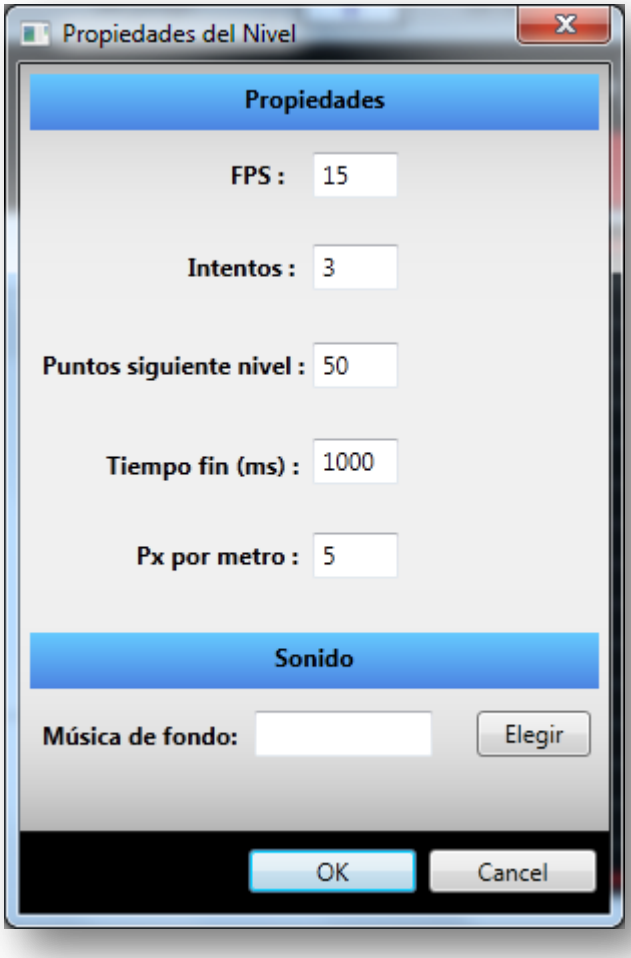


Ilustración 9.62. Propiedades del nivel

- **FPS**: frames por segundo. Velocidad a la que se ejecutarán las animaciones. El valor que aparece cargado es el que el usuario escogió en la primera pantalla del editor.
- **Intentos**: número de vidas disponibles en el nivel. Sólo se aplicará si el juego tiene vidas.
- **Puntos siguiente nivel**: puntuación que si es alcanzada por el usuario en el nivel, hace pasar automáticamente al siguiente.
- **Tiempo fin**: tiempo disponible para terminar el nivel.
- **Px por metro**: píxeles a los que corresponde un metro. Se utiliza cuando se contabilizan los metros avanzados en el juego.
- **Música de fondo**: música que sonará en el nivel.

Una vez configuradas las propiedades, se pulsa sobre el botón "*Ok*" para finalizar el proceso.

El siguiente paso (2), consiste en la elección de la imagen de descripción del nivel y la miniatura del nivel. Al pulsar sobre el botón para añadir una imagen, en ambos casos se abrirá un cuadro de diálogo para seleccionar la imagen. Una vez elegidas las imágenes, se pulsa sobre el botón "*Ok*" para finalizar el proceso.

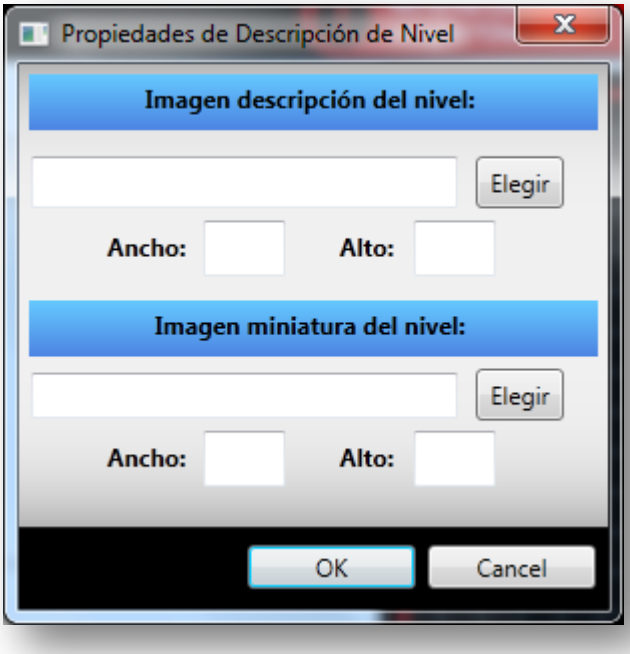


Ilustración 9.63. Propiedades descripción del nivel

Ilustración 9.64. Descripción de nivel

Ilustración 9.65. Miniatura del nivel

En el tercer paso (3) se definirán los delimitadores, es decir, los topes o niveles por los que el personaje principal se moverá por la pantalla. En este caso como la orientación del juego es horizontal los delimitadores por defecto son 1 y 320. Si fuese vertical serían 1 y 480. Siempre tiene que haber dos delimitadores.

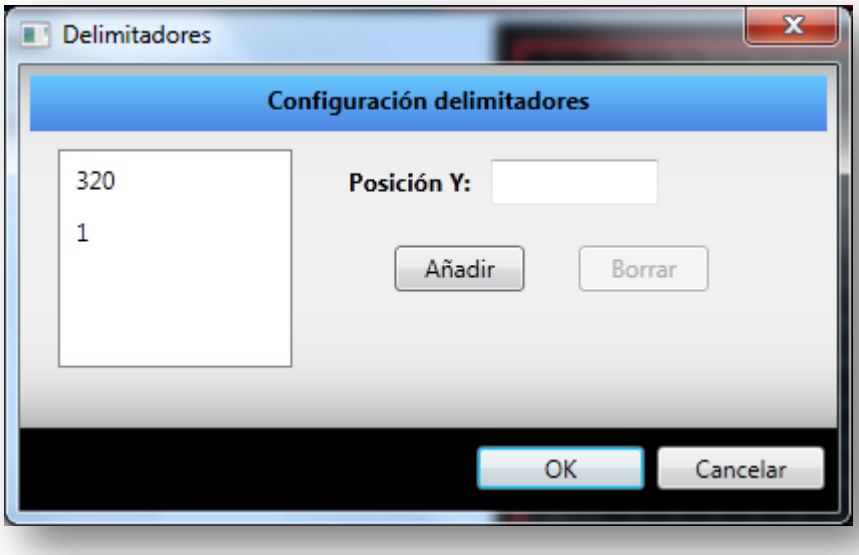


Ilustración 9.66. Delimitadores del nivel

Para añadir más solo habría que introducir el número deseado en el cuadro de texto "*Posición Y*" y darle al botón "*Añadir*". Para modificar uno ya existe, se hace clic sobre él y se modifica su valor. Una vez configurados los delimitadores, se pulsa sobre el botón "*Ok*" para finalizar el proceso.

El siguiente paso a realizar (4) es la configuración de los fondos del nivel. Hay dos opciones de configuración:

- **Fondo scrolling:** solamente se tiene que escoger una imagen de fondo. Se tiene que definir su velocidad en el eje X o Y.
- **Fondo parallax:** Este tipo de fondo puede estar compuesto por tres fondos superpuestos, así se daría más sensación del movimiento al fondo. También se tiene que configurar la velocidad de esos fondos.

Si se asigna velocidad 0 a los fondos estos no se moverán, permanecerán fijos en la pantalla.

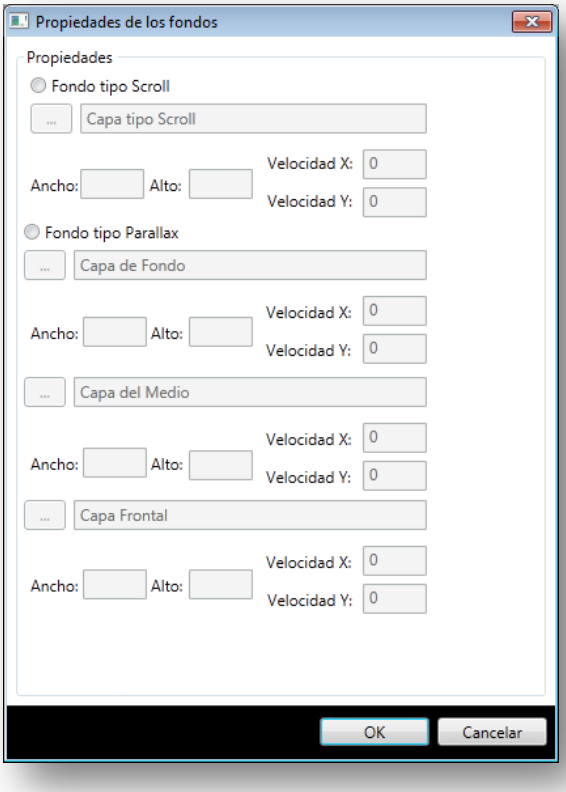


Ilustración 9.67. Propiedades del fondo

Para definir la velocidad de los fondos, se muestra la velocidad del escenario, en el propio escenario, a modo de orientación para el usuario.

Ilustración 9.68. Velocidad del escenario

Una vez configurado el fondo o fondos del nivel, se pulsará sobre "OK" para guardar los cambios realizados.

Ahora que están configuradas todas las propiedades del nivel, se va a proceder a diseñar el mismo.

Para situar los elementos en el escenario bastará con hacer clic sobre el personaje que se quiere añadir al nivel y hacer clic en el escenario, y este quedará colocado en el nivel. Para borrarlo, se deberá hacer clic con el botón derecho sobre él.

Para crear un nivel es obligatorio añadir un personaje principal y una meta.

Ilustración 9.69. Avisos escenario.

Según se vayan colocando alguno de esos dos elementos, se quitará el aviso correspondiente.

Ilustración 9.70. Eliminación avisos escenario.

El escenario viene delimitado por un rectángulo rojo que indica en todo momento cual sería el tamaño de la pantalla del dispositivo móvil. La esquina superior izquierda del rectángulo marca la coordenada (0,0) de la pantalla. Se puede aumentar el tamaño del escenario donde se van a colocar los elementos, por la derecha o por la izquierda, pulsando sobre los respectivos botones. En caso de hacerlo por la izquierda, los elementos que se sitúen a la izquierda del cuadrado rojo estarán ubicados en coordenadas del eje X negativas. Y si se sitúan a la derecha del cuadrado rojo se colocarán fuera de la pantalla, así según avancen con la velocidad que se le haya asignado a cada uno, entrarán a la pantalla de juego transcurrido un cierto tiempo, diferente para cada uno dependiendo de su velocidad.

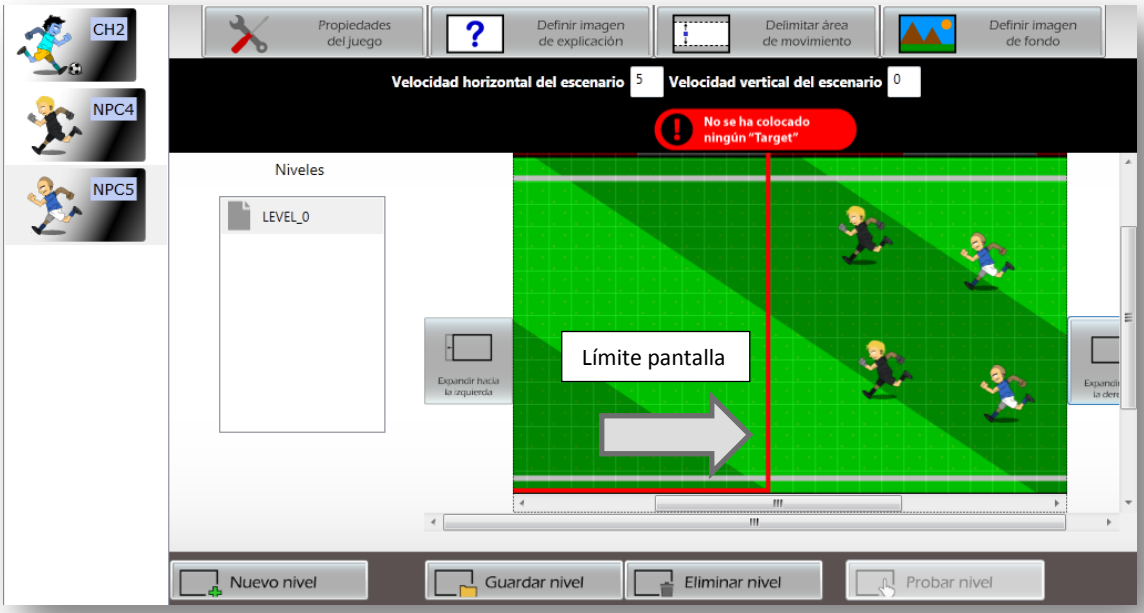


Ilustración 9.71. Coordenadas fuera pantalla.

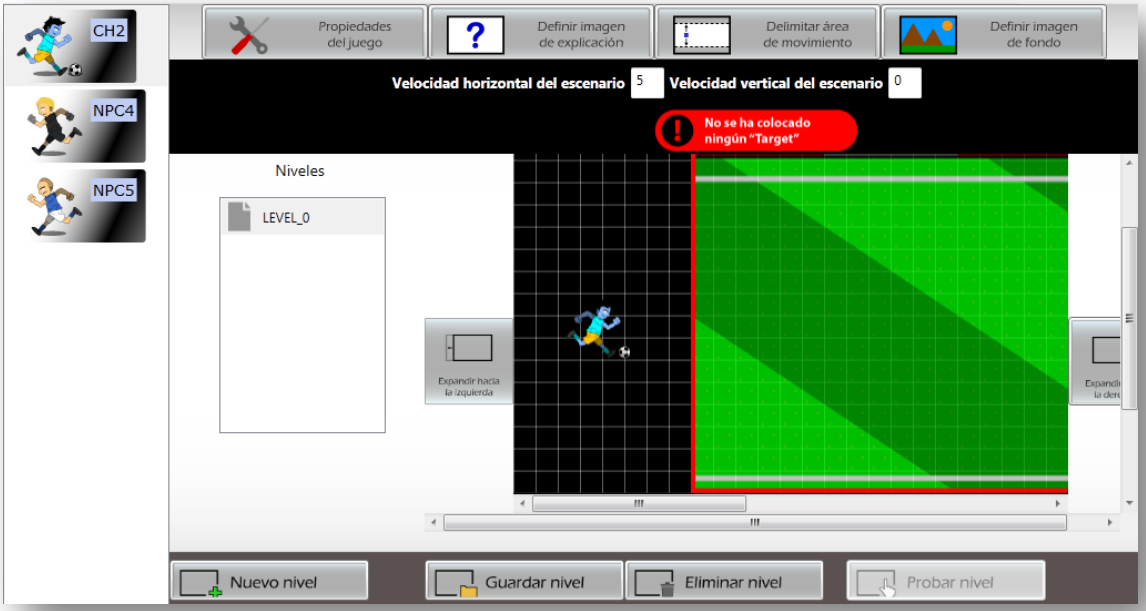


Ilustración 9.72. Coordenadas negativas

Si la orientación fuese vertical, el escenario tendría este aspecto:

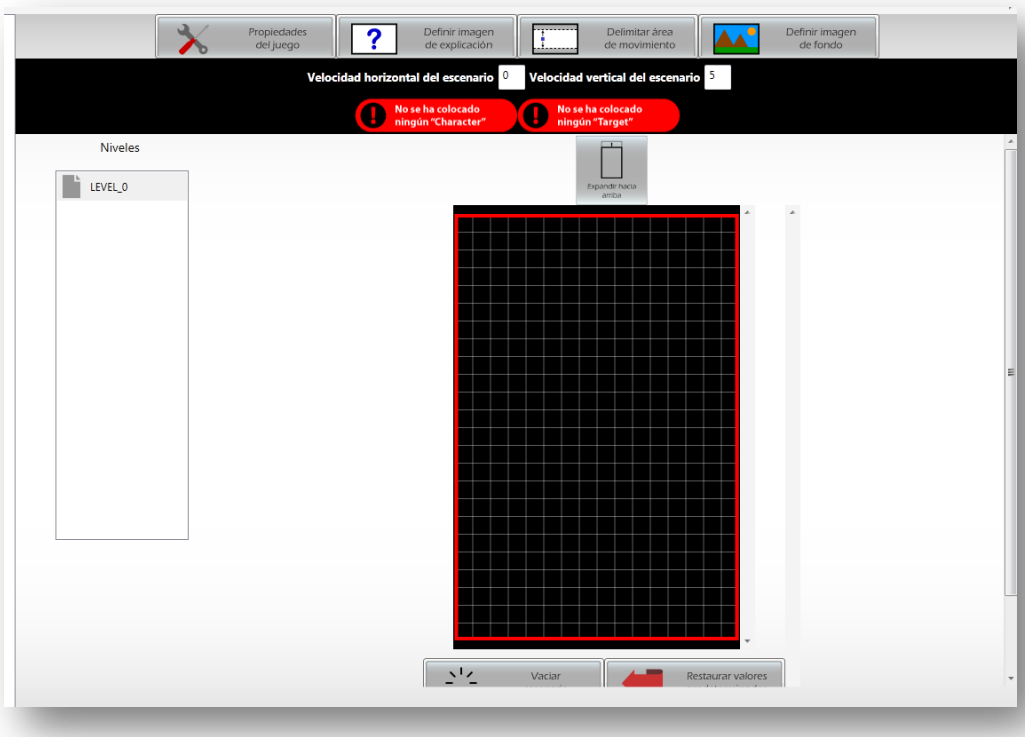


Ilustración 9.73. Escenario vertical

Una vez finalizado el diseño del nivel se pulsaría sobre el botón "Guardar Nivel" para salvar el nivel. Para diseñar otro nuevo nivel se pulsaría sobre "Nuevo Nivel" y se repetirían los pasos anteriores. Si se desea eliminar, se pulsaría sobre "Eliminar nivel".

Ilustración 9.74. Opciones niveles

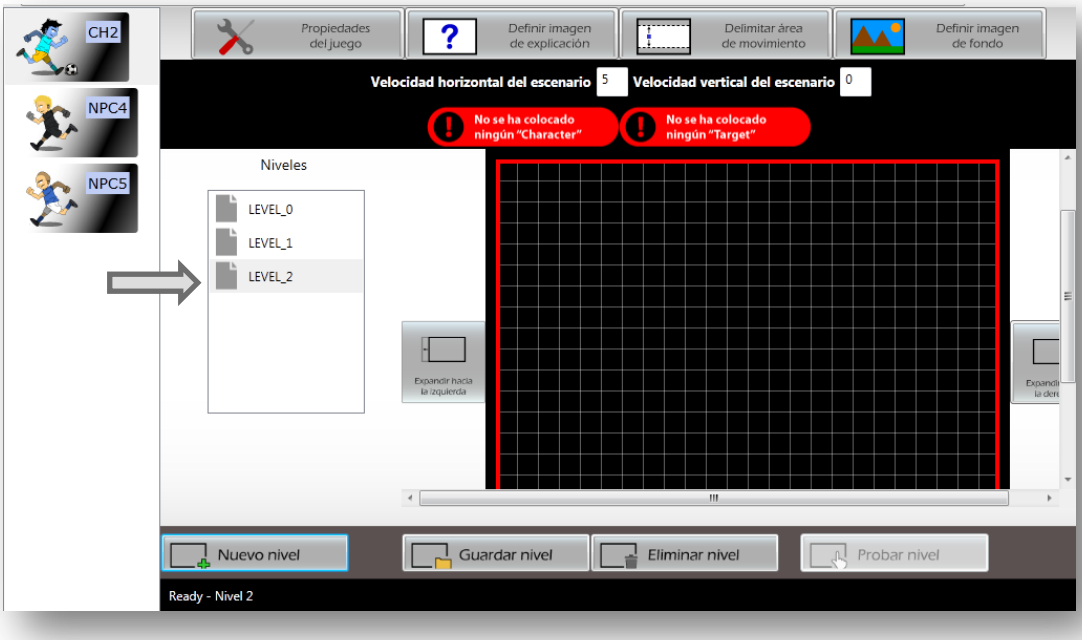


Ilustración 9.75. Cargar niveles

Según se van creando los niveles, se van almacenando en una lista de niveles que aparece a la izquierda del escenario. De esta forma, si se desea volver a un nivel creado anteriormente bastaría con hacer clic sobre el nombre del nivel que se desea cargar y se cargará su contenido en escenario.

En la parte inferior del escenario hay dos botones: uno para poder limpiarlo (1), es decir, borrar todos los elementos que se hayan añadido y otro para restablecer los valores predeterminados (2), como la velocidad del escenario y el tamaño del mismo.

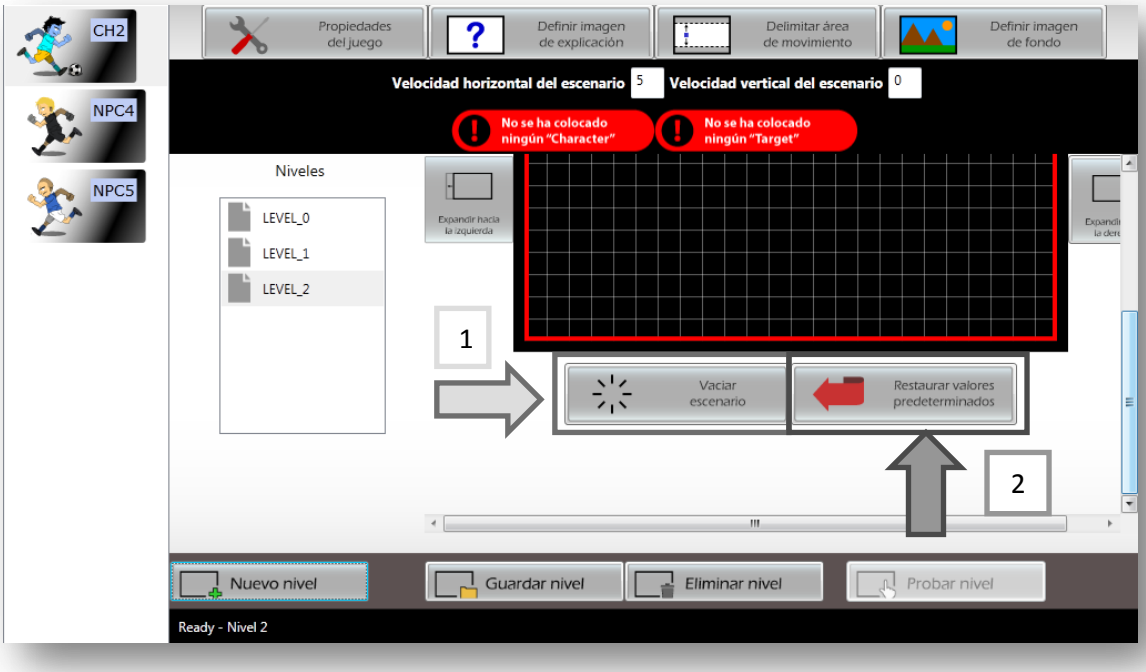


Ilustración 9.76. Botones vaciar y restaurar escenario

Al hacer clic en "*Siguiente*" se construirían los ficheros .xml de los niveles y se almacenarían en disco.

9.3.2.21 Pantalla selección ruta generación

Una vez finalizado el proceso de diseño del videojuego, el siguiente paso es la generación del mismo. En esta pantalla se escogerá la ruta donde se almacenará el juego generado.

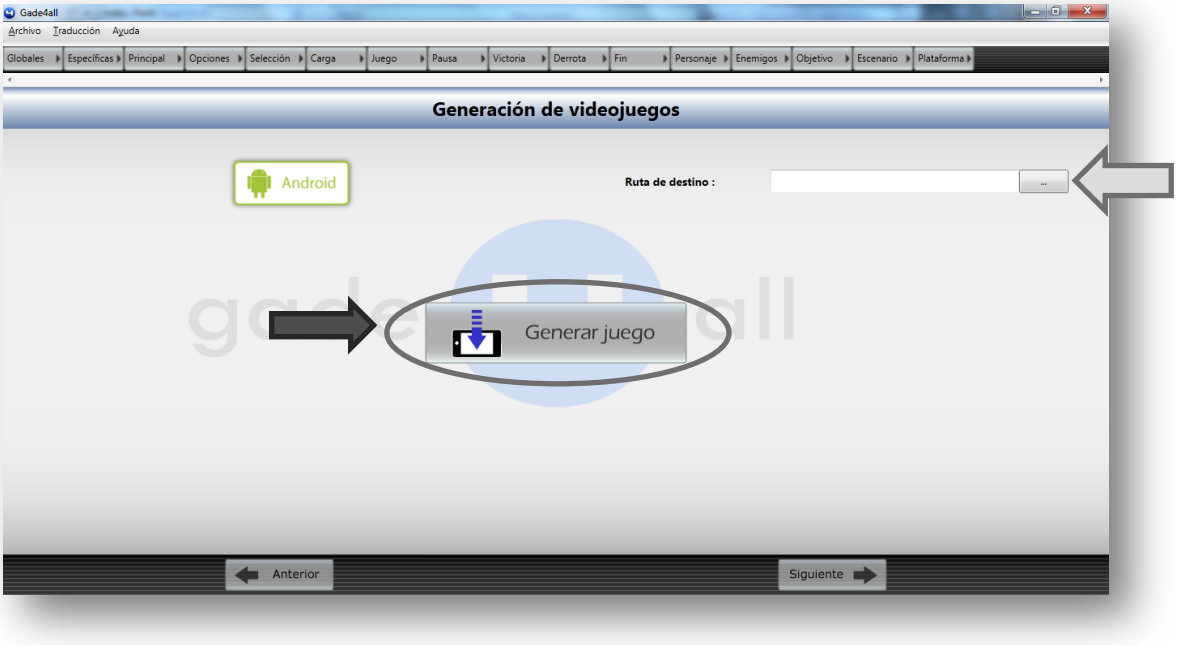


Ilustración 9.77. Configuración pantalla plataforma videojuego

Haciendo clic en el botón señalado por las flechas, se abrirá un cuadro de dialogo para seleccionar la carpeta donde se va a guardar el videojuego. Una vez seleccionada la ruta de destino, se deberá de hacer clic sobre el botón "*Generar Juego*", para comenzar el proceso de generación del juego.

Mientras se está generando el juego se mostrará al usuario el siguiente mensaje:

Ilustración 9.78. Imagen espera generación

Si se realizó el proceso correctamente se mostrará un mensaje de aviso para informar al usuario.

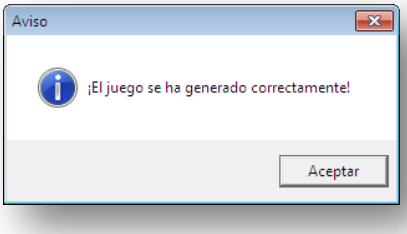


Ilustración 9.79. Juego generado correctamente

El siguiente paso a realizar es ejecutar el proyecto generado para obtener el archivo de juego que se instalará en el dispositivo móvil.

9.3.3 Compilar juego

El proyecto generado es un proyecto de Eclipse, que deberá ser importado en el Eclipse, bastará con ejecutarlo y se podrá jugar desde el emulador o bien desplegarlo directamente en el móvil, y ya generará la .apk para poder, finalmente, jugar al juego creado con la herramienta.

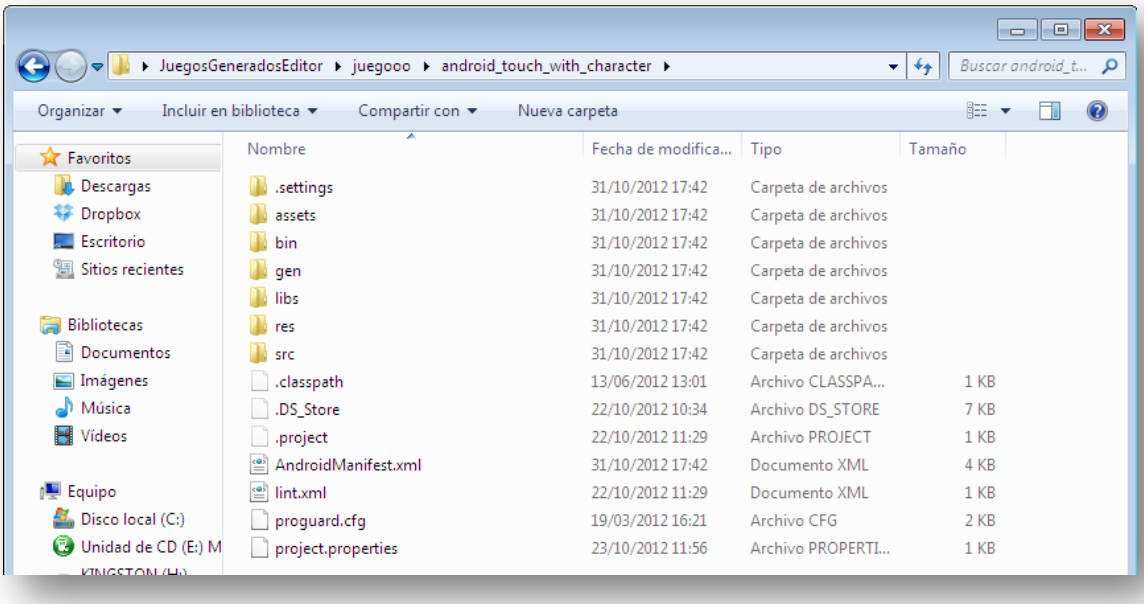


Ilustración 9.80. Proyecto Eclipse generado para Android

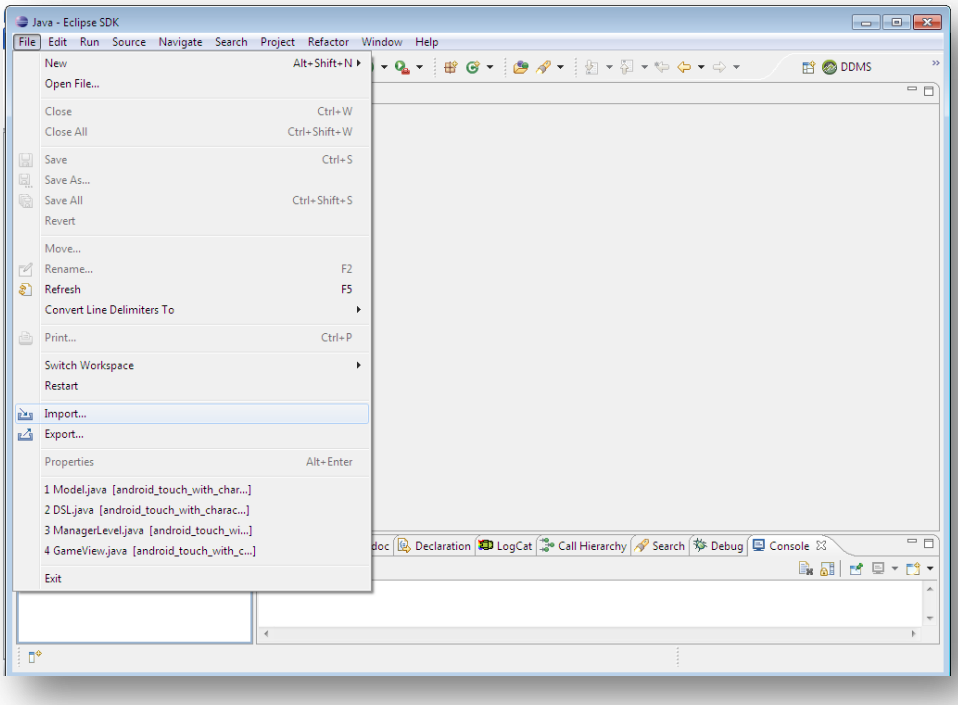


Ilustración 9.81. Importar proyecto Eclipse

Al seleccionar "Import" se abrirá un cuadro de diálogo en que se deberá escoger la ruta donde se almacenó el proyecto Eclipse generado por la herramienta.

Una vez importado el proyecto se podrá limpiar este antes de compilarlo. Para ello: Proyect → Clean y se abrirá una ventana donde se deberá seleccionar el proyecto a limpiar.

Para ejecutarlo se hará clic con el botón derecho sobre el proyecto, se escogerá la opción "Run As" y se seleccionará "Android Application". Si el dispositivo móvil se encuentra conectado al PC y Eclipse lo ha detectado, se ejecutará directamente en el móvil. Si no, se desplegará en el emulador.

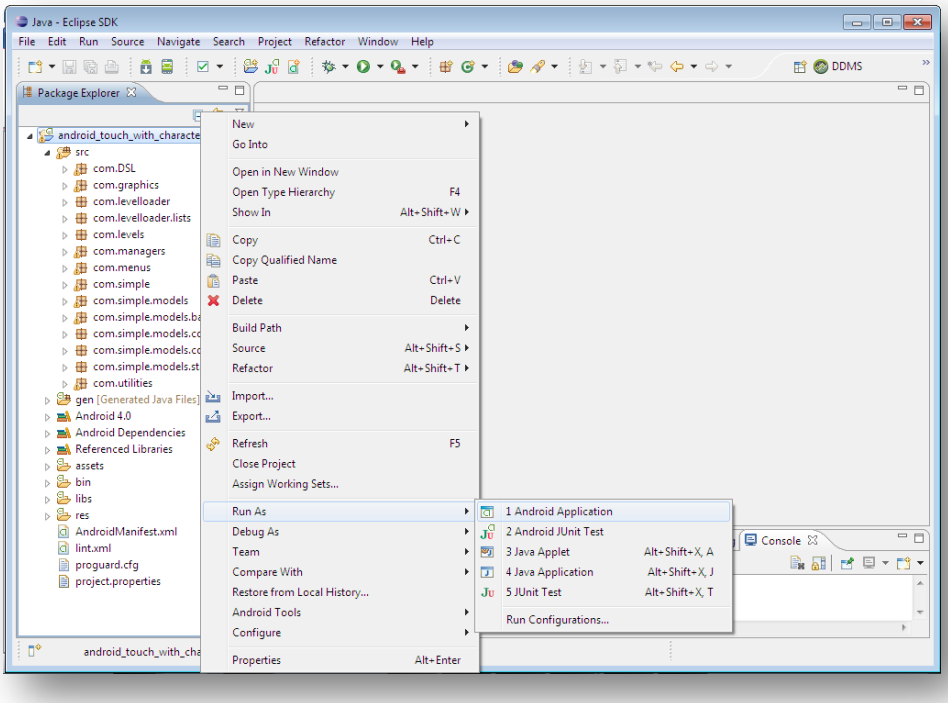


Ilustración 9.82. Ejecutar proyecto Eclipse

Ilustración 9.83. Juego desplegado en emulador Android

9.4 Manual del Programador

En este manual se va a describir el procedimiento a seguir para realizar algunos cambios en la herramienta. Hay que tener en cuenta que los cambios que se hagan en la interfaz (añadir nuevas pantallas, elementos a las pantallas, propiedades de los elementos) deben aplicarse posteriormente también en la instancia del DSL y en la plantilla. Si solamente son cambios de aspecto gráfico no será necesario.

Se va a explicar cada parte de la herramienta por separado. Solamente se va a profundizar en algunos cambios determinados.

9.4.1 Interfaz

En esta se pueden realizar varios cambios, tanto de diseño como de funcionalidad.

9.4.1.1 Añadir nuevas funcionalidades al menú

En la herramienta desarrollada para este proyecto fin de máster, en el menú de opciones solamente están disponibles las opciones "*Guardar*" y "*Salir*". Si se desease añadir nuevas opciones, bastaría con añadirlas a la ventana contenedora, donde se abren todas las pantallas, y dar funcionalidad a esas opciones nuevas. Estos cambios se realizarían sobre la clase "*BaseWindow*".

9.4.1.2 Añadir nuevas pantallas

Si el usuario programador desease añadir nuevas pantallas a la interfaz, como por ejemplo nuevas pantallas a los menús del videojuego o bien añadir otro tipo de personajes al juego, como disparos, tendría que incluir esas nuevas pantallas en el flujo de pantallas existente, en el que están todas las pantallas que se muestran en la interfaz en el orden en el que se encuentran en el flujo. También se podría variar el orden de aparición de las pantallas. Para llevar a cabo estos cambios se tendría que modificar la clase "*BaseWindow*".

Al añadir nuevas pantallas habría que modificar la instancia del DSL añadiendo las etiquetas correspondientes a esa pantalla nueva, y también habría que modificar la plantilla, creando las clases necesarias para definirla.

9.4.1.3 Modificar aspecto gráfico

Si se desease modificar el aspecto gráfico de alguna de las pantallas, como cambiar los colores, cambiar la disposición de los elementos, etc., se tendría que modificar el archivo con extensión ".xaml" de la pantalla correspondiente. Y si se añadiesen imágenes nuevas que luego se almacenen para ser utilizadas en la plantilla, se tendrían que guardar en la carpeta "Images". Y si fuesen imágenes que solamente se van a utilizar en el aspecto gráfico de la interfaz, se deberán copiar a la carpeta "ImagesEditor".

9.4.1.4 Añadir botones nuevos a las pantallas de los menús del videojuego

Cada pantalla de los menús del videojuego contiene un número determinado de botones, cada uno con una funcionalidad diferente. Si se desease añadir botones nuevos a cada pantalla, habría que añadir un nuevo control (*ControlButtonLabel*) al archivo con extensión "*.cs*" de la pantalla correspondiente y configurar los valores (imagen de fondo, tamaño, posición) de ese nuevo botón. También habría que añadir a la instancia del DSL las etiquetas correspondientes a las propiedades de cada botón añadido. Y finalmente, habría que crear también ese botón en la plantilla, añadiéndolo a la vista correspondiente de la pantalla del menú donde se quiera incluir el nuevo botón.

9.4.1.5 Modificar propiedades

Si el usuario programador quisiera añadir nuevas propiedades a los personajes del juego o bien a algún elemento de los menús del juego, debería aplicar esos cambios en el archivo con extensión "*.xaml*", si se desea mostrar esas propiedades en la interfaz, y también en él "*.cs*", para guardar el valor de esas nuevas propiedades y luego escribirlos en la instancia del DSL. Al añadir nuevas propiedades será necesario incluirlas también en la plantilla. Asi que, si por ejemplo, se añaden nuevas propiedades al personaje principal, en la plantilla se tendría que modificar el modelo correspondiente al personaje y agregar las nuevas propiedades.

9.4.1.6 Añadir nuevas plataformas de despliegue

Tal y como están diseñados la interfaz y la instancia del DSL, se permite añadir nuevas plataformas de despliegue aplicando pocos cambios. En la interfaz se tendría que añadir en la pantalla de propiedades globales (*PageGlobals.xaml*) la opción de añadir los ID's de la publicidad necesarios para las nuevas plataformas. Y en la pantalla de selección de plataforma, añadir la opción de elegir la ruta de despliegue para las nuevas.

En la instancia del DSL habría que añadir etiquetas para los ID's de la publicidad. Y sería necesario crear una nueva plantilla para cada plataforma añadida, desarrollada en el lenguaje de la plataforma.

Las opciones de plataformas nuevas serían:

- iOS
- HTML 5
- WindowsPhone

9.4.1.7 Añadir nuevas tipologías de juegos

En el caso de este trabajo fin de máster solamente se ha desarrollado una tipología de juego, Acción, pero se podrían incluir otras tipologías. Habría que añadirlas a la pantalla principal "*MainWindow*". Se tendría que crear un flujo de ventanas (en la clase "*BaseWindow*") para la nueva tipología y diseñar pantallas nuevas si fuese necesario, porque algunas de las pantallas se podrían reutilizar.

También habría que estudiar qué nuevas propiedades se tendría que añadir a la instancia del DSL y se tendría que crear la plantilla para el nuevo tipo de juego.

9.4.2 Analizador

En caso de añadir nuevas plataformas de despliegue, habría que crear un nuevo analizador para cada plataforma, ya que las plantillas serán diferentes, tendrán otra jerarquía, y por tanto el analizador tendría que buscar los archivos en carpetas que se encuentran en diferente ubicación dentro de la plantilla.

9.4.3 Instancia del DSL

Como se ha visto en la fase del diseño de la instancia del DSL, cada elemento del juego tiene sus correspondientes etiquetas dentro del fichero. Si se desease añadir nuevos elementos o nuevas propiedades, habría que crear nuevas etiquetas para ellos y darle un valor. Cada elemento tiene que tener una etiqueta de apertura y otra de cierre.

Para que estos cambios se reflejasen en la interfaz y en la plantilla, habría que modificar ambas para que se adaptasen a los cambios introducidos.

9.4.4 Plantilla Android

Como se describió en la fase del diseño de la plantilla, esta tiene una estructura de carpetas determinada, y cada clase va en su correspondiente carpeta.

9.4.4.1 Añadir personajes nuevos

Si se desease añadir nuevos personajes al videojuego, se tendría que crear una nueva clase, que heredaría de modelo, para cada tipo de personaje nuevo. Luego habría que pintarlo en la pantalla del juego, y para ello se tendría que añadir a la clase "GameView", que es la contiene la lógica del juego, y si este se moviera se tendría que incluir también una llamada al método que ejecutase el movimiento de ese personaje.

En el cargador de niveles, habría que guardar en variables los valores leídos de la instancia del DSL de ese personaje, para luego asignarle esos valores.

9.4.4.2 Añadir botones nuevos

El usuario programador podría querer añadir botones nuevos a las pantallas de los menús del videojuego. Para ello, tendría que añadir un nuevo control (*ControlButton*) a la vista que se corresponde con la pantalla en la que se quiere añadir el nuevo elemento. Se tendría que pintar en la misma y darle funcionalidad, programando el evento que se lanzará cuando se pulse sobre ese botón.

Para poder asignarle valores a las propiedades del nuevo botón, habría que modificar la clase "DSL.java" y asignar a las nuevas propiedades la correspondiente etiqueta de la instancia del DSL:

```
int  newElementSize = %new_element_size%;
```

9.4.4.3 Añadir pantallas nuevas

Si se desease ampliar el videojuego añadiéndole pantallas nuevas, habría que crear una actividad (*Activity*) y una nueva vista (*View*) para cada pantalla, y habría que definir los botones y fondos que tendría esa pantalla. Al crear una nueva actividad, habrá que incluirla en el fichero de la plantilla "*AndroidManifest*.xml".

Y habría que asignarle a algún botón u otro elemento del videojuego, la funcionalidad necesaria para que lanzase esta nueva pantalla.

Capítulo 10. Conclusiones y Ampliaciones

10.1 Conclusiones

Con el desarrollo de la herramienta presentada en este trabajo fin de máster hemos tratado de cubrir un hueco importante existente en la actualidad en el campo del diseño y creación de videojuegos, buscando poder facilitar el acceso a los mismos de usuarios inexpertos en la materia.

La fase inicial de este proyecto consistió en desarrollar dos videojuegos de forma convencional, con las herramientas profesionales al uso. Esta etapa se realizó de forma paralela con una fase de análisis, en la que se buscaron los elementos comunes entre los dos juegos que se estaban diseñando y otros videojuegos ya existentes en el mercado de la misma tipología.

El desarrollo de estos prototipos proporcionó los conocimientos necesarios sobre el funcionamiento de un videojuego. Además, se obtuvo una idea clara de qué partes se debían automatizar y ello dio lugar a la aparición de los primeros elementos propuestos para la instancia del DSL, los cuales fueron concebidos en base al tipo de comportamientos y aspectos que pudieran modificarse en el juego para obtener varios diferentes a partir de un modelo base. Se definieron pues los aspectos comunes entre distintos juegos, como las pantallas de los menús, en las que se definió un patrón que permitiera agregar elementos de forma sencilla.

El siguiente paso fue convertir estos videojuegos base en una única plantilla, cuya funcionalidad no fuera adquirida directamente desde el lenguaje de programación, sino más bien que modificara su comportamiento y apariencia física en base a lo definido en la instancia del DSL.

Dicha plantilla se alimenta de ficheros XML (uno para los menús y uno por cada nivel) en los que se detallan todos los elementos que definen el juego: aspectos gráficos, sonidos, reglas del juego, etc.

Una vez construida la plantilla se desarrolló un analizador que permitiera hacer la sustitución de los elementos definidos en la instancia del DSL en la plantilla de forma automática, así como de los recursos gráficos utilizados.

Para finalizar, se llevó a cabo la fase más compleja de desarrollo de este proyecto, la creación y diseño de una interfaz gráfica, que permitiera automatizar todo el proceso de creación de videojuego, permitiendo que este se realizase de forma sencilla y en pocos pasos.

Lo resultados obtenidos con la herramienta desarrollada han sido satisfactorios: se ha demostrado la potencia de la implementación de una instancia del DSL, que ha permitido crear

una abstracción de alto nivel, y además, la aplicación ha tenido ya una experiencia real en los videojuegos desarrollados a propósito de la Eurocopa 2012, que aquí se han explicado, y que se encuentran a disposición de los usuarios en el market de Android.

Se ha logrado reducir el tiempo de creación de los videojuegos notablemente, tal y como se muestra en la siguiente tabla, en la que se compara el tiempo invertido en el desarrollo de los videojuegos base realizados en este proyecto, y uno creado con la herramienta desarrollada. Se excluye el tiempo invertido en el diseño de los recursos gráficos.

	Videojuego base 1	**Videojuego base 2**	**Videojuego desarrollado con la herramienta**
Tiempos	90 horas	90 horas	5 horas

En definitiva, se han cumplido de una forma satisfactoria todos los objetivos planteados al inicio de este Trabajo Fin de Máster.

10.2 Ampliaciones

10.2.1 Mejorar proceso generación videojuego

El proceso de generación del videojuego no requiere que el usuario que lo haga tenga conocimientos de programación, pero si se necesita abrir el Eclipse e importar el proyecto obtenido con la herramienta. Este proceso se detalla al completo en el Manual de Usuario, pero una futura ampliación y que reduciría el proceso de generación del videojuego, sería poder obtener directamente el videojuego listo para instalar en el dispositivo móvil desde la herramienta, sin tener que abrir ningún otro programa.

10.2.2 Nuevas funcionalidades

Este proyecto presenta un gran potencial para la creación de gran variedad de videojuegos, sin embargo la cantidad de videojuegos podría verse incrementada al incluir más funcionalidades predefinidas. La mayor aportación que se podría hacer al respecto es, incluir nuevos comportamientos para el personaje principal y los enemigos, los cuales pudieran aumentar las posibilidades de reglas de juego.

Otra nueva funcionalidad que podría incorporarse sería crear videojuegos sin personaje principal, que solamente se tuviera que interactuar con los enemigos.

10.2.3 Biblioteca de juegos

Actualmente en la herramienta solamente hay dos opciones al comenzar el proceso:

- Crear un videojuego desde cero.
- Cargar uno guardado previamente.

Pero para facilitar más aún el proceso de creación de videojuegos con la herramienta, se podría incluir una biblioteca de videojuegos ya creados. Estos videojuegos de la biblioteca ya tendrían configurados los menús, el personaje principal, los enemigos, la meta y un nivel de ejemplo. De esta manera el usuario de la herramienta podría crear sus videojuegos partiendo de los de la biblioteca, y con solo añadir personajes nuevos o bien creando más niveles ya tendría listo el videojuego.

10.2.4 Mejorar funcionalidad editor de niveles

Como resultado de las pruebas de usabilidad realizadas para este proyecto, la mayor dificultad que tuvieron los usuarios fue en la creación de los niveles.

Por tanto, una posible solución sería reducir el número de acciones a realizar o bien dividir las tareas en dos pantallas, configurando en una las propiedades del nivel, y en otra definir el nivel.

También se podría añadir una pre-visualización del nivel, ya que actualmente hay que compilar e instalar el videojuego para ver el funcionamiento de los niveles. Se podría incluir una vista previa del nivel, mostrando este al completo y así el usuario podría ver cómo está quedando el nivel que está diseñando.

10.2.5 Ampliación a videojuegos 3D

En este proyecto se ha realizado el desarrollo para videojuego de acción 2D pero ampliar a videojuegos en 3D (tres dimensiones), es decir, añadirle el componente Z en el eje de coordenadas, conllevaría escenarios con más realismo y más vida.

Capítulo 11. Aplicaciones reales del proyecto

11.1 Videojuegos base

Como ya se ha mencionado a lo largo de la documentación, el primer paso que se ha llevado a cabo en la elaboración de este proyecto ha sido la creación de dos tipos de videojuegos base pertenecientes a la misma tipología, videojuegos de acción en 2D.

Con ello se ha fijado como objetivo que a partir de los videojuegos se identificasen las partes variables y fijas del código que permitieran crear un juego u otro, para después construir una plantilla dejando en blanco los valores de las partes variables, que se cubrirán con los datos obtenidos de la instancia del DSL.

En este caso se han creado dos juegos de la misma familia pero con diferente funcionalidad. A continuación se van a explicar los objetivos y funcionamiento de cada uno de ellos.

11.1.1.1 Juego 1 – Bichos

Ilustración 11.1 . Icono juego Bichos

11.1.1.1.1 Objetivos

El objetivo de este juego es destruir enemigos que se están moviendo por la pantalla tocándolos y evitando tocar las bombas, que harán que se pierda la partida. Para el juego que se ha construido en este caso la temática son "bichos": los enemigos a destruir, hormigas y las abejas, bombas.

El juego cuenta con tres niveles, cada uno de los cuales tiene un modo de juego diferente:

- En el primer nivel el objetivo es destruir el mayor número de hormigas posible para ganar puntos, evitando que salgan de la pantalla sin ser eliminadas, ya

que por cada una que se escape se pierde una vida. Las abejas eliminan todas las vidas de golpe y hacen perder la partida inmediatamente. Hay un tercer elemento, una mariquita en este caso, que permite conseguir vidas extra. Si se acaban las vidas disponibles se perdería la partida. En la parte superior de la pantalla se van mostrando los puntos y las vidas disponibles.

Ilustración 11.2 . Explicación nivel uno juego Bichos

Ilustración 11.3 . Nivel 1 juego Bichos

- En el segundo nivel el objetivo es destruir el mayor número posible de hormigas en el menor tiempo. El tiempo se va descontando según pasan los segundos. Si se termina el tiempo y no se han destruido todas las hormigas se pierde la partida, pero si, en cambio, se destruyen antes de que se termine el tiempo se pasa al siguiente nivel directamente. En este modo de juego no hay vidas y las hormigas pueden salirse de la pantalla sin quitar vidas. Las abejas restan puntos o hacen perder la partida directamente, o incluso podrían restar tiempo. Las mariquitas dan unos segundos extra que se sumarían al cronómetro.

Ilustración 11.4 . Explicación nivel dos juego Bichos

Ilustración 11.5 . Nivel 2 juego Bichos

- Y en el tercer nivel el objetivo es coger todas las hormigas que se pueda, mediante un caza-mariposas que controla el usuario con el desplazamiento del dedo por la pantalla, o bien esquivarlas. Con cada hormiga que se salga de la pantalla, se pierde una vida. Las abejas restan vidas o puntos y las mariquitas dan vidas extra. Si el objetivo es recoger las hormigas y se cazan todas, se supera el nivel y se termina el juego. En cambio si el objetivo es esquivarlas y se tocan se restan vidas, y al acabarlas, se pierde el nivel.

Ilustración 11.6 . Explicación nivel tres juego Bichos

Ilustración 11.7 . Nivel 3 juego Bichos

Como se puede observar, los diferentes personajes del juego (los "bichos") tienen diferentes comportamientos en cada nivel. Todos los elementos tienen las mismas propiedades ya que todos ellos son del mismo tipo de objeto, *Enemy*, pero con diferentes comportamientos dependiendo de los valores que tomen esas propiedades. De esta manera a partir de un solo enemigo se pueden obtener muchos más y ello permite crear muchos juegos diferentes.

Ilustración 11.8 .Enemigo con aspecto de vida extra

Ilustración 11.9 . Enemigo con aspecto de hormiga

Ilustración 11.10 . Enemigo con aspecto de abeja

Por ejemplo las hormigas pueden tener la misma imagen pero distintos comportamientos: unan pueden dar puntos, otras quitarlos o restar vidas. Esto permite generar aún más tipos de objetos.

El caza-mariposas, sin embargo, es un objeto de tipo *Character*, es decir, el personaje cuyo movimiento es controlado por el usuario. Este objeto también varias características configurables que permiten obtener muchos objetos diferentes.

Ilustración 11.11 . Imagen caza-mariposas

11.1.1.1.2 Funcionamiento

En la pantalla del menú principal del juego hay dos botones: "Jugar" y "Salir". Haciendo clic en el segundo de ellos se saldría del juego y haciéndolo sobre el primero se accedería a la pantalla de selección de nivel.

Ilustración 11.12 . Botón jugar juego Bichos

Ilustración 11.13 .Botón salir juego Bichos

Una vez situado el usuario en la pantalla de selección de nivel se podrá escoger uno de los diferentes niveles haciendo clic en las flechas para ver los que está disponibles, y al elegir uno se toca la imagen miniatura del nivel o bien el botón de "Empezar". Si es la primera vez que se juega solo se podrá escoger el nivel uno, y según se vayan superando los niveles se podrá ir accediendo a los siguientes.

Ilustración 11.14 . Selección de nivel juego Bichos

Ya seleccionado el nivel se muestra en pantalla una breve descripción del nivel y de sus objetivos. Pulsado sobre ésta se accede al juego.

En la parte superior de la pantalla hay una barra de estado en la que se muestran diferentes iconos y marcadores dependiendo del nivel: los puntos, las vidas disponibles, el tiempo restante para finalizar el nivel, los enemigos destruidos y un botón de pausa. Si se hace clic sobre este último botón se para el juego temporalmente y se abre una ventana de pausa en la que hay dos botones que permiten, o bien regresar al juego, o salir de este e ir a la pantalla de selección de nivel.

Ilustración 11.15 . Iconos de la barra de estado

Ilustración 11.16 . Pantalla de pausa juego Bichos

En cada nivel hay un juego de modo distinto, pero la manera de interactuar con los elementos del juego es la misma, para destruir enemigos u obtener puntos o vidas extra se toca con el dedo sobre él. Los enemigos pueden tener varios tipos de movimientos:

- De arriba abajo de la pantalla, a velocidad constante.
- De izquierda a derecha, a velocidad constante.
- Aleatorio: rebotan con las paredes sin salirse de la pantalla con movimientos diagonales.
- Gravedad vertical: son lanzados desde la parte inferior de la pantalla y cuando alcanzan una altura determinada vuelven a caer hacia abajo.
- Gravedad horizontal: salen del lado izquierdo de la pantalla y se desplazan hacia la derecha con una velocidad constante y al llegar a una coordenada determinada se precipitan hacia la parte inferior de la pantalla.

Y pueden tener varios comportamientos:

- Dar vidas extra o quitar vidas
- Dar puntos o quitarlos
- Quitar tiempo o añadir unos segundos al tiempo restante
- Tener que ser esquivados o tener que chocar contra ellos

En el caso de la existencia de un personaje principal controlado por el usuario, basta con poner el dedo sobre él y moverlo sin levantar el dedo de la pantalla, y así el objeto seguirá el movimiento marcado con el desplazamiento sobre la pantalla o bien irá directamente al punto de la pantalla que se haya tocado con el dedo.

Una vez superado el nivel se accede a una pantalla de victoria en la que se muestran los puntos totales obtenidos y los enemigos destruidos, además de tres botones que permitirán al usuario repetir el nivel, pasar al nivel siguiente (si este no es el último del juego) o volver a la pantalla de selección de nivel.

Ilustración 11.17 . Pantalla de victoria juego Bichos

En caso de no haber superado el nivel se muestra una pantalla de derrota en la que se permite, mediante dos botones, repetir el nivel o ir a la pantalla de selección de nivel.

En caso de superar el último nivel se muestra al usuario una pantalla de fin de juego, desde la cual podrá volver al menú de inicio del mismo.

11.1.1.2 Juego 2 – Herocopter

Ilustración 11.18 . Icono juego Herocopter

11.1.1.2.1 Objetivos

El objetivo de este juego es mover un personaje principal evitando los obstáculos que se desplazan por la pantalla en diferentes alturas. Para el juego que se ha desarrollado en este proyecto el objetivo es mover un helicóptero y conseguir que llegue al final del nivel recogiendo monedas y evitando las bombas.

Este juego cuenta también con tres niveles, cada uno de los cuales tiene un modo de juego distinto:

- En el nivel uno el objetivo es llegar al final recogiendo el mayor número posible de monedas, que darán puntos o vidas, sin perder las vidas disponibles. Si se colisiona con las bombas, se pierde una vida por cada una de ellas. Si se terminan las vidas se termina la partida. Se contabilizarán los metros recorridos para poder batir el récord propio en futuras partidas. El helicóptero estará subiendo mientras se esté pulsando la pantalla; si se deja de tocar, descenderá hasta el nivel del suelo.

Ilustración 11.19 Explicación nivel uno juego Hercopter

Ilustración 11.20 . Nivel 1 juego Herocopter

- En el segundo nivel el objetivo es llegar al final en el menor tiempo posible, recogiendo todas las monedas que se consigan y evitando las bombas, ya que estas podrán quitar puntos o tiempo. Las monedas pueden dar puntos o tiempo extra. Si se termina el nivel antes de finalizar el tiempo se empieza directamente en el siguiente nivel. En este modo de juego no hay vidas disponibles. El helicóptero subirá hasta el punto que se toque en la pantalla. Una vez que llegue descenderá hasta el suelo.

Ilustración 11.21 . Explicación nivel dos juego Herocopter

Ilustración 11.22 . Nivel 2 juego Herocopter

- Finalmente, en el nivel tres el objetivo es hacer un número máximo de monedas, de tal manera que si se alcanza se pasa directamente al siguiente nivel. En este caso como es el último nivel se terminaría el juego. Las bombas te quitarían puntos o vidas. En este modo de juego el helicóptero se mueve en tres niveles. Si el usuario toca la mitad superior de la pantalla el helicóptero sube un nivel y si toca la mitad inferior, baja un nivel.

Ilustración 11.23 . Explicación nivel tres juego Herocopter

Ilustración 11.24 . Nivel 3 juego Herocopter

Al igual que en el juego 1, los diferentes objetos de la pantalla varían su comportamiento dependiendo del nivel. El helicóptero es un objeto de tipo *Character*, y dependiendo de sus propiedades puede generar diferentes objetos.

Ilustración 11.25 . Character con aspecto de helicoptero

Las monedas y bombas son objetos de tipo *Enemy*. Se trata del mismo objeto, pero configurado con diferentes valores.

Ilustración 11.26 .Enemigo con aspecto de bomba

Ilustración 11.27 . Enemigo con aspecto de moneda

11.1.1.2.2 Funcionamiento

Al igual que en el juego "Bichos" en la pantalla del menú principal del juego hay dos botones: "Jugar" y "Salir", y el comportamiento de ambos es el mismo que en el juego anterior.

Ilustración 11.28 . Botón jugar juego Herocopter

Ilustración 11.29 . Botón salir juego Herocopter

Ya seleccionado el nivel, se muestra en pantalla una breve descripción del nivel y de sus objetivos. Pulsado sobre ella se accede al juego.

Ilustración 11.30 . Pantalla selección de nivel juego Herocopter

En la parte superior de la pantalla hay una barra de estado en la que se muestran diferentes iconos y marcadores dependiendo del nivel: los puntos, las vidas disponibles, el tiempo restante para finalizar el nivel, los metros recorridos y un botón de pausa. Si se hace clic sobre este último botón, se para el juego temporalmente y se abre una ventana de pausa con idéntico funcionamiento a la del juego "Bichos".

Ilustración 11.31 . Iconos de barra de estado juego Herocopter

Ilustración 11.32 . Pantalla de pausa juego de Herocopter

En cada nivel hay un modo de juego distinto, pero la manera de interactuar con los elementos es la misma, para recoger las monedas o evitar las bombas el usuario mueve el helicóptero. La pantalla de juego se encuentra dividida en varios topes o niveles en los que se sitúan las bombas y monedas que se mueven con una velocidad constante de derecha a izquierda, y pueden tener diferentes comportamientos:

- Dar vidas extra o quitar vidas
- Dar puntos o quitarlos
- Quitar tiempo o añadir unos segundos al tiempo restante
- Tener que ser esquivados o tener que chocar contra ellos

El helicóptero, que inicialmente se encuentra situado en el suelo, se puede mover de diferentes formas:

- Mientras el usuario mantenga pulsada la pantalla, el helicóptero ganará altura hasta llegar a la parte superior y se mantendrá en esa posición mientras el usuario no levante el dedo. Una vez que lo haga, el helicóptero descenderá lentamente hacia el nivel del suelo. Si mientras está cayendo el usuario vuelve a tocar la pantalla, el helicóptero volverá a subir.
- En la pantalla de juego hay varios niveles o topes en los que se encuentran situados las bombas o monedas. Si el usuario toca la parte superior de la pantalla el helicóptero subirá un nivel y se quedará fijo en él, y subirá tanto niveles como veces toque la pantalla hasta llegar al último tope. Si toca la parte inferior de la pantalla el helicóptero descenderá un nivel, y así hasta que llegue al suelo.
- El helicóptero se moverá poco a poco hasta la posición del eje y donde el usuario haya tocado la pantalla. Una vez que haya llegado a la coordenada de destino el helicóptero caerá suavemente hasta el suelo. Si mientras el helicóptero está subiendo el usuario toca la pantalla no sucederá nada, solo subirá si se encuentra en el tope 1, es decir, el suelo.

Una vez superado el nivel se accede a una pantalla de victoria en la que se muestran los puntos totales obtenidos y los metros recorridos, además de tres botones que permitirán al usuario repetir el nivel, pasar al nivel siguiente (si este no es el último del juego) o volver a la pantalla de selección de nivel.

Ilustración 11.33 . Pantalla de victoria juego Herocopter

En caso de no haber superado el nivel, se muestra una pantalla de derrota en la que se permite, mediante dos botones, repetir el nivel o ir a la pantalla de selección de nivel.

En caso de superar el último nivel, se muestra al usuario una pantalla de fin de juego, desde la cual podrá volver al menú de inicio.

11.1.1.3 Características de los elementos del juego

Los objetos *Enemy* y *Character* utilizados en ambos juegos son los mismos, pero cada uno se ha configurado de forma distinta.

Los enemigos *Enemy* tienen las siguientes propiedades configurables:

- Imágenes: por defecto, movimiento a la izquierda, a la derecha, hacia delante, hacia atrás, destrucción.
- Ancho y alto
- Posición en eje X y posición en eje Y
- Velocidad en el eje X y en el eje Y
- Velocidad máxima y mínima
- Tipo de movimiento y tipo de gravedad
- Gravedad y aceleración de la gravedad
- Tiempo de explotar
- Si da puntos o quita vidas
- Puntos que se obtienen al destruirlo
- Si hay que esquivarlo o chocar contra él
- Sonidos

Los personajes principales o *Character* tienen las siguientes:

- Imágenes: por defecto, movimiento a la izquierda, a la derecha, hacia delante, hacia atrás, destrucción.
- Ancho y alto
- Posición en eje X y posición en eje Y
- Velocidad en el eje X y en el eje Y
- Velocidad máxima y mínima
- Gravedad y aceleración de la gravedad
- Si sigue el movimiento del dedo del usuario por la pantalla o va directamente a donde haga el último toque.
- Lista de topes/niveles por los que se moverá en la pantalla.
- Propulsión de la gravedad
- Sonidos

De esta manera se pueden obtener muchísimos objetos diferentes a partir de solo dos tipos, lo que permite crear muchos juegos diferentes. Por ejemplo, solamente cambiando las imágenes de los objetos de los juegos base por otras distintas ya se obtendrían otros juegos diferentes.

Además de poder configurar los enemigos y los personajes principales, también se podrán configurar los siguientes elementos:

- Pantalla de menú principal
- Pantalla de selección de nivel

- Explicación del funcionamiento de los diferentes niveles
- Fondos de pantalla y elementos de la pantalla (marcadores, iconos, etc.) de la pantalla principal del juego
- Pantalla de pausa
- Pantalla de derrota o de victoria
- Pantalla de fin de juego
- Sonidos de fondo o de paso de pantalla

Por tanto, todos esos valores configurables son las partes variables de los juegos, que se utilizaron para construir la plantilla y cuando se hayan obtenido todos los valores de los diferentes elementos de la instancia del DSL se sustituirán en la plantilla, generando los nuevos juegos que el usuario hay diseñado desde la interfaz de usuario de la herramienta construida en este proyecto fin de máster.

Mediante unas imágenes de los juegos se ilustran a continuación esas diferencias de configuración de los valores:

Pantalla menú principal, de selección de nivel y de fin de juego del juego "Bichos":

Pantalla menú principal, de selección de nivel y fin de juego del Juego "Herocopter":

Como se puede observar, solamente cambiando las imágenes de los fondos, de los botones y de los iconos, y los valores de sus posiciones en la pantalla, ya se obtienen dos interfaces para dos juegos diferentes.

En el caso de la pantalla principal de juego, se puede ver que los enemigos tienen diferentes imágenes, lo que hace que ya sean personajes distintos, independientemente de su comportamiento.

11.2 Videojuegos desarrollados y disponibles en Android

Una de las empresas participantes en el proyecto Gade4all decidió utilizar la herramienta para crear juegos relacionados con la Eurocopa 2012 y los Juegos Olímpicos de Londres 2012. Se realizaron varios juegos diferentes utilizando la instancia del DSL desarrollada y el editor de niveles (en ese momento este aún estaba en fase de construcción), y como resultado final se obtuvieron una serie de juegos que se encuentran disponibles para ser descargados en el market de Android.

11.2.1 Regateo Extremo

Este juego está basado en que el personaje principal esquive a los jugadores del equipo contrario mientras corre por el campo con el balón en su poder. Si se pulsa en la mitad superior de la pantalla el jugador se moverá hacia arriba y, al hacerlo en la inferior, se moverá hacia abajo. Se dispone de un número limitado de vidas y el jugador tiene que intentar llegar a la portería sin chocar con los otros jugadores. El juego cuenta con diferentes niveles de dificultad, establecidos en base a la cantidad de jugadores del equipo contrario que se encuentran en el campo y la velocidad de los mismos. Si se agotan las vidas disponibles, se pierde la partida.

Además cuenta con un registro de récords, en el que se calculan los puntos del jugador dependiendo de los metros recorridos en el campo; cuantas más vidas se disponga al final de nivel, se adjudicarán más puntos.

Ilustración 11.34. Regateo Extremo, pantalla principal

Ilustración 11.35. Regateo Extremo, pantalla juego

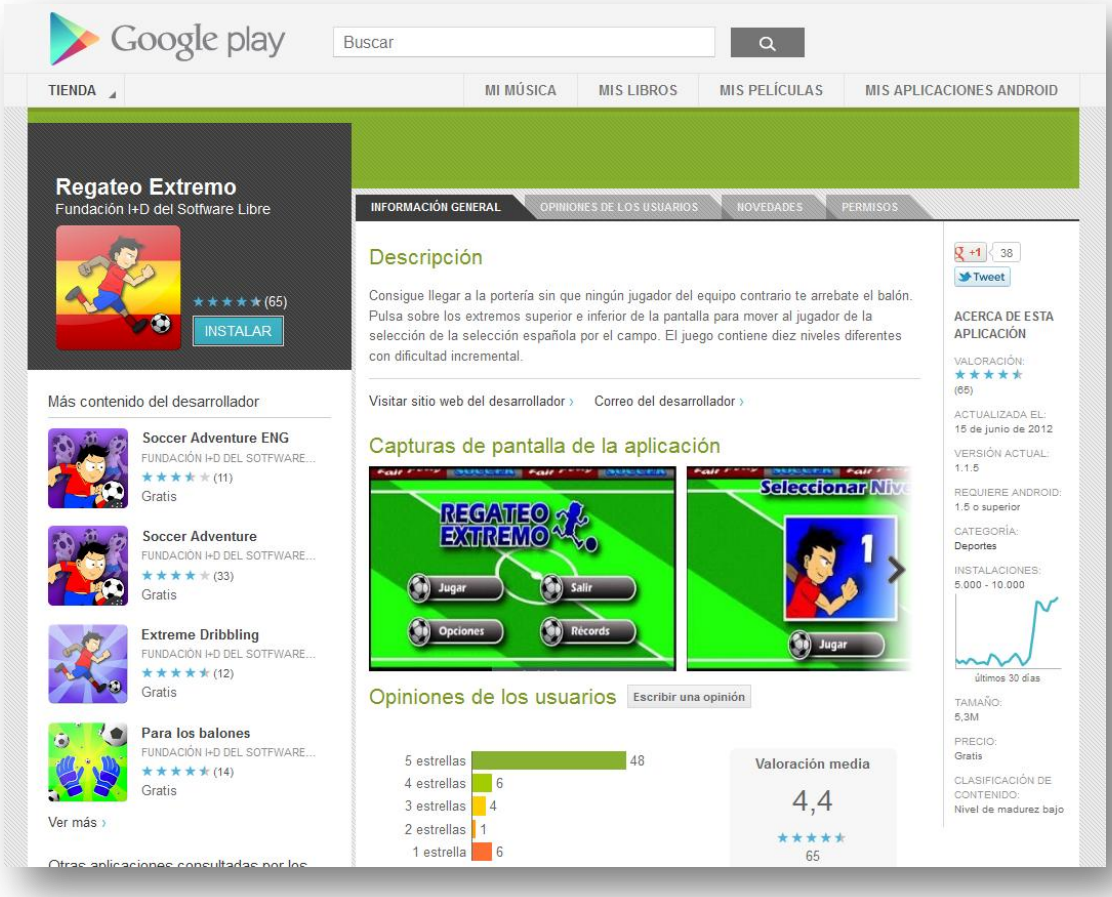


Ilustración 11.36. Regateo Extremo en Google Play

11.2.2 Balonazo a la grada

En este juego el usuario tiene que mover, mediante toques en la pantalla, una mirilla y disparar balonazos a los jugadores de la selección de Polonia. Mientras mantenga pulsado el dedo sobre la pantalla, la mirilla se moverá hacia arriba y si deja de hacerlo, caerá hasta la parte inferior. En caso de disparar balonazos a los jugadores de la selección de España, se perderá una vida por cada jugador golpeado; en cambio, al golpear a los polacos, aumentará su puntuación. El juego cuenta con diferentes niveles de dificultad, establecidos en base a la cantidad de jugadores del equipo de Polonia que se encuentran sentados en las gradas y la velocidad de los mismos. Si se agotan las vidas disponibles, se pierde la partida.

También cuenta con un registro de récords, en el que se calculan los puntos del jugador dependiendo de los balonazos que haya dado disparando con la mirilla. Cuantas más vidas se conserven al final de nivel, se lograrán más puntos.

Ilustración 11.37. Balonazo A La Grada, pantalla principal

Ilustración 11.38. Balonazo A La Grada, pantalla juego

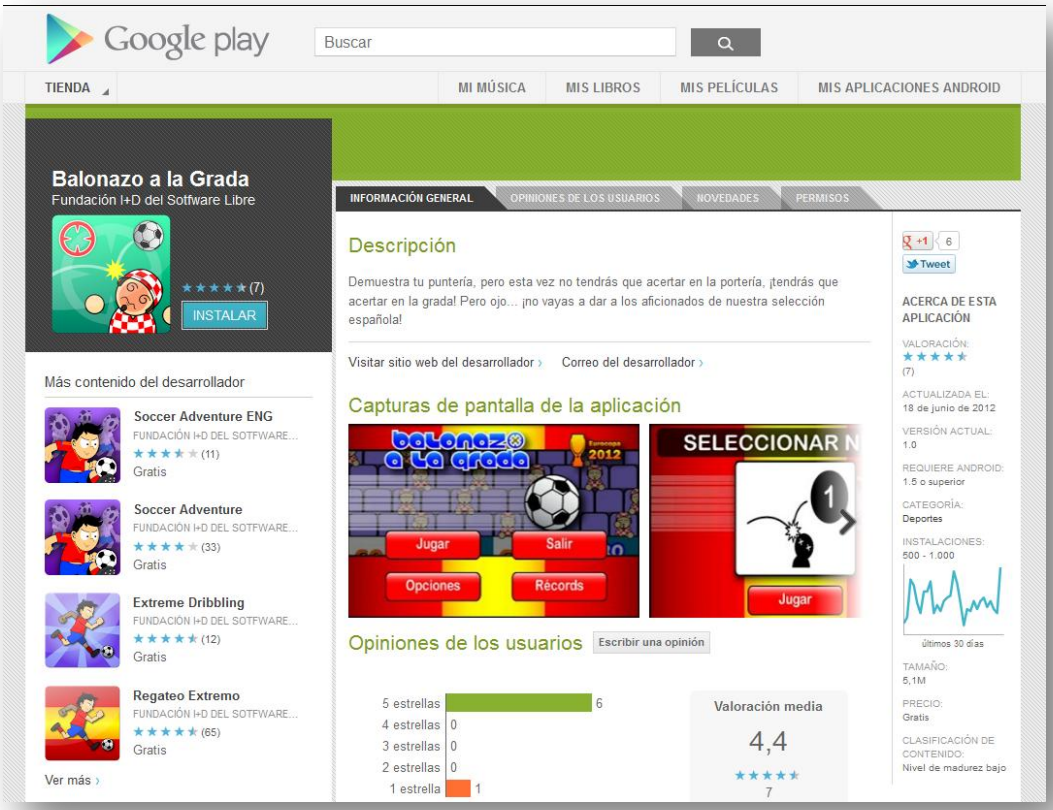


Ilustración 11.39. Balonazo A La Grada en Google Play

11.2.3 Para los balones

En este juego el usuario tiene que mover, desplazando su dedo por la pantalla, unos guantes e intentar parar los balones que se mueven hacia la portería. Los guantes seguirán el movimiento del dedo por la pantalla, y cuando se deje de pulsar, los guantes dejarán de moverse. Cada balón que no sea atrapado y atraviese la portería restará una vida del total disponible y cada balón que se pare con los guantes sumará puntos al marcador. El juego cuenta con diferentes niveles de dificultad, establecidos en base a la cantidad de balones que son lanzados, al tipo de movimiento y a la velocidad de los mismos. Si se agotan las vidas disponibles, se pierde la partida.

También cuenta con un registro de récords, en el que se calculan los puntos del jugador dependiendo de los balones que haya parado. Cuantas más vidas se disponga al final de nivel, se adjudicarán más puntos.

Ilustración 11.40. Para Los Balones, pantalla principal

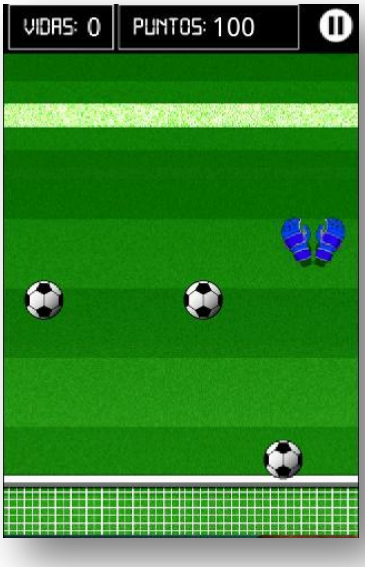


Ilustración 11.41. Para Los Balones, pantalla juego

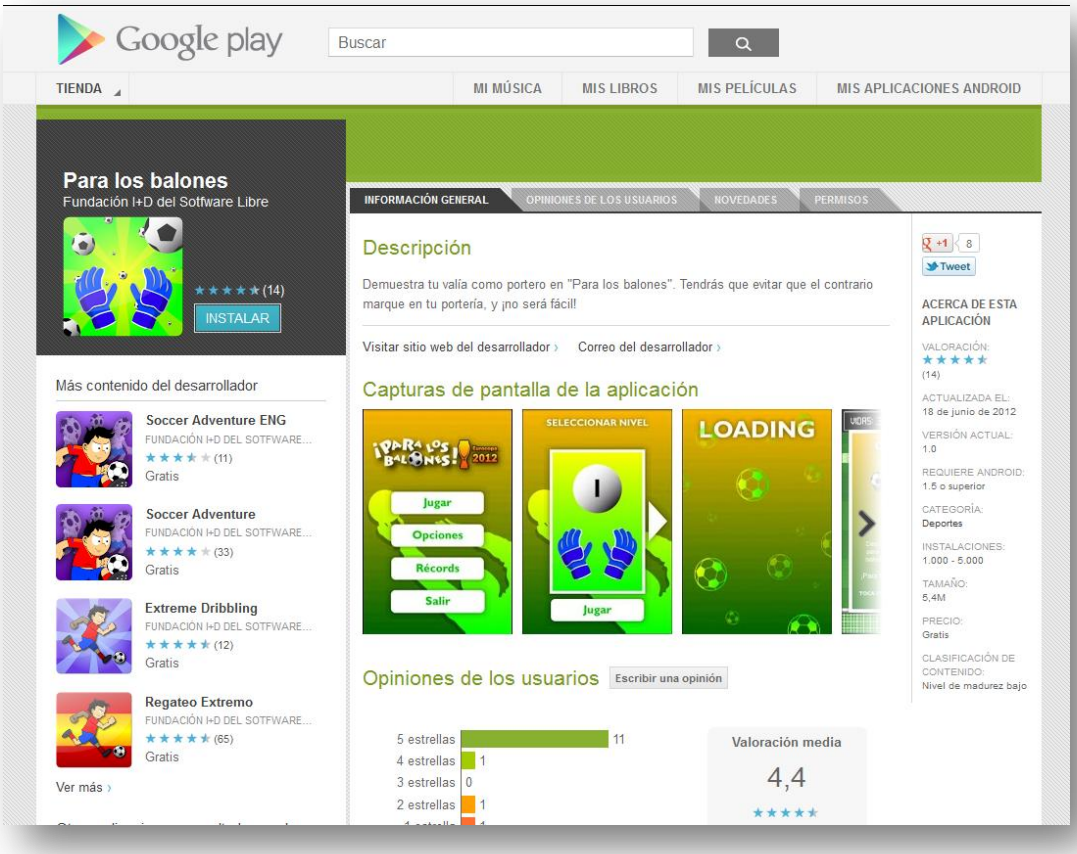


Ilustración 11.42. Para Los Balones en Google Play

11.2.4 Tiro Skeet

En este juego el usuario tiene que mover, mediante toques en la pantalla, una escopeta y disparar a los discos que van apareciendo. Mientras mantenga pulsado el dedo sobre la pantalla, la escopeta se moverá hacia arriba y si deja de pulsar caerá hasta la parte inferior. En caso de disparar a los pájaros, perderá una vida por cada uno al que dispare, en cambio, si dispara a los discos aumentará su puntuación. El juego cuenta con diferentes niveles de dificultad, establecidos en base a la cantidad de pájaros que sobrevuelan el campo y la velocidad de los mismos. Si se agotan las vidas disponibles, se pierde la partida.

También cuenta con un registro de récords, en el que se calculan los puntos del jugador dependiendo de los discos que haya destruido con la escopeta. Cuantas más vidas se conserven al final de nivel, se lograrán más puntos.

Ilustración 11.43. Tiro Skeet, pantalla principal

Ilustración 11.44. Tiro Skeet, pantalla juego

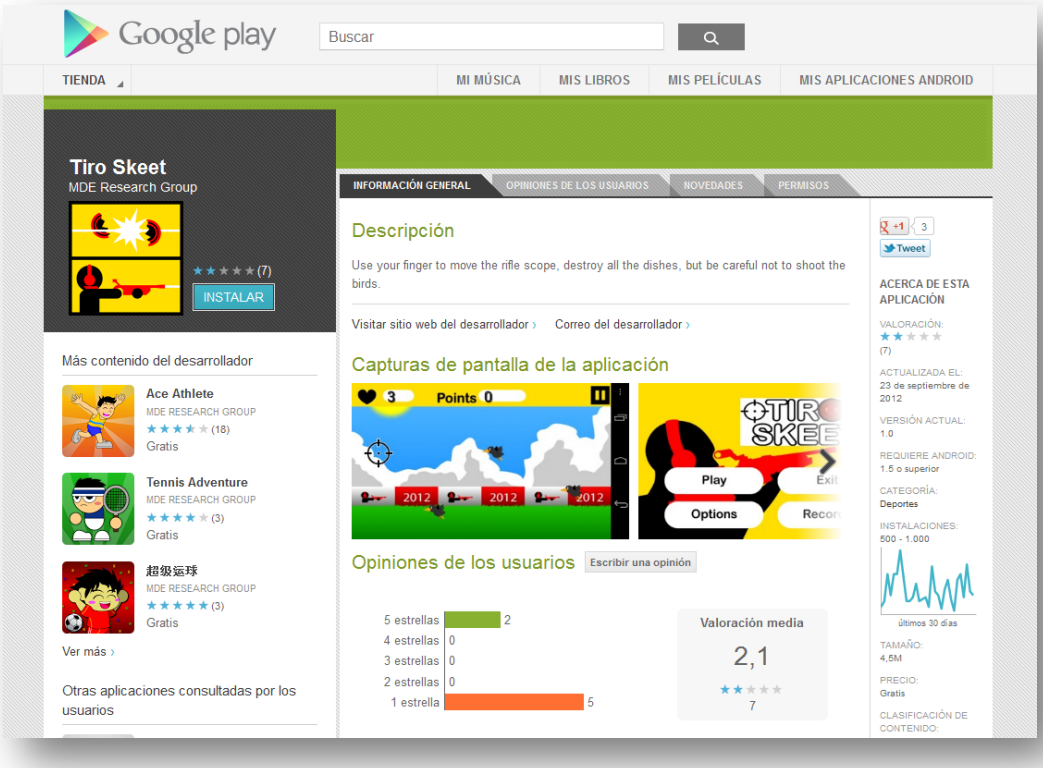


Ilustración 11.45. Tiro Skeet en Google Play

11.2.5 Ace Athlete

En este juego un personaje principal tiene que adelantar a los otros jugadores que corren por la pista sin chocar con ninguno de ellos. Si se pulsa en la mitad superior de la pantalla, el jugador se moverá hacia arriba y al hacerlo en la inferior, se moverá hacia abajo. Se cuenta con número limitado de vidas y el corredor tiene que intentar llegar a la meta sin chocar con los otros atletas. El juego cuenta con diferentes niveles de dificultad, establecidos en base a la cantidad de atletas que corren por la pista y la velocidad de los mismos. Si se agotan las vidas disponibles, se pierde la partida.

Además cuenta con un registro de récords, en el que se calculan los puntos del jugador dependiendo de los metros recorridos en la pista. Cuantas más vidas se tenga al final de nivel, se adjudicarán más puntos.

Ilustración 11.46. Ace Athlete, pantalla principal

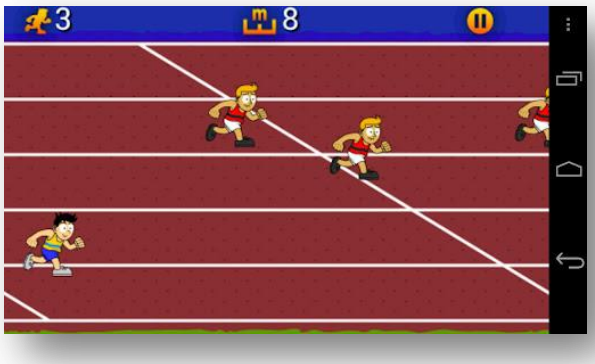


Ilustración 11.47. Ace Athlete, pantalla juego

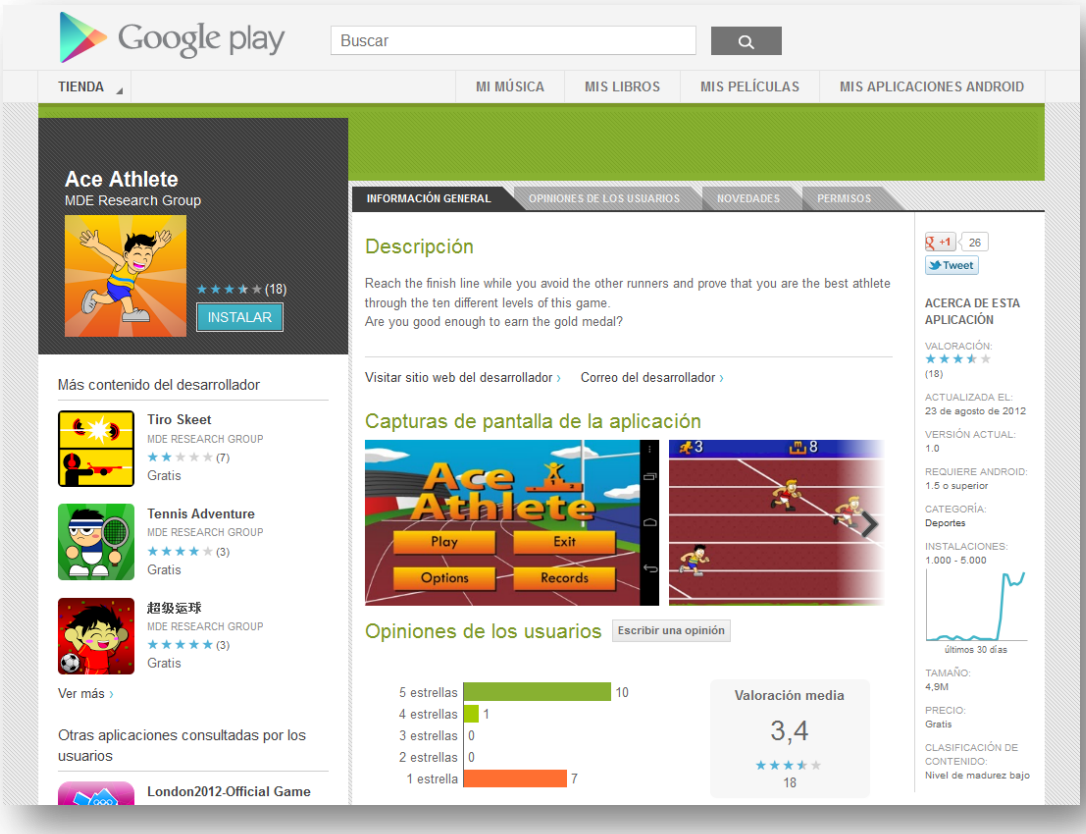


Ilustración 11.48. Ace Athlete en Google Play

Capítulo 12. Referencias Bibliográficas

12.1 Libros y Artículos

[FOWLER99] Fowler, Martín. Scott, Kendall. "UML: GOTA A GOTA". Año 1999.

[SALTZMAN01] Saltzman, Marc. "COMO DISEÑAR VIDEOJUEGOS. LOS SECRETOS DE LOS EXPERTOS". Norma Editorial. ISBN: 84-8431-435-9. Año 2001.

[KLEPPE03] Kleppe, A. Warmer, J. "MDA EXPLAINED. THE MODEL DRIVEN ARCHITECTURE. PRACTICE AND PROMISE". Addison-Wesley. Año 2003.

[SCHMIDT06] Schmidt, D. C. "MODEL-DRIVEN ENGINEERING". IEEE Computer. Año 2006.

[TOLVANEN08] Kelly, Steve. Tolvanen, Juha-Pekka. "APLICACIÓN DE LA INGENIERÍA POR MODELOS: EJEMPLOS". Año 2008.

[SANJUAN09] Sanjuán Martínez, Oscar. Cueva Lovelle, Juan Manuel. "XML. EXTENSIBLE MARKUP LANGUAGE". OOTLab, Universidad de Oviedo. Año 2009.

[BURNETTE09] Burnette, Ed. "HELLO, ANDROID: INTRODUCING GOOGLE'S MOBILE DEVELOPMENT PLATFORM". Pragmatic Bookshelf Editorial. ISBN: 1934356492-9781934356494.Año 2009.

[ROGERS09] Rogers, Rick. Lombardo, John. Mednieks, Zigurd. Meike, Blake. "ANDROID APPLICATION DEVELOPMENT: PROGRAMMING WITH THE GOOGLE SDK." O'Reilly Media, Inc. ISBN:0596521472 – 97805962521479.Año 2009

12.2 Referencias en Internet

Páginas Web consultadas para cualquier aspecto relacionado con el desarrollo del sistema o su documentación.

C#: http://msdn.microsoft.com/es-es/library/z1zx9t92(v=vs.80).aspx

XML: http://www.w3.org/XML/

XAML:http://msdn.microsoft.com/es-es/library/ms752059.aspx , http://msdn.microsoft.com/es-es/library/cc295302.aspx

ANDROID: http://developer.android.com/index.html

DSL: http://msdn.microsoft.com/en-us/library/bb126278.aspx

[REENSKAUG78] Reenskaug, Trygve M. H. "MVC (MODEL-VIEW-CONTROLER). XEROX PARC 1978-79". http://heim.ifi.uio.no/~trygver/. Año 1978.

[FURTADO06] W. B. Furtado, André. L. de M. Santos, André. "APPLYING DOMAIN-SPECIFIC MODELING TO GAME DEVELOPMENT WITH THE MICROSOFT DSL TOOLS". http://www.cin.ufpe.br/~awbf/files/SBGames2006_TutorialGamesDSM.pdf. Año 2006.

[ANACLETO06] Anacleto, Valerio Adrián. "MDA: REUSABILIDAD ORIENTADA AL NEGOCIO". http://www.epidataconsulting.com/tikiwiki/tiki-read_article.php?articleId=55. Año 2006.

[DESETA09] De Seta, Leonardo. "LOS LENGUAJES ESPECÍFICOS DE DOMINO". http://www.dosideas.com/actualidad/487-los-lenguajes-especificos-de-dominio.html. Año 2009.

[MONTERO11] Montero, Emanuel. Carsí, José A. "MDA Y DESARROLLO DE VIDEOJUEGOS". http://lbd.udc.es/jornadas2011/actas/JISBD/JISBD/S5/Regulares/jisbd2011_submission_13.pdf. Año 2011.

Capítulo 13. Apéndices

13.1 Glosario y Diccionario de Datos

- **Android**: Sistema operativo basado en Linux desarrollado para su ejecución en teléfonos inteligentes y otros dispositivos móviles.
- **Colisionar**: Acción que se produce cuando dos elementos chocan entre sí.
- **DSL**: Lenguaje específico de dominio (Domain Specific Language) es un lenguaje empleado para resolver un problema en particular.
- **Eclipse**: Entorno de desarrollo utilizado para desarrollar aplicaciones en Java.

13.2 Contenido Entregado en el CD-ROM

13.2.1 Contenidos

En esta sección se presenta el contenido del CD que se presenta junto con este proyecto.

13.2.1.1.1 Estructura de directorios

Directorio	Contenido
./Directorio raíz del CD	Contiene un fichero leeme.txt explicando toda esta estructura.
./DSL_VideojuegoAccion2D	Contiene toda la estructura de directorios del proyecto.
./ DSL_VideojuegoAccion2D /proyecto	Contiene el código fuente de la herramienta divido en subcarpetas. También contiene los ficheros necesarios para realizar la instalación y unas imágenes de ejemplo para diseñar un juego.
./ DSL_VideojuegoAccion2D /documentacion	Contiene toda la documentación asociada al proyecto. PROTEGIDA
./DSL_VideojuegoAccion2D /documentacion/uml	Ficheros que genera la herramienta Enterprise Architect, con la que se han generado los diagramas UML.
./DSL_VideojuegoAccion2D /documentacion/planificacion	Fichero que contiene la planificación del proyecto que se ha generado con la herramienta Gantt Proyect.
./DSL_VideojuegoAccion2D /documentacion/ResultadosPruebasUsabilidad	Fichero que contiene los resultados de las pruebas de usabilidad realizadas a los 12 individuos. Se incluye una versión docx.
./DSL_VideojuegoAccion2D /documentacion/Javadoc	Ficheros generados por la herramienta Javadoc de generación de documentación automática del código de la plantilla.
./DSL_VideojuegoAccion2D /documentacion/SandCastle	Ficheros generados por la herramienta SandCastle de generación de documentación automática del código de la interfaz y analizador.

13.3 Índice Alfabético

A

analizador, 27, 32, 62, 85, 97, 105, 108, 110, 146, 185, 207, 208, 210, 223, 226, 276, 354, 357
Android, 5, 7, 9, 11, 23, 24, 25, 26, 28, 31, 32, 33, 34, 35, 38, 39, 44, 45, 49, 50, 52, 55, 69, 70, 71, 72, 73, 86, 142, 187, 207, 217, 223, 226, 275, 292, 298, 349, 350, 354, 358, 376, 389

D

diagrama, 63, 119, 147, 152, 156
DSL, 53, 56, 105, 106, 107, 108

E

editor, 9, 25, 31, 32, 34, 39, 40, 44, 45, 46, 49, 50, 53, 72, 84, 97, 105, 106, 109, 110, 207, 218, 219, 221, 285, 338, 359, 376

G

Gade4All, 53

I

iPhone, 53, 105, 106, 107, 108

P

personajes, 50
plantillas, 5, 50, 51, 223, 354
pruebas, 26, 28, 29, 65, 73, 120, 121, 211, 215, 216, 241, 242, 279, 285, 359

S

Stencyl, 51

V

videojuegos, 53, 56

X

XML, 53

13.4 Anexo I. Instancia del DSL generada con la herramienta.

```
<game>
        <global>
        <global_name><!-- String --></global_name>
        <global_package_name><!-- String --></global_package_name>
        <global_screen_orientation><!-- INT {0,1} --></global_screen_orientation>
        <global_icon_source><!-- String --></global_icon_source>
        <global_language><!-- String --></global_language>
        <global_number_fps><!-- INT --></global_number_fps>
        <global_has_advertising><!-- String {NO,YES} --></global_has_advertising>
        <global_admob_id><!-- String --></global_admob_id>
        <global_sound_source><!-- String --></global_sound_source>
        </global>

<screens>
<main_screen>
<main_screen_start_button>
<main_screen_start_button_image_source><!-- String --></main_screen_start_button_image_source>
<main_screen_start_button_image_width><!-- INT --></main_screen_start_button_image_width>
<main_screen_start_button_image_height><!-- INT --></main_screen_start_button_image_height>
<main_screen_start_button_x_position><!-- INT --></main_screen_start_button_x_position>
<main_screen_start_button_y_position><!-- INT --></main_screen_start_button_y_position>
<main_screen_start_button_sound_source><!-- String --></main_screen_start_button_sound_source>
</main_screen_start_button>
<main_screen_options_button>
<main_screen_options_button_image_source><!-- String --></main_screen_options_button_image_source>
<main_screen_options_button_image_width><!-- INT --></main_screen_options_button_image_width>
<main_screen_options_button_image_height><!-- INT --></main_screen_options_button_image_height>
<main_screen_options_button_x_position><!-- INT --></main_screen_options_button_x_position>
<main_screen_options_button_y_position><!-- INT --></main_screen_options_button_y_position>
<main_screen_options_button_sound_source> <!-- String --> </main_screen_options_button_sound_source>
</main_screen_options_button>
<main_screen_records_button>
<main_screen_records_button_image_source><!-- String --></main_screen_records_button_image_source>
<main_screen_records_button_image_width><!-- INT --></main_screen_records_button_image_width>
<main_screen_records_button_image_height><!-- INT --></main_screen_records_button_image_height>
<main_screen_records_button_x_position><!-- INT --></main_screen_records_button_x_position>
<main_screen_records_button_y_position><!-- INT --></main_screen_records_button_y_position>
<main_screen_records_button_sound_source><!-- String --></main_screen_records_button_sound_source>
</main_screen_records_button>
<main_screen_exit_button>
<main_screen_exit_button_image_source><!-- String --></main_screen_exit_button_image_source>
<main_screen_exit_button_image_width><!-- INT --></main_screen_exit_button_image_width>
<main_screen_exit_button_image_height><!-- INT --></main_screen_exit_button_image_height>
<main_screen_exit_button_x_position><!-- INT --></main_screen_exit_button_x_position>
<main_screen_exit_button_y_position><!-- INT --></main_screen_exit_button_y_position>
<main_screen_exit_button_sound_source><!-- String --></main_screen_exit_button_sound_source>
</main_screen_exit_button>
<main_screen_images>
<main_screen_image>
<main_screen_image_source><!-- String --></main_screen_image_source>
<main_screen_image_width><!-- INT --></main_screen_image_width>
<main_screen_image_height><!-- INT --></main_screen_image_height>
<main_screen_image_x_position><!-- INT --></main_screen_image_x_position>
<main_screen_image_y_position><!-- INT --></main_screen_image_y_position>
</main_screen_image>
</main_screen_images>
</main_screen>

<options_screen>
<options_screen_music_off_button>
```

```
<options_screen_music_off_button_image_source><!-- String --></options_screen_music_off_button_image_source>
<options_screen_music_off_button_image_width><!-- INT --></options_screen_music_off_button_image_width>
<options_screen_music_off_button_image_height><!-- INT --></options_screen_music_off_button_image_height>
<options_screen_music_off_button_x_position><!-- INT --></options_screen_music_off_button_x_position>
<options_screen_music_off_button_y_position><!-- INT --></options_screen_music_off_button_y_position>
<options_screen_music_off_button_sound_source><!-- String --></options_screen_music_off_button_sound_source>
</options_screen_music_off_button>
<options_screen_music_on_button>
<options_screen_music_on_button_image_source><!-- String --></options_screen_music_on_button_image_source>
<options_screen_music_on_button_image_width><!-- INT --></options_screen_music_on_button_image_width>
<options_screen_music_on_button_image_height><!-- INT --></options_screen_music_on_button_image_height>
<options_screen_music_on_button_x_position><!-- INT --></options_screen_music_on_button_x_position>
<options_screen_music_on_button_y_position><!-- INT --></options_screen_music_on_button_y_position>
<options_screen_music_on_button_sound_source><!-- String --></options_screen_music_on_button_sound_source>
</options_screen_music_on_button>
<options_screen_images>
<options_screen_image>
<options_screen_image_source><!-- String --></options_screen_image_source>
<options_screen_image_width><!-- INT --></options_screen_image_width>
<options_screen_image_height><!-- INT --></options_screen_image_height>
<options_screen_image_x_position><!-- INT --></options_screen_image_x_position>
<options_screen_image_y_position><!-- INT --></options_screen_image_y_position>
</options_screen_image>
</options_screen_images>
</options_screen>

<select_level_screen>
<select_level_screen_next_level_button>
<select_level_screen_next_level_button_image_source><!-- String --></select_level_screen_next_level_button_image_source>
<select_level_screen_next_level_button_image_width><!-- INT --></select_level_screen_next_level_button_image_width>
<select_level_screen_next_level_button_image_height><!-- INT --></select_level_screen_next_level_button_image_height>
<select_level_screen_next_level_button_x_position><!-- INT --></select_level_screen_next_level_button_x_position>
<select_level_screen_next_level_button_y_position><!-- INT --></select_level_screen_next_level_button_y_position>
<select_level_screen_next_level_button_sound_source><!-- String --></select_level_screen_next_level_button_sound_source>
</select_level_screen_next_level_button>
<select_level_screen_previous_level_button>
<select_level_screen_previous_level_button_image_source><!--String--></select_level_screen_previous_level_button_image_source>
<select_level_screen_previous_level_button_image_width><!-- INT --></select_level_screen_previous_level_button_image_width>
<select_level_screen_previous_level_button_image_height><!-- INT --></select_level_screen_previous_level_button_image_height>
<select_level_screen_previous_level_button_x_position><!-- INT --></select_level_screen_previous_level_button_x_position>
<select_level_screen_previous_level_button_y_position><!-- INT --></select_level_screen_previous_level_button_y_position>
<select_level_screen_previous_level_button_sound_source><!-- String --></select_level_screen_previous_level_button_sound_source>
</select_level_screen_previous_level_button>
<select_level_screen_play_level_button>
<select_level_screen_play_level_button_image_source><!-- String --></select_level_screen_play_level_button_image_source>
<select_level_screen_play_level_button_image_width><!-- INT --></select_level_screen_play_level_button_image_width>
<select_level_screen_play_level_button_image_height><!-- INT --></select_level_screen_play_level_button_image_height>
<select_level_screen_play_level_button_x_position><!-- INT --></select_level_screen_play_level_button_x_position>
<select_level_screen_play_level_button_y_position><!-- INT --></select_level_screen_play_level_button_y_position>
<select_level_screen_play_level_button_sound_source><!-- String --></select_level_screen_play_level_button_sound_source>
</select_level_screen_play_level_button>
<select_level_screen_level_thumbnail_view>
<select_level_screen_level_thumbnail_view_image_width><!-- INT --></select_level_screen_level_thumbnail_view_image_width>
<select_level_screen_level_thumbnail_view_image_height><!-- INT --></select_level_screen_level_thumbnail_view_image_height>
<select_level_screen_level_thumbnail_view_x_position><!-- INT --></select_level_screen_level_thumbnail_view_x_position>
<select_level_screen_level_thumbnail_view_y_position><!-- INT --></select_level_screen_level_thumbnail_view_y_position>
<select_level_screen_level_thumbnail_view_sound_source><!-- String --></select_level_screen_level_thumbnail_view_sound_source>
</select_level_screen_level_thumbnail_view>
<select_level_screen_level_images>
<select_level_screen_level_image>
<select_level_screen_level_image_source><!-- String --></select_level_screen_level_image_source>
<select_level_screen_level_image_width><!-- INT --></select_level_screen_level_image_width>
<select_level_screen_level_image_height><!-- INT --></select_level_screen_level_image_height>
<select_level_screen_level_image_x_position><!-- INT --></select_level_screen_level_image_x_position>
<select_level_screen_level_image_y_position><!-- INT --></select_level_screen_level_image_y_position>
</select_level_screen_level_image>
</select_level_screen_level_images>
</select_level_screen>

<loading_screen>
<loading_screen_images>
```

```
<loading_screen_image>
<loading_screen_image_source><!-- String --></loading_screen_image_source>
<loading_screen_image_width><!-- INT --></loading_screen_image_width>
<loading_screen_image_height><!-- INT --></loading_screen_image_height>
<loading_screen_image_x_position><!-- INT --></loading_screen_image_x_position>
<loading_screen_image_y_position><!-- INT --></loading_screen_image_y_position>
</loading_screen_image>
</loading_screen_images>
</loading_screen>

<pause_screen>
<pause_screen_reanude_button>
<pause_screen_reanude_button_image_source><!-- String --></pause_screen_reanude_button_image_source>
<pause_screen_reanude_button_image_width><!-- INT --></pause_screen_reanude_button_image_width>
<pause_screen_reanude_button_image_height><!-- INT --></pause_screen_reanude_button_image_height>
<pause_screen_reanude_button_x_position><!-- INT --></pause_screen_reanude_button_x_position>
<pause_screen_reanude_button_y_position><!-- INT --></pause_screen_reanude_button_y_position>
<pause_screen_reanude_button_sound_source><!-- String --></pause_screen_reanude_button_sound_source>
</pause_screen_reanude_button>
<pause_screen_exit_button>
<pause_screen_exit_button_image_source><!-- String --></pause_screen_exit_button_image_source>
<pause_screen_exit_button_image_width><!-- INT --></pause_screen_exit_button_image_width>
<pause_screen_exit_button_image_height><!-- INT --></pause_screen_exit_button_image_height>
<pause_screen_exit_button_x_position><!-- INT --></pause_screen_exit_button_x_position>
<pause_screen_exit_button_y_position><!-- INT --></pause_screen_exit_button_y_position>
<pause_screen_exit_button_sound_source><!-- String --></pause_screen_exit_button_sound_source>
</pause_screen_exit_button>
<pause_screen_images>
<pause_screen_image>
<pause_screen_image_source><!-- String --></pause_screen_image_source>
<pause_screen_image_width><!-- INT --></pause_screen_image_width>
<pause_screen_image_height><!-- INT --></pause_screen_image_height>
<pause_screen_image_x_position><!-- INT --></pause_screen_image_x_position>
<pause_screen_image_y_position><!-- INT --></pause_screen_image_y_position>
</pause_screen_image>
</pause_screen_images>
</pause_screen>

<end_level_screen>
<end_level_screen_retry_button>
<end_level_screen_retry_button_image_source><!-- String --></end_level_screen_retry_button_image_source>
<end_level_screen_retry_button_image_width><!-- INT --></end_level_screen_retry_button_image_width>
<end_level_screen_retry_button_image_height><!-- INT --></end_level_screen_retry_button_image_height>
<end_level_screen_retry_button_x_position><!-- INT --></end_level_screen_retry_button_x_position>
<end_level_screen_retry_button_y_position><!-- INT --></end_level_screen_retry_button_y_position>
<end_level_screen_retry_button_sound_source><!-- String --></end_level_screen_retry_button_sound_source>
</end_level_screen_retry_button>
<end_level_screen_play_next_level_button>
<end_level_screen_play_next_level_button_image_source><!-- String --></end_level_screen_play_next_level_button_image_source>
<end_level_screen_play_next_level_button_image_width><!-- INT --></end_level_screen_play_next_level_button_image_width>
<end_level_screen_play_next_level_button_image_height><!-- INT --></end_level_screen_play_next_level_button_image_height>
<end_level_screen_play_next_level_button_x_position><!-- INT --></end_level_screen_play_next_level_button_x_position>
<end_level_screen_play_next_level_button_y_position><!-- INT --></end_level_screen_play_next_level_button_y_position>
<end_level_screen_play_next_level_button_sound_source><!-- String --></end_level_screen_play_next_level_button_sound_source>
</end_level_screen_play_next_level_button>
<end_level_screen_images>
<end_level_screen_image>
<end_level_screen_image_source><!-- String --></end_level_screen_image_source>
<end_level_screen_image_width><!-- INT --></end_level_screen_image_width>
<end_level_screen_image_height><!-- INT --></end_level_screen_image_height>
<end_level_screen_image_x_position><!-- INT --></end_level_screen_image_x_position>
<end_level_screen_image_y_position><!-- INT --></end_level_screen_image_y_position>
</end_level_screen_image>
</end_level_screen_images>
<end_level_screen_scoreboard>
<end_level_screen_scoreboard_width><!-- INT --></end_level_screen_scoreboard_width>
<end_level_screen_scoreboard_height><!-- INT --></end_level_screen_scoreboard_height>
<end_level_screen_scoreboard_x_position><!-- INT --></end_level_screen_scoreboard_x_position>
<end_level_screen_scoreboard_y_position><!-- INT --></end_level_screen_scoreboard_y_position>
<end_level_screen_scoreboard_text_font_size><!-- INT --></end_level_screen_scoreboard_text_font_size>
<end_level_screen_scoreboard_text_font_color><!-- String --></end_level_screen_scoreboard_text_font_color>
```

```
</end_level_screen_scoreboard>
</end_level_screen>

<lose_level_screen>
<lose_level_screen_retry_button>
<lose_level_screen_retry_button_image_source><!-- String --></lose_level_screen_retry_button_image_source>
<lose_level_screen_retry_button_image_width><!-- INT --></lose_level_screen_retry_button_image_width>
<lose_level_screen_retry_button_image_height><!-- INT --></lose_level_screen_retry_button_image_height>
<lose_level_screen_retry_button_x_position><!-- INT --></lose_level_screen_retry_button_x_position>
<lose_level_screen_retry_button_y_position><!-- INT --></lose_level_screen_retry_button_y_position>
<lose_level_screen_retry_button_sound_source><!-- String --></lose_level_screen_retry_button_sound_source>
</lose_level_screen_retry_button>
<lose_level_screen_select_level_button>
<lose_level_screen_select_level_button_image_source><!-- String --></lose_level_screen_select_level_button_image_source>
<lose_level_screen_select_level_button_image_width><!-- INT --></lose_level_screen_select_level_button_image_width>
<lose_level_screen_select_level_button_image_height><!-- INT --></lose_level_screen_select_level_button_image_height>
<lose_level_screen_select_level_button_x_position><!-- INT --></lose_level_screen_select_level_button_x_position>
<lose_level_screen_select_level_button_y_position><!-- INT --></lose_level_screen_select_level_button_y_position>
<lose_level_screen_select_level_button_sound_source><!-- String --></lose_level_screen_select_level_button_sound_source>
</lose_level_screen_select_level_button>
<lose_level_screen_images>
<lose_level_screen_image>
<lose_level_screen_image_source><!-- String --></lose_level_screen_image_source>
<lose_level_screen_image_width><!-- INT --></lose_level_screen_image_width>
<lose_level_screen_image_height><!-- INT --></lose_level_screen_image_height>
<lose_level_screen_image_x_position><!-- INT --></lose_level_screen_image_x_position>
<lose_level_screen_image_y_position><!-- INT --></lose_level_screen_image_y_position>
</lose_level_screen_image>
</lose_level_screen_images>
<lose_level_screen_scoreboard>
<lose_level_screen_scoreboard_width><!-- INT --></lose_level_screen_scoreboard_width>
<lose_level_screen_scoreboard_height><!-- INT --></lose_level_screen_scoreboard_height>
<lose_level_screen_scoreboard_x_position><!-- INT --></lose_level_screen_scoreboard_x_position>
<lose_level_screen_scoreboard_y_position><!-- INT --></lose_level_screen_scoreboard_y_position>
<lose_level_screen_scoreboard_text_font_size><!-- INT --></lose_level_screen_scoreboard_text_font_size>
<lose_level_screen_scoreboard_text_font_color><!-- String --></lose_level_screen_scoreboard_text_font_color>
</lose_level_screen_scoreboard>
</lose_level_screen>

<end_game_screen>
<end_game_screen_return_level_button>
<end_game_screen_return_level_button_image_source><!-- String --></end_game_screen_return_level_button_image_source>
<end_game_screen_return_level_button_image_width><!-- INT --></end_game_screen_return_level_button_image_width>
<end_game_screen_return_level_button_image_height><!-- INT --></end_game_screen_return_level_button_image_height>
<end_game_screen_return_level_button_x_position><!-- INT --></end_game_screen_return_level_button_x_position>
<end_game_screen_return_level_button_y_position><!-- INT --></end_game_screen_return_level_button_y_position>
<end_game_screen_return_level_button_sound_source><!-- String --></end_game_screen_return_level_button_sound_source>
</end_game_screen_return_level_button>
<end_game_screen_images>
<end_game_screen_image>
<end_game_screen_image_source><!-- String --></end_game_screen_image_source>
<end_game_screen_image_width><!-- INT --></end_game_screen_image_width>
<end_game_screen_image_height><!-- INT --></end_game_screen_image_height>
<end_game_screen_image_x_position><!-- INT --></end_game_screen_image_x_position>
<end_game_screen_image_y_position><!-- INT --></end_game_screen_image_y_position>
</end_game_screen_image>
</end_game_screen_images>
</end_game_screen>

<game_screen>
<game_screen_pause_button>
<game_screen_pause_button_image_source><!-- String --></game_screen_pause_button_image_source>
<game_screen_pause_button_image_width><!-- INT --></game_screen_pause_button_image_width>
<game_screen_pause_button_image_height><!-- INT --></game_screen_pause_button_image_height>
<game_screen_pause_button_x_position><!-- INT --></game_screen_pause_button_x_position>
<game_screen_pause_button_y_position><!-- INT --></game_screen_pause_button_y_position>
<game_screen_pause_button_sound_source><!-- String --></game_screen_pause_button_sound_source>
</game_screen_pause_button>
<game_screen_status_bar>
<game_screen_status_bar_background>
<game_screen_status_bar_background_source><!-- String --></game_screen_status_bar_background_source>
```

```
<game_screen_status_bar_background_width><!-- INT --></game_screen_status_bar_background_width>
<game_screen_status_bar_background_height><!-- INT --></game_screen_status_bar_background_height>
<game_screen_status_bar_background_x_position><!-- INT --></game_screen_status_bar_background_x_position>
<game_screen_status_bar_background_y_position><!-- INT --></game_screen_status_bar_background_y_position>
</game_screen_status_bar_background>
<game_screen_status_bar_life>
<game_screen_status_bar_life_image_source><!-- String --></game_screen_status_bar_life_image_source>
<game_screen_status_bar_life_image_width><!-- INT --></game_screen_status_bar_life_image_width>
<game_screen_status_bar_life_image_height><!-- INT --></game_screen_status_bar_life_image_height>
<game_screen_status_bar_life_x_position><!-- INT --></game_screen_status_bar_life_x_position>
<game_screen_status_bar_life_y_position><!-- INT --></game_screen_status_bar_life_y_position>
<game_screen_status_bar_life_sound_source><!-- String --></game_screen_status_bar_life_sound_source>
<game_screen_status_bar_life_scoreboard>
<game_screen_status_bar_life_scoreboard_width><!-- INT --></game_screen_status_bar_life_scoreboard_width>
<game_screen_status_bar_life_scoreboard_height><!-- INT --></game_screen_status_bar_life_scoreboard_height>
<game_screen_status_bar_life_scoreboard_x_position><!-- INT --></game_screen_status_bar_life_scoreboard_x_position>
<game_screen_status_bar_life_scoreboard_y_position><!-- INT --></game_screen_status_bar_life_scoreboard_y_position>
<game_screen_status_bar_life_scoreboard_text_font_size><!-- INT --></game_screen_status_bar_life_scoreboard_text_font_size>
<game_screen_status_bar_life_scoreboard_text_font_color><!-- String --></game_screen_status_bar_life_scoreboard_text_font_color>
</game_screen_status_bar_life_scoreboard>
</game_screen_status_bar_life>
<game_screen_status_bar_points>
<game_screen_status_bar_points_image_source><!-- String --></game_screen_status_bar_points_image_source>
<game_screen_status_bar_points_image_width><!-- INT --></game_screen_status_bar_points_image_width>
<game_screen_status_bar_points_image_height><!-- INT --></game_screen_status_bar_points_image_height>
<game_screen_status_bar_points_x_position><!-- INT --></game_screen_status_bar_points_x_position>
<game_screen_status_bar_points_y_position><!-- INT --></game_screen_status_bar_points_y_position>
<game_screen_status_bar_points_sound_source><!-- String --></game_screen_status_bar_points_sound_source>
<game_screen_status_bar_points_scoreboard>
<game_screen_status_bar_points_scoreboard_width><!-- INT --></game_screen_status_bar_points_scoreboard_width>
<game_screen_status_bar_points_scoreboard_height><!-- INT --></game_screen_status_bar_points_scoreboard_height>
<game_screen_status_bar_points_scoreboard_x_position><!-- INT --></game_screen_status_bar_points_scoreboard_x_position>
<game_screen_status_bar_points_scoreboard_y_position><!-- INT --></game_screen_status_bar_points_scoreboard_y_position>
<game_screen_status_bar_points_scoreboard_text_font_size><!-- INT --></game_screen_status_bar_points_scoreboard_text_font_size>
<game_screen_status_bar_points_scoreboard_text_font_color><!-- String --
></game_screen_status_bar_points_scoreboard_text_font_color>
</game_screen_status_bar_points_scoreboard>
</game_screen_status_bar_points>
<game_screen_status_bar_meters>
<game_screen_status_bar_meters_image_source><!-- String --></game_screen_status_bar_meters_image_source>
<game_screen_status_bar_meters_image_width><!-- INT --></game_screen_status_bar_meters_image_width>
<game_screen_status_bar_meters_image_height><!-- INT --></game_screen_status_bar_meters_image_height>
<game_screen_status_bar_meters_x_position><!-- INT --></game_screen_status_bar_meters_x_position>
<game_screen_status_bar_meters_y_position><!-- INT --></game_screen_status_bar_meters_y_position>
<game_screen_status_bar_meters_sound_source><!-- String --></game_screen_status_bar_meters_sound_source>
<game_screen_status_bar_meters_scoreboard>
<game_screen_status_bar_meters_scoreboard_width><!-- INT --></game_screen_status_bar_meters_scoreboard_width>
<game_screen_status_bar_meters_scoreboard_height><!-- INT --></game_screen_status_bar_meters_scoreboard_height>
<game_screen_status_bar_meters_scoreboard_x_position><!-- INT --></game_screen_status_bar_meters_scoreboard_x_position>
<game_screen_status_bar_meters_scoreboard_y_position><!-- INT --></game_screen_status_bar_meters_scoreboard_y_position>
<game_screen_status_bar_meters_scoreboard_text_font_size><!-- INT --
></game_screen_status_bar_meters_scoreboard_text_font_size>
<game_screen_status_bar_meters_scoreboard_text_font_color><!-- String --
></game_screen_status_bar_meters_scoreboard_text_font_color>
</game_screen_status_bar_meters_scoreboard>
</game_screen_status_bar_meters>
</game_screen_status_bar>
</game_screen>

</screens>
<game_properties>
<game_properties_has_meters><!-- String {NO,YES} --></game_properties_has_meters>
<game_properties_has_points><!-- String {NO,YES} --></game_properties_has_points>
<game_properties_has_lives><!-- String {NO,YES} --></game_properties_has_lives>
<game_properties_has_main_character><!-- String {YES} --></game_properties_has_main_character>
<game_properties_has_control_user><!-- String {NO,YES} --></game_properties_has_control_user>
<game_properties_has_chasing_finger><!-- String {NO,YES} --></game_properties_has_chasing_finger>
<game_properties_has_target><!-- String {NO,YES} --></game_properties_has_target>
<game_properties_has_reach_to_end><!-- String {NO,YES} --></game_properties_has_reach_to_end>
</game_properties>
```

```
<levels>
	<level>
		<stage>
	<stage_description_image_source><!-- String --></stage_description_image_source>
	<stage_description_image_width><!-- INT --></stage_description_image_width>
	<stage_description_image_height><!-- INT --></stage_description_image_height>
	<stage_description_image_x_position><!-- INT --></stage_description_image_x_position>
	<stage_description_image_y_position><!-- INT --></stage_description_image_y_position>
	<stage_thumbnail_image_source><!-- String --></stage_thumbnail_image_source>
	 <stage_thumbnail_image_width><!-- INT --></stage_thumbnail_image_width>
	<stage_thumbnail_image_height><!-- INT --></stage_thumbnail_image_height>
	<stage_background_layer_back_image_source><!-- String --></stage_background_layer_back_image_source>
	<stage_background_layer_back_image_width><!-- INT --></stage_background_layer_back_image_width>
	<stage_background_layer_back_image_height><!-- INT --></stage_background_layer_back_image_height>
	<stage_background_layer_back_x_speed><!-- INT --></stage_background_layer_back_x_speed>
	<stage_background_layer_back_y_speed><!-- INT --></stage_background_layer_back_y_speed>
	<stage_background_layer_medium_image_source><!-- String --></stage_background_layer_medium_image_source>
	<stage_background_layer_medium_image_width><!-- INT --></stage_background_layer_medium_image_width>
	<stage_background_layer_medium_image_height><!-- INT --></stage_background_layer_medium_image_height>
	<stage_background_layer_medium_x_speed><!-- INT --></stage_background_layer_medium_x_speed>
	<stage_background_layer_medium_y_speed><!-- INT --></stage_background_layer_medium_y_speed>
	<stage_background_layer_front_image_source><!-- String --></stage_background_layer_front_image_source>
	 <stage_background_layer_front_image_width><!-- INT --></stage_background_layer_front_image_width>
	<stage_background_layer_front_image_height><!-- INT --></stage_background_layer_front_image_height>
	<stage_background_layer_front_x_speed><!-- INT --></stage_background_layer_front_x_speed>
	<stage_background_layer_front_y_speed><!-- INT --></stage_background_layer_front_y_speed>
	<stage_fps><!-- INT --></stage_fps>
	<stage_attemps><!-- INT --></stage_attemps>
	<stage_points><!-- INT --></stage_points>
	<stage_time_to_finish><!-- INT --></stage_time_to_finish>
	<stage_pixels_per_meter><!-- INT --></stage_pixels_per_meter>
	<stage_x_speed><!-- INT --></stage_x_speed>
	<stage_y_speed><!-- INT --></stage_y_speed>
	<stage_background_music_source><!-- String --></stage_background_music_source>
</stage>
<tops>
	<top>
		<top_y_position><!-- INT --></top_y_position>
	</top>
</tops>
<character>
	<character_name><!-- String --></character_name>
	<character_x_position><!-- INT --></character_x_position>
	<character_y_position><!-- INT --></character_y_position>
	<character_width><!-- INT --></character_width>
	<character_height><!-- INT --></character_height>
	<character_collision_area_left><!-- INT --></character_collision_area_left>
	<character_collision_area_top><!-- INT --></character_collision_area_top>
	<character_collision_area_right><!-- INT --></character_collision_area_right>
	<character_collision_area_down><!-- INT --></character_collision_area_down>
	<character_x_speed><!-- INT --></character_x_speed>
	<character_y_speed><!-- INT --></character_y_speed>
	<character_gravity_speed><!-- INT --></character_gravity_speed>
	<character_sound_die_source><!-- String --></character_sound_die_source>
	<character_sound_collision_bad_item_source><!-- String --></character_sound_collision_bad_item_source>
	<character_sound_collision_good_item_source><!-- String --></character_sound_collision_good_item_source>
	<character_movement_type><!-- INT --></character_movement_type>
	<character_gravity_type><!-- INT --></character_gravity_type>
	<character_has_chasing_finger><!-- String {YES,NO}--></character_has_chasing_finger>
	<character_has_propulsion><!-- String {YES,NO}--></character_has_propulsion>
	<character_x_max_speed><!-- INT --></character_x_max_speed>
	<character_x_min_speed><!-- INT --></character_x_min_speed>
	<character_y_max_speed><!-- INT --></character_y_max_speed>
	<character_y_min_speed><!-- INT --></character_y_min_speed>
	<character_has_gravity><!-- String {YES,NO}--></character_has_gravity>
	<character_sprite_default_source><!-- String --></character_sprite_default_source>
	<character_sprite_default_width><!-- INT --></character_sprite_default_width>
	<character_sprite_default_height><!-- INT --></character_sprite_default_height>
	<character_sprite_default_frames><!-- INT --></character_sprite_default_frames>
	<character_sprite_default_has_loop><!-- String {YES,NO} --></character_sprite_default_has_loop>
```

```
<character_sprite_left_source><!-- String --></character_sprite_left_source>
<character_sprite_left_width><!-- INT --></character_sprite_left_width>
<character_sprite_left_height><!-- INT --></character_sprite_left_height>
<character_sprite_left_frames><!-- INT --></character_sprite_left_frames>
<character_sprite_left_has_loop><!-- String {YES,NO} --></character_sprite_left_has_loop>

<character_sprite_right_source><!-- String --></character_sprite_right_source>
<character_sprite_right_width><!-- INT --></character_sprite_right_width>
<character_sprite_right_height><!-- INT --></character_sprite_right_height>
<character_sprite_right_frames><!-- INT --></character_sprite_right_frames>
<character_sprite_right_has_loop><!-- String {YES,NO} --></character_sprite_right_has_loop>

<character_sprite_up_source><!-- String --></character_sprite_up_source>
<character_sprite_up_width><!-- INT --></character_sprite_up_width>
<character_sprite_up_height><!-- INT --></character_sprite_up_height>
<character_sprite_up_frames><!-- INT --></character_sprite_up_frames>
<character_sprite_up_has_loop><!-- String {YES,NO} --></character_sprite_up_has_loop>

<character_sprite_down_source><!-- String --></character_sprite_down_source>
<character_sprite_down_width><!-- INT --></character_sprite_down_width>
<character_sprite_down_height><!-- INT --></character_sprite_down_height>
<character_sprite_down_frames><!-- INT --></character_sprite_down_frames>
<character_sprite_down_has_loop><!-- String {YES,NO} --></character_sprite_down_has_loop>

<character_sprite_collision_bad_item_source><!-- String --></character_sprite_collision_bad_item_source>
<character_sprite_collision_bad_item_width><!-- INT --></character_sprite_collision_bad_item_width>
<character_sprite_collision_bad_item_height><!-- INT --></character_sprite_collision_bad_item_height>
<character_sprite_collision_bad_item_frames><!-- INT --></character_sprite_collision_bad_item_frames>
<character_sprite_collision_bad_item_has_loop><!-- String {YES,NO} --></character_sprite_collision_bad_item_has_loop>

<character_sprite_collision_good_item_source><!-- String --></character_sprite_collision_good_item_source>
<character_sprite_collision_good_item_width><!-- INT --></character_sprite_collision_good_item_width>
<character_sprite_collision_good_item_height><!-- INT --></character_sprite_collision_good_item_height>
<character_sprite_collision_good_item_frames><!-- INT --></character_sprite_collision_good_item_frames>
<character_sprite_collision_good_item_has_loop><!-- String {YES,NO} --></character_sprite_collision_good_item_has_loop>

<character_sprite_die_item_source><!-- String --></character_sprite_die_item_source>
<character_sprite_die_item_width><!-- INT --></character_sprite_die_item_width>
<character_sprite_die_item_height><!-- INT --></character_sprite_die_item_height>
<character_sprite_die_item_frames><!-- INT --></character_sprite_die_item_frames>
<character_sprite_die_item_has_loop><!-- String {YES,NO} --></character_sprite_die_item_has_loop>

</character>
<npcs>
<npc>
<npc_name><!-- String --></npc_name>
<npc_x_position><!-- INT --></npc_x_position>
<npc_y_position><!-- INT --></npc_y_position>
<npc_width><!-- INT --></npc_width>
<npc_height><!-- INT --></npc_height>
<npc_collision_area_left><!-- INT --></npc_collision_area_left>
<npc_collision_area_top><!-- INT --></npc_collision_area_top>
<npc_collision_area_right><!-- INT --></npc_collision_area_right>
<npc_collision_area_down><!-- INT --></npc_collision_area_down>
<npc_x_speed><!-- INT --></npc_x_speed>
<npc_y_speed><!-- INT --></npc_y_speed>
<npc_is_avoidable><!-- String {YES,NO} --></npc_is_avoidable>
<npc_died_dissapear_time><!-- INT --></npc_died_dissapear_time>
<npc_give_points><!-- String {YES,NO} --></npc_give_points>
<npc_points><!-- INT --></npc_points>
<npc_sound_die_source><!-- String --></npc_sound_die_source>
<npc_sound_collision_bad_item_source><!-- String --></npc_sound_collision_bad_item_source>
<npc_sound_collision_good_item_source><!-- String --></npc_sound_collision_good_item_source>
<npc_movement_type><!-- INT --></npc_movement_type>
<npc_x_max_speed><!-- INT --></npc_x_max_speed>
<npc_x_min_speed><!-- INT --></npc_x_min_speed>
<npc_y_max_speed><!-- INT --></npc_y_max_speed>
<npc_y_min_speed><!-- INT --></npc_y_min_speed>
<npc_has_gravity><!-- String {YES,NO} --></npc_has_gravity>
<npc_gravity_type><!-- INT --></npc_gravity_type>
<npc_gravity_aceleration><!-- INT --></npc_gravity_aceleration>
```

```
            <npc_sprite_default_source><!-- String --></npc_sprite_default_source>
            <npc_sprite_default_width><!-- INT --></npc_sprite_default_width>
            <npc_sprite_default_height><!-- INT --></npc_sprite_default_height>
            <npc_sprite_default_frames><!-- INT --></npc_sprite_default_frames>
            <npc_sprite_default_has_loop><!-- String {YES,NO} --></npc_sprite_default_has_loop>

            <npc_sprite_left_source><!-- String --></npc_sprite_left_source>
            <npc_sprite_left_width><!-- INT --></npc_sprite_left_width>
            <npc_sprite_left_height><!-- INT --></npc_sprite_left_height>
            <npc_sprite_left_frames><!-- INT --></npc_sprite_left_frames>
            <npc_sprite_left_has_loop><!-- String {YES,NO} --></npc_sprite_left_has_loop>

            <npc_sprite_right_source><!-- String --></npc_sprite_right_source>
            <npc_sprite_right_width><!-- INT --></npc_sprite_right_width>
            <npc_sprite_right_height><!-- INT --></npc_sprite_right_height>
            <npc_sprite_right_frames><!-- INT --></npc_sprite_right_frames>
            <npc_sprite_right_has_loop><!-- String {YES,NO} --></npc_sprite_right_has_loop>

            <npc_sprite_up_source><!-- String --></npc_sprite_up_source>
            <npc_sprite_up_width><!-- INT --></npc_sprite_up_width>
            <npc_sprite_up_height><!-- INT --></npc_sprite_up_height>
            <npc_sprite_up_frames><!-- INT --></npc_sprite_up_frames>
            <npc_sprite_up_has_loop><!-- String {YES,NO} --></npc_sprite_up_has_loop>

            <npc_sprite_down_source><!-- String --></npc_sprite_down_source>
            <npc_sprite_down_width><!-- INT --></npc_sprite_down_width>
            <npc_sprite_down_height><!-- INT --></npc_sprite_down_height>
            <npc_sprite_down_frames><!-- INT --></npc_sprite_down_frames>
            <npc_sprite_down_has_loop><!-- String {YES,NO} --></npc_sprite_down_has_loop>

            <npc_sprite_collision_bad_item_source><!-- String --></npc_sprite_collision_bad_item_source>
            <npc_sprite_collision_bad_item_width><!-- INT --></npc_sprite_collision_bad_item_width>
            <npc_sprite_collision_bad_item_height><!-- INT --></npc_sprite_collision_bad_item_height>
            <npc_sprite_collision_bad_item_frames><!-- INT --></npc_sprite_collision_bad_item_frames>
            <npc_sprite_collision_bad_item_has_loop><!-- String {YES,NO} --></npc_sprite_collision_bad_item_has_loop>

            <npc_sprite_collision_good_item_source><!-- String --></npc_sprite_collision_good_item_source>
            <npc_sprite_collision_good_item_width><!-- INT --></npc_sprite_collision_good_item_width>
            <npc_sprite_collision_good_item_height><!-- INT --></npc_sprite_collision_good_item_height>
            <npc_sprite_collision_good_item_frames><!-- INT --></npc_sprite_collision_good_item_frames>
            <npc_sprite_collision_good_item_has_loop><!-- String {YES,NO} --></npc_sprite_collision_good_item_has_loop>

            <npc_sprite_die_item_source><!-- String --></npc_sprite_die_item_source>
            <npc_sprite_die_item_width><!-- INT --></npc_sprite_die_item_width>
            <npc_sprite_die_item_height><!-- INT --></npc_sprite_die_item_height>
            <npc_sprite_die_item_frames><!-- INT --></npc_sprite_die_item_frames>
            <npc_sprite_die_item_has_loop><!-- String {YES,NO} --></npc_sprite_die_item_has_loop>
</npc>
</npcs>
<target>
            <target_name><!-- String --></target_name>
            <target_x_position><!-- INT --></target_x_position>
            <target_y_position><!-- INT --></target_y_position>
            <target_width><!-- INT --></target_width>
            <target_height><!-- INT --></target_height>
            <target_collision_area_left><!-- INT --></target_collision_area_left>
            <target_collision_area_top><!-- INT --></target_collision_area_top>
            <target_collision_area_right><!-- INT --></target_collision_area_right>
            <target_collision_area_down><!-- INT --></target_collision_area_down>
            <target_sound_on_touched_source><!-- String --></target_sound_on_touched_source>
            <target_image_source><!-- String --></target_image_source>
            <target_image_width><!-- INT --></target_image_width>
            <target_image_height><!-- INT --></target_image_height>
            <target_image_frames><!-- INT --></target_image_frames>
            <target_image_has_loop><!-- String {YES,NO} --></target_image_has_loop>
</target>
</level>

</levels>
</game>
```