

Análisis de herramientas para la creación de videojuegos para no programadores.



Grado en Ingeniería Multimedia

Trabajo Fin de Grado

Autor:

Óscar Sánchez Blázquez

Tutor/es:

Carlos Villagrá Arnedo.
Francisco José Gallego Duran

Septiembre 2017



Universitat d'Alacant
Universidad de Alicante

Resumen

Dada el crecimiento exponencial del mundo de los videojuegos en los últimos años también han crecido el número de herramientas y programas que pretenden ser nuevos paradigmas de creación de videojuegos sin la necesidad de programación (al menos en un primer momento), por lo que cualquier usuario de cualquier nivel que desee usar alguna de ellas se encontrará en la disyuntiva de elegir una de ellas, ¿pero cuál?. En el presente trabajo se pretende dar una respuesta a esta pregunta mediante un exhaustivo análisis de los frameworks disponibles.

Mediante este análisis de los frameworks se va a comprobar su viabilidad como plataformas de desarrollos profesionales y estudiar las posibilidades de escalabilidad a todos los niveles.

También busca una comparación entre la programación con este tipo de frameworks y la programación clásica. Mediante la implementación del mismo juego en una herramienta de este tipo y un lenguaje de programación clásico como C++.

DEDICATORIA

CITA

Indice General

RESUMEN	3
1. JUSTIFICACIÓN Y OBJETIVOS.....	23
1.2. OBJETIVOS.....	23
1.2.1. <i>Objetivos Generales.....</i>	23
Objetivo General: Conocer los principales Frameworks de creación de videojuegos 2D para no programadores.....	23
Objetivo General: Analizar en profundidad los Frameworks y seleccionar un candidato adecuado para el estudio.....	23
1.2.2. <i>Objetivos Específicos.</i>	23
Comparación de juegos en programación clásica y con un framework.....	23
2. INTRODUCCIÓN.....	25
3. MARCO TEÓRICO.....	26
3.1. ESTUDIOS PREVIOS	27
3.1.1. <i>Dev Master.</i>	27
3.1.2. <i>Comparison and evaluation of 3D mobile game engines.....</i>	27
3.1.3. <i>Comparación de motores de videojuegos para la creación de juegos serios.</i>	28
3.2. CONCEPTOS PREVIOS.....	30
3.2.1. <i>Framework.</i>	30
3.2.2. <i>SDK (Kit de desarrollo).</i>	30
3.2.3. <i>Game Engine (Motor de videojuegos).</i>	31
3.2 CRITERIOS DE SELECCIÓN.....	33
Criterio 1: <i>Requisitos e instalación.....</i>	34
Medición de la puntuación: Instalación.....	34
Criterio 2: <i>Documentación, tutoriales y comunidad.</i>	37
Medición de la puntuación: Documentación, tutoriales y comunidad.....	38
Criterio 3: <i>Funcionamiento y dificultad de uso.</i>	39
Medición de la puntuación: Funcionamiento y dificultad de uso.....	40
Criterio 4: <i>Editor Gráfico/Visual.</i>	42
Medición de la puntuación: Editor Gráfico/Visual.....	42
Criterio 5: <i>Escalabilidad.</i>	45
Criterio 6: <i>Resultados profesionales.</i>	46
Criterio 7: <i>Estado actual.</i>	47
Criterio 8: <i>Sistemas operativos soportados y requerimientos.</i>	48
Medición puntuación posibilidades de exportación.....	48
Criterio 9: <i>Licencia.....</i>	50
Puntuación total.	51
3.2 PRIMERA SELECCIÓN DE FRAMEWORK.....	52
3.2.1. <i>Motores descartados y justificación.....</i>	53
Torque 2D.....	53
Unreal 4 2D.....	53
Cocos2D.....	54
GODOT Engine.....	54
Marmalade.....	55
Eclipse Game Engine.....	55
Unity 2D.....	55
Stencyl	56
Tabla resumen de programa descartados	56

3.2.2. <i>Motores preseleccionados</i>	56
Gdevelop	57
Clickteam Fusion.....	59
Construct 2	61
Game Maker.....	63
4. METODOLOGÍA.....	64
4.1. METODOLOGÍA DE TRABAJO EN GENERAL.....	64
4.2. PROCEDIMIENTO Y SELECCIÓN DE TAREAS.....	67
4.3. TUTORIZACIÓN DEL PROYECTO.	67
4.4. ORGANIZACIÓN Y VISIÓN GENERAL VÍA PROJECT.....	68
4.5. GESTIÓN DE RIESGOS DEL PROYECTO.	69
4.5.1. <i>Identificación de los riesgos</i>	69
4.5.2. <i>Análisis de riesgos</i>	71
4.5.3. <i>Planificación de prevención</i>	73
4.5.4. <i>Monitorización de riesgos</i>	76
5. CUERPO DEL TRABAJO	78
5.2. ANÁLISIS DE FRAMEWORKS.....	78
5.2.1 <i>Consideraciones previas</i>	78
Equipo de trabajo.	78
Versiones de los programas.....	78
Precio medio de las licencias y de los módulos de exportación.	78
5.2.2. <i>Análisis Gdevelop</i>	81
Criterio 1: Requisitos e instalación.	81
Puntuación subcriterio requisitos	82
Resumen proceso de instalación	84
Datos de instalación encuesta.	84
Puntuación subcriterio instalación.	85
Resumen puntuación requisitos e instalación.....	85
Criterio 2: Documentación, tutoriales y comunidad.....	86
Puntuación subcriterio documentación, foros y tutoriales oficiales.....	88
Tutoriales oficiales realizados.	89
Resumen de tutoriales realizados.....	93
Puntuación subcriterio análisis de tutoriales oficiales.....	94
Taringa	95
Puntuación subcriterio comunidades y foros de desarrolladores.	95
Resumen de puntuación Documentación, tutoriales y comunidad.....	96
Criterio 3: Funcionamiento y dificultad de uso.....	97
Puntuación subcriterio funcionamiento.	97
Puntuación primeros pasos: creación de un pequeño proyecto.	97
Funciones básicas	98
Resumen subcriterio funcionalidades.....	112
Puntuación subcriterio lógica de programación necesaria.....	112
Resumen puntuación funcionamiento y dificultad de uso	113
Criterio 4: Editor Gráfico/Visual.....	114
Criterio 5: Escalabilidad.	115
Puntuación subcriterio posibilidad de creación de funciones, scripts y/o lenguaje de programación propio.	115
Resumen puntuación Escalabilidad,	116
Criterio 6: Resultados profesionales.....	117
Puntuación subcriterio principales juegos desarrollados	117
Puntuación subcriterio diferenciación	118

Análisis de Herramientas para la creación de videojuegos | Óscar Sánchez

Resumen puntuación resultados profesionales.....	118
Criterio 7: Estado actual.	119
Resumen puntuación criterio Estado actual	120
Criterio 8: Sistemas operativos soportados y requerimientos.	121
Resumen puntuación sistemas operativos soportados y posibilidades de exportación.....	122
Criterio 9: Licencia.	123
Resumen puntuación licencias.	123
Puntuación final.....	124
5.2.3. <i>Análisis Construct 2.</i>	125
Criterio 1: Requisitos e instalación.	125
Puntuación subcriterio requisitos.....	125
Resumen proceso de instalación	126
Datos de instalación encuesta.	127
Puntuación subcriterio instalación.	127
Resumen puntuación requisitos e instalación	128
Criterio 2: Documentación, tutoriales y comunidad.....	129
Puntuación subcriterio documentación, foros y tutoriales oficiales.....	132
Tutoriales oficiales realizados.	132
Tutorial How to make a Platform game.	132
Puntuación subcriterio tutoriales realizados.....	137
Puntuación subcriterio análisis de tutoriales oficiales.....	138
Reddit	139
Taringa	139
Puntuación subcriterio comunidades y foros de desarrolladores.	139
Resumen de puntuación Documentación, tutoriales y comunidad.....	139
Criterio 3: Funcionamiento y dificultad de uso.....	141
Puntuación subcriterio funcionamiento.	141
Puntuación primeros pasos: creación de un pequeño proyecto.	141
Funciones básicas	142
Puntuación subcriterio lógica de programación necesaria.....	157
Resumen puntuación funcionamiento y dificultad de uso	158
Criterio 4: Editor Gráfico/Visual.....	158
Criterio 5: Escalabilidad.	160
Puntuación subcriterio posibilidad de creación de funciones, scripts y/o lenguaje de programación propio.	160
Puntuación subcriterio Add-ons y plug-ins	161
Resumen puntuación escalabilidad.	163
Criterio 6: Resultados profesionales.	164
Mortal Melon.....	165
Puntuación subcriterio principales juegos desarrollados	166
Puntuación subcriterio diferenciación.....	166
Resumen puntuación resultados profesionales.....	167
Criterio 7: Estado actual.	169
Resumen puntuación criterio Estado actual	169
Criterio 8: Sistemas operativos soportados y requerimientos.	170
Resumen puntuación sistemas operativos soportados y posibilidades de exportación.....	170
Criterio 9: Licencia.	172
Construct 2 Free edition.	172
Construct 2: Personal licence.....	173
Construct 2: Bussines.....	173
Puntuación de subcriterio precio de licencias	173
Puntuación de subcriterio modo gratuito.....	174
Puntuación de subcriterio precio de modulos de exportación.....	174
Resumen puntuación licencias.	175
Valoración final y resumen de puntuación	176

Análisis de Herramientas para la creación de videojuegos | Óscar Sánchez

5.2.4. Análisis de Click Team Fusion (<i>Lo tengo corregido en mail</i>)	177
Criterio 1: Requisitos e instalación.	177
Puntuación subcriterio requisitos.....	177
Resumen proceso de instalación	178
Datos de instalación encuesta.	179
Puntuación criterio requisitos e instalación	179
Criterio 2: Documentación, tutoriales y comunidad.....	181
Puntuación subcriterio documentación, foros y tutoriales oficiales.....	182
Tutoriales oficiales realizados.....	182
Catch The Fruit.....	182
Otros tutoriales.....	185
Resumen de tutoriales realizados.....	187
Puntuación subcriterio análisis de tutoriales oficiales.	187
Reddit	188
Taringa	188
Puntuación subcriterio comunidades y foros de desarrolladores.	188
Resumen de puntuación Documentación, tutoriales y comunidad.....	189
Criterio 3: Funcionamiento y dificultad de uso.....	190
Puntuación subcriterio funcionamiento.	190
Puntuación primeros pasos: creación de un pequeño proyecto.	190
Funciones básicas	191
Resumen de funciones básicas y puntuación del subcriterio.	205
Puntuación subcriterio lógica de programación necesaria	206
Resumen puntuación funcionamiento y dificultad de uso	207
Criterio 4: Editor Gráfico/Visual.....	208
Criterio 5: Escalabilidad.	209
Puntuación subcriterio posibilidad de creación de funciones, scripts y/o lenguaje de programación propio.	209
Resumen puntuación Escalabilidad.	211
Criterio 6: Resultados profesionales.....	213
Resumen puntuación resultados profesionales.....	217
Criterio 7: Estado actual.	218
Resumen puntuación criterio Estado actual	218
Criterio 8: Sistemas operativos soportados y requerimientos.	220
Resumen puntuación sistemas operativos soportados y posibilidades de exportación.....	220
Criterio 9: Licencia.	222
Puntuación subcriterio versión gratuito.	222
Puntuación de subcriterio precio de licencias	223
Puntuación de subcriterio precio módulo de exportación,	223
Resumen puntuación Licencias.....	224
Valoración final y resumen de puntuación	226
5.2.5. Análisis de Game Maker.....	227
Criterio 1: Requisitos e instalación.	227
Puntuación subcriterio requisitos.....	227
Resumen proceso de instalación	228
Datos de instalación encuesta.	229
Puntuación subcriterio instalación.	229
Puntuación criterio requisitos e instalación	230
Criterio 2: Documentación, tutoriales y comunidad.....	231
Puntuación subcriterio documentación, foros y tutoriales oficiales.....	236
Tutoriales oficiales realizados.....	237
Resumen de tutoriales realizados.....	241
Puntuación subcriterio análisis de tutoriales oficiales.	242
Puntuación subcriterio comunidades y foros de desarrolladores.	243
Resumen de puntuación Documentación, tutoriales y comunidad.....	243

Análisis de Herramientas para la creación de videojuegos | Óscar Sánchez

Criterio 3: Funcionamiento y dificultad de uso.....	245
Puntuación subcriterio funcionamiento.....	245
Puntuación subcriterio funcionamiento general	247
Puntuación subcriterio primeros pasos: creación de un pequeño proyecto.....	247
Funciones básicas	249
Resumen de funciones básicas y puntuación del subcriterio.	258
Puntuación subcriterio lógica de programación necesaria.....	259
Resumen puntuación funcionamiento y dificultad de uso	260
Criterio 4: Editor Gráfico/Visual.....	261
Criterio 5: Escalabilidad.	261
Puntuación subcriterio posibilidad de creación de funciones, scripts y/o lenguaje de programación propio.	261
Puntuación subcriterio Add-ons y plug-ins	263
Resumen puntuación Escalabilidad,	267
Criterio 6: Resultados profesionales.....	268
Puntuación subcriterio principales juegos comerciales desarrollados.....	272
Puntuación subcriterio diferenciación	273
Resumen puntuación resultados profesionales.....	274
Criterio 7: Estado actual.	275
Resumen puntuación criterio Estado actual	275
Criterio 8: Sistemas operativos soportados y requerimientos.	276
Resumen puntuación sistemas operativos soportados y posibilidades de exportación.....	276
Criterio 9: Licencia.	278
Puntuación subcriterio versión gratuito.	278
Puntuación de subcriterio precio módulo de exportación,	279
Resumen puntuación licencias.	280
Valoración final y resumen de puntuación	281
5.3. PROGRAMACIÓN EN GAME MAKER VS PROGRAMACIÓN C++ SFML.....	282
5.3.1. Razón de la compración.	282
5.3.2. Requisitos juego a desarrollar.	282
5.3.3. Desarrollo juego SFML.....	283
Estructura general del juego.....	283
Archivo Config.....	284
Resumen de juego en C++ y SFML.	310
5.3.4. Desarrollo juego Game Maker.	311
Room Main Menu.....	311
Objetos en Room Main menú.....	312
Room Game.....	312
Objetos en Room Game.....	312
Room Victory y room dead.	316
Resumen de juego en Game Maker.	316
6. ANÁLISIS DE RESULTADOS.....	317
6.1. RESULTADOS DE ANÁLISIS DE FRAMEWORKS DE CREACIÓN DE VIDEOJUEGOS.....	317
6.1.1 Análisis resultados Click Team Fusion.....	319
6.1.2 Análisis resultados Gdevelop.	322
6.1.3. Análisis resultados Construct 2	324
6.1.4. Análisis resultados Game Maker.	327
6.2 RESULTADOS COMPARACIÓN VIDEOJUEGO EN C++ VS VIDEOJUEGOS EN GAME MAKER.....	329
6.2.1 Resultados visuales.....	329
6.2.2 Rendimiento.....	330
6.2.3 Tiempo invertido.	331
6.2.4 Resumen final y conclusión.	332

ANEXO I: DATOS COMPLETOS ENCUESTA SOBRE FRAMEWORKS DE CREACIÓN DE VIDEOJUEGOS.	333
1. PROCESO DE INSTALACIÓN.....	341
1.1. <i>Datos instalación Gdevelop.</i>	341
1.2. <i>Datos instalación Construct 2.</i>	341
1.3. <i>Datos instalación CTF.</i>	342
1.4. <i>Datos instalación Game Maker.</i>	342
2. VALORACIÓN DE PROCESO DE INSTALACIÓN.	343
2.1. <i>Valoración del proceso de instalación de Gdevelop.</i>	343
2.2. <i>Valoración proceso de instalación Construct 2.</i>	344
2.3. <i>Valoración proceso de instalación Click Team Fusion.</i>	345
2.4. <i>Valoración proceso de instalación Game Maker.</i>	347
3. VALORACIÓN DE LA INTERFAZ.....	348
3.1. <i>Valoración interfaz de Gdevelop.</i>	348
3.2. <i>Valoración interfaz de Construct 2.</i>	348
3.3. <i>Valoración interfaz de Click Team Fusion.</i>	349
1.1. <i>Valoración interfaz de Game Maker.</i>	350
ANEXO II: INSTALACIÓN Y CONFIGURACIÓN DE SFML EN VISUAL STUDIO 2017.....	352
1. COMPILACIÓN DE SFML COMPATIBLE CON VISUAL STUDIO INSTALADO.	352
1.1. <i>Generar proyecto Visual Studio vía CMake.</i>	352
1.2. <i>Compilar solución generada en Visual Studio.</i>	355
2. CONFIGURAR PROYECTO VISUAL STUDIO.....	356
BIBLIOGRAFÍA	359

Indice de Figuras

Figura 1 : Wof3d Pimer motor de juegos 3D.....	32
Figura 2: Torque 2D.....	53
Figura 3: Unreal Engine.....	53
Figura 4: Cocos2D.....	54
Figura 5: GODOT Engine	54
Figura 6: Marmalade	55
Figura 7: Unity	55
Figura 8: Gdevelop.....	57
Figura 9: Sistema de eventos Click Team Fusion	59
Figura 10: Construct 2.....	61
Figura 11: Hoja de Eventos Construct 2	61
Figura 12: Game Maker	63
Figura 13: Cola de tareas por prioridades.....	66
Figura 14: Gdevapp.....	82
Figura 15: Gdevelop instalación de recursos accesorios	83
Figura 16: Portada de la wiki de Gdevelop	86
Figura 17: Portada de la wiki de Gdevelop	87
Figura 18:Portada del tutorial: How to make a platformer game?.....	89
Figura 19: Tutorial Gdevelop Begginers.....	91
Figura 20: Eventos puntuación	92
Figura 21: Destrucción del jugador	93
Figura 22: Gdevelop resumen tutoriales realizados	94
Figura 23: Gdevelop en Reddit.....	95
Figura 24: Gdevelop en Taringa.....	95
Figura 25: Gdevelop creación de un nuevo proyecto.....	98
Figura 26: Gdevelop: Variables movimiento en 2D.....	101
Figura 27: Gdevelop: variables movimiento en 8 direcciones.....	104
Figura 28: Gdevelop puntos de imagen	105
Figura 29: Gdevelop: creación de bala	108
Figura 30: Gdevelop colisión por píxel	110
Figura 31: Construct 2 Instalación	126
Figura 32: Construct 2 manual de usuario	130
Figura 33. Perfil oficial tutoriales Construct 2.....	131
Figura 34: Sección How do I , foro Construct 2	132
Figura 35: Resultado buscado tutorial How to make a Platform game	133
Figura 36: Funcionalidad enemigos destruyen a personaje	134
Figura 37: Nueva condiciones tutorial Construct 2	135
Figura 38: Gdevelop juego desarrollado tutorial 1.....	135
Figura 39: Sistema de eventos introducido para aumentar la dificultad.....	136
Figura 40: Eventos Gdevelop tutorial 2.....	137
Figura 42: Construct mensaje de error	141
Figura 43: Selector tipo de proyecto	143
Figura 44: Construct 2: selector de objetos	144
Figura 45: Construct 2: Movimiento sencillos 2 dimensiones	146

Figura 46: Construct 2 Jugador y box de colisión.....	147
Figura 47: Construct 2 acción "set postion to another object"	149
Figura 48: Movimiento 2d modo avanzado.....	150
Figura 49: Construct 2: Comportamiento tipo Bullet	153
Figura 50: Construct 2 disparo de bala tras pulsación	154
Figura 51: Construct 2 : posicionar origen de objeto para disparo	154
Figura 52:Construct 2: Evento y acción de destrucción de objeto tras colisión	156
Figura 53: Airscape	164
Figura 54: Mortal Melon.....	165
Figura 55: Mortal Melon datos	166
Figura 56: Eurgava.....	167
Figura 58: Posibilidades de exportación CrossWalk	173
Figura 59: CTF instalación de librerías y ejemplos.....	178
Figura 60: CTF tutorial Catch the fruit.....	183
Figura 61: Catch the fruit modificado	184
Figura 62: Tutorial Create an Artillery Game From Scratch.....	185
Figura 63: CTF Tutorial externo realizado.....	186
Figura 64: CTF vista de proyecto	192
Figura 65: CTF posicionar background	193
Figura 66: CTF Movimiento 2D.....	195
Figura 67: CTF Colision.....	197
Figura 68: CTF Salto 2D	198
Figura 69: CTF Variables del movimiento en 8 direcciones	200
Figura 70: CTF Cambio de en Sprite.....	201
Figura 71: CTF Disparo de bala.....	203
Figura 72: CTF uso de la función Random	206
Figura 73: Juego Five nights at Freddy´s	214
Figura 74: The Escapists.....	214
Figura 75: Freedom Planet	216
Figura 76: No a hero	217
Figura 77: Datos Not a hero	217
Figura 78: Proceso de instalación Game Maker.....	228
Figura 79: GM modo de actualización.....	228
Figura 80: GM instalación equipo de pruebas	229
Figura 81: Manual online GM	231
Figura 82: GM base de conocimiento	233
Figura 83: GM categorización tutoriales.....	234
Figura 84: GM:.Indice de tutoriales en el programa	235
Figura 85: GM: Tutorial en el programa	236
Figura 86: GM subcriterio documentación, foros y tutoriales oficiales	237
Figura 87: GM: Tutorial Scroll shooter	237
Figura 88: GM tutorial breakout	239
Figura 89: Resultado tutorial Breakdout	240
Figura 90: Gamer Maker lenguaje GML	245
Figura 91: GM: organización por carpetas	246

Figura 92: GM : Sistema de comentarios	247
Figura 93: GM: juego desarrollado naves	248
Figura 94: GM insertar fondo	251
Figura 95: GM: Evento colisión	257
Figura 96: Lenguaje de programación GML	262
Figura 97: Hotline Miami	268
Figura 98: Undertale	270
Figura 99: Spelunky	271
Figura 100: Nuclear Throne	272
Figura 101: Nuclear Throne datos	272
Figura 102: Gods will be watching	273
Figura 103: Space Invaders	282
Figura 104: Diagrama general juego C++	284
Figura 105: Constantes de config	284
Figura 106: Diagrama Clase Game	285
Figura 107: Diagrama Clase Player	290
Figura 108: Diagrama Hive Enemy	294
Figura 109: Diagrama EnemygShip	297
Figura 110: Diagrama clase bullet	301
Figura 111: Diagrama de la clase Build	305
Figura 112: Include necesario para listas	309
Figura 113: Listas: Definición y declaración	309
Figura 114: Recorrer lista de balas y usar método	309
Figura 115: Eliminar bala de la lista	310
Figura 116: Room Main Menu	311
Figura 117: Evento cambio de Room	312
Figura 118: Jugador movimiento y disparo	313
Figura 119: Creación de enemigos y edificios	314
Figura 120: Movimiento de enemigos	315
Figura 121: Eventos objeto edificio	316
Figura 122: Puntuación programas	318
Figura 123: Gráfico CTF criterios	319
Figura 124: Consideración criterios	319
Figura 125: Sistema de eventos CTF	321
Figura 126: Gdevelop criterios	322
Figura 127: Gdevelop clasificación criterios	322
Figura 128: Construct 2 puntuación criterios	324
Figura 129: Construct 2 consideración criterios	325
Figura 130: Game Maker puntuación criterios	327
Figura 131: Game maker clasificación criterios	327
Figura 132: Space Invasors C++	329
Figura 133: Space Invasors Game Maker	330
Figura 134: Configuración Cmake	352
Figura 135: Cmake SFML	353
Figura 136: Carpeta generada por Cmake	354

Figura 137: Proyecto Visual Studio.....	355
Figura 138: Depurador Visual Studio	355
Figura 139: Opciones de proyecto Visua Studio	356
Figura 140: Visual Studio archivos de inclusión.....	357
Figura 141: Visual Studio directorio de bibliotecas adicionales.....	357
Figura 142: Visual Studio dependencias adicionales	358
Figura 143: Código test SFML	358

Índice de tablas

Tabla 1: Dev Master	27
Tabla 2: Comparison and evaluation of 3D mobile game engines	28
Tabla 3: Comparación de motores de videojuegos para la creación de juegos serios..	29
Tabla 4: Clasificación de puntuaciones	33
Tabla 5: Criterio: Instalación y requisitos	34
Tabla 6: Ponderación tiempo de instalación.....	35
Tabla 7: Ponderación valoración del proceso de instalación.....	35
Tabla 8: Criterio: Documentación, tutoriales y comunidad.....	37
Tabla 9: Criterio: Funcionamiento general y curva de aprendizaje	39
Tabla 10: Criterio: Editor gráfico/visual	42
Tabla 11: Participantes en la encuesta	43
Tabla 12: Criterio: Escalabilidad	45
Tabla 13: Criterio: Resultados profesionales	46
Tabla 14: Criterio: Estado actual	47
Tabla 15: Criterio: Sistemas operativos soportados.....	48
Tabla 16: Sistemas de exportación a evaluar.....	49
Tabla 17: Criterio: Licencias.....	50
Tabla 18: Primera selección de motores	52
Tabla 19: Logo Torque 2D	53
Tabla 20: Unreal 4	53
Tabla 21: Exclusiones de Frameworks	56
Tabla 22: Gdevelop Datos	58
Tabla 23: Click Team Fusion Datos	60
Tabla 24: Construct 2 Datos	62
Tabla 25: Game Maker Datos.....	63
Tabla 26: Tipo de riesgos del proyecto.....	71
Tabla 27: Riesgos, probabilidades y efectos	73
Tabla 28: Planificación de riesgos	76
Tabla 29: Monitorización de riesgos	77
Tabla 30: Datos equipo de pruebas.....	78
Tabla 31: Precio medio de licencias.....	79
Tabla 32: Baremo puntuación precio licencias	79
Tabla 33: Precio medio módulos de exportación	79
Tabla 34: Baremo precio módulos de exportación.....	80
Tabla 35: Requisitos Gdevelop	81
Tabla 36: Gdevelop puntuación instalación.....	82
Tabla 37: Resumen proceso de instalación Gdevelop	84
Tabla 38: Gdevelop datos encuesta instalación.....	84
Tabla 39: Gdevelop subcriterio instalación.....	85
Tabla 40: Gdevelop criterio requisitos e instalación.....	85
Tabla 41: Gdevelop puntuación Documentación, foros y tutoriales oficiales	88
Tabla 42: Subcriterio Gdevelop Análisis de tutoriales oficiales	94
Tabla 43: Subcriterio foros y comunidades de desarrolladores	96
Tabla 44: Gdevelop criterio documentación, tutoriales y comunidad	96

Tabla 45: Gdevelop subcriterio funcionamiento general	97
Tabla 46: Gdevelop subcriterio funcionamiento general	97
Tabla 47: Gdevelop resumen de funcionalidades	111
Tabla 48: Gdevelop puntuación funciones básicas	112
Tabla 49: Gdevelop: subcriterio lógica de programación	112
Tabla 50: Gdevelop criterio funcionamiento y facilidad de uso.....	113
Tabla 51: Resultado encuestas sobre interfaz.....	114
Tabla 52: Criterio: Editor gráfico/visual	114
Tabla 53: Gdevelop: Subcriterio Posibilidad de creación de Scripts propios.....	115
Tabla 54: Gdevelop: Subcriterio Add-ons y plugins	116
Tabla 55: Gdevelop: puntuación criterio escalabilidad	116
Tabla 56: Gdevelop: subcriterio Principales juegos desarrollados.....	117
Tabla 57: Gdevelop subcriterio principales juegos desarrollados.....	118
Tabla 58: Trello desarrollo Gdevelop	119
Tabla 59: Gdevelop: estado y actualizaciones.....	119
Tabla 60: Gdevelop: estado actual.....	120
Tabla 61: Gdevelop puntuación exportación	122
Tabla 62: Gdevelop: puntuación licencias.....	123
Tabla 63: Gdevelop puntuación total.....	124
Tabla 64: Gdevelop Requisitos de instalación	125
Tabla 65: Construct 2: subcriterio requisitos.....	125
Tabla 66: Resumen proceso de instalación Gdevelop	127
Tabla 67: Gdevelop datos encuesta instalación.....	127
Tabla 68: Gdevelop subcriterio instalación.....	128
Tabla 69: Construct 2 criterio requisitos e instalación	128
Tabla 70: Construct 2 subcriterio ducumentación oficial.....	132
Tabla 71: Gdevelop resumen tutoriales realizados	138
Tabla 72: Subcriterio Gdevelop Análisis de tutoriales oficiales	139
Tabla 73: Subcriterio foros y comunidades de desarrolladores	139
Tabla 74: Gdevelop criterio documentación, tutoriales y comunidad	140
Tabla 75: Gdevelop subcriterio funcionamiento general	141
Tabla 76: Gdevelop subcriterio funcionamiento general	142
Tabla 77: Construct 2: puntuación funcionalidades	156
Tabla 78: Puntuación subcriterio funcionalidades.....	157
Tabla 79: Construct 2 Subcriterio lógica de programación	157
Tabla 80: Construct 2 criterio funcionamiento y dificultad de uso	158
Tabla 81: Resultado encuestas sobre interfaz Construct 2.....	159
Tabla 82: Criterio: Editor gráfico/visual	159
Tabla 83: Código Javascrip Plug-ing Keyboard.....	160
Tabla 84: Construct 2: Puntuación subcriterio creación de funciones propias.....	161
Tabla 85: Construct 2 addons store	162
Tabla 86: Puntuación subcriterio addons y plugins.....	163
Tabla 87: Construct 2 puntuación escalabilidad	163
Tabla 88: Airscape datos	165
Tabla 89: Construct 2 subcriterio diferenciación.....	167

Tabla 90: Puntuación criterio resultados profesionales Construct 2.....	168
Tabla 91: Posibilidades de exportación Construct 2	170
Tabla 92: Construct 2 puntuación sistemas exportación.....	171
Tabla 93: Construct 2 subcriterio precio de licencias	174
Tabla 94: Construct 2: subcriteiro modo gratuito.....	174
Tabla 95: Construct subcriterio precio módulos de exportación	175
Tabla 96: Contruct 2: puntuación criterio licencias.....	175
Tabla 97: Gdevelop Requisitos de instalación	177
Tabla 98: CTF subcriterio requisitos	177
Tabla 99: CTF Datos de instalación en el equipo de pruebas	179
Tabla 100: Gdevelop datos encuesta instalación	179
Tabla 101: CTF Subcriterio instalación.....	179
Tabla 102: CTF puntuación requisitos e instalación.....	180
Tabla 103: CTF subcriterio documentación, foros y tutoriales oficiales.....	182
Tabla 104: CTF Resumen tutoriales oficiales realizados.....	187
Tabla 105: Subcriterio Gdevelop Análisis de tutoriales oficiales	188
Tabla 106: CTF hilo Reedit	188
Tabla 107: CTF subcriterio foros y comunidad de desarrolladores	189
Tabla 108: CTF criterio documentación, tutoriales y comunidad	189
Tabla 109: CTF subcriterio funcionamiento general	190
Tabla 110: Gdevelop subcriterio funcionamiento general	191
Tabla 111: CTF resumen funcionalidades	205
Tabla 112: CTF subcriterio funcionalidades.....	205
Tabla 113: CTF subcriterio lógica de programación necesaria.....	206
Tabla 114: Puntuación criterio funcionamiento y dificultad de uso.....	207
Tabla 115: CTF Resultado encuestra interfaz.....	208
Tabla 116: Criterio: Editor gráfico/visual	208
Tabla 117: CTF Subcriterio funciones propias	209
Tabla 118: CTF Recursos.....	211
Tabla 119: CTF subcriterio addons y plugins	211
Tabla 120: Datos Five nights at Freddy´s.....	213
Tabla 121: Datos the Escapists	215
Tabla 122: Datos Freedom Planet	216
Tabla 123: CTF puntuación resultados profesionales.....	218
Tabla 124: CTF Estado	218
Tabla 125: CTF Criterio estado actual.....	219
Tabla 126: CTF Plataformas de exportación	220
Tabla 127: CTF puntuación sistema de exportación	220
Tabla 128: CTF subcriterio modo gratuito.....	222
Tabla 129: CTF subcriterio precio de licencias	223
Tabla 130: CTF precio módulos de exportación	224
Tabla 131: CTF Precio módulos de exportación y precio medio	224
Tabla 132: CTF Subcriterio Precio módulos de exportación	224
Tabla 133: CTF Criterio liencias.....	225
Tabla 134: CTF Puntuación Total.	226

Tabla 135: Game Maker Requisitos de instalación.....	227
T Tabla 136: Game Maker puntuación subriterio requisitos	227
Tabla 137: GM datos encuesta instalación.....	229
Tabla 138: GM subcriterio instalación	230
Tabla 139: GM puntuación requisitos e instalación.....	230
Tabla 140: GM Resumen toturiales oficiales realizados.....	241
Tabla 141:GM: Subcriterio Análisis de tutoriales oficiales	242
Tabla 142: GM Hilo Reedit	243
Tabla 143: GM: subcriterio foros y comunidades de desarrolladores.....	243
Tabla 144: GM: puntuación criterio documentación ,tutoriales y comunidad.	244
Tabla 145: GM subcriterio funcionamiento general	247
Tabla 146: GM subcriterio funcionamiento general	249
Tabla 147: GM resumen de funciones básicas	259
Tabla 148: GM subcreiterio funciones básicas.....	259
Tabla 149: Puntuación subcriterio lógica de programación necesaria.....	260
Tabla 150: GM puntaución criterio funcionamiento y dificulta de uso.....	260
Tabla 151: GM resultado encuesta interfaz.....	261
Tabla 152: Criterio: Editor gráfico/visual	261
Tabla 153: GML 2: Puntuación subcriterio creación de funciones propias.....	263
Tabla 154: GM extensiones.....	265
Tabla 155: GM: Puntuación subcriterio addons y plugins.....	265
Tabla 156: Hotline Miami datos.....	269
Tabla 157: Undertale datos	270
Tabla 158: Spelunky datos	271
Tabla 159: GM subcriterio principales juegos desarrollados	273
Tabla 160: GM subcriterio diferenciación	274
Tabla 161: GM criterio resultados profesionales.....	274
Tabla 162:GM estado actual	275
Tabla 163: GM puntuación estado actual.....	275
Tabla 164: GM sistemas de exportación	276
Tabla 165: GM criterio posibilidades de exportación.....	277
Tabla 166: GM subcriterio modo gratuito	279
Tabla 167: GM puntuación subcriterio precio de licencias.....	279
Tabla 168: GM precio módulos de exportación	279
Tabla 169: GM precio medio modulos con influencia en la puntuación.....	280
Tabla 170: GM puntuación licencias	280
Tabla 171: GM Puntuación final	281
Tabla 172: Variables clase Game	286
Tabla 173: Método game de la clase Game	287
Tabla 174:: Método run de la clase Game	287
Tabla 175: Método createBackground de la clase Game	287
Tabla 176: Método mainMenu de la clase Game	288
Tabla 177: Método gameMode de la clase Game	288
Tabla 178: Método deadMode de la clase Game	289
Tabla 179: Método victoryMode de la clase Game	289

Tabla 180: Variables clase Player	290
Tabla 181: Método player de la clase Player	291
Tabla 182 :Método update de la clase Player.....	292
Tabla 183: Método draw de la clase Player.....	292
Tabla 184: Método reduceHealt de la clase Player	293
Tabla 185: Método getAlive de la clase Player	293
Tabla 186: Variables clase HiveEnemy	294
Tabla 187: Método HiveEnemy de la clase HiveEnemy	295
Tabla 188 :Método update de la clase Hive Enemy	296
Tabla 189: Método draw de la clase HiveEnemy	296
Tabla 190: Método getTotalEnemy.....	296
Tabla 191: Variables clase EnemygyShip	297
Tabla 192: Método EnemygyShip de la clase EnemygyShip	298
Tabla 193 :Método update de la clase EnemygyShip.....	299
Tabla 194: Método draw de la clase EnemygyShip.....	299
Tabla 195: Método move de la clase EnemygyShip.....	300
Tabla 196: Método reduceHealt de la clase EnemygyShip	300
Tabla 197: Método getAlive de la clase EnemygyShip	300
Tabla 198: Método getSprite de la clase EnemygyShip	301
Tabla 199: Variables clase Bullet.....	302
Tabla 200: Método HiveEnemy de la clase Bullet	303
Tabla 201 :Método update de la clase Bullet	303
Tabla 202: Método draw de la clase Bullet	303
Tabla 203: Método getSprite de la clase Bullet	304
Tabla 204: Método getAlive de la clase Bullet.....	304
Tabla 205: Método getAlive de la clase Bullet.....	304
Tabla 206: Método getAlive de la clase Bullet.....	305
Tabla 207: Variables clase Build.....	306
Tabla 208: Método HiveEnemy de la clase Build	306
Tabla 209 :Método update de la clase Build	307
Tabla 210: Método draw de la clase Build	307
Tabla 211: Método reduceHealt de la clase Build	308
Tabla 212: Método getAlive de la clase Build	308
Tabla 213: Resumen juego C++	310
Tabla 214: Space Invasors Game Maker datos	316
Tabla 215: Consideración de criterios.....	317
Tabla 216: Rendimiento Space Invasor C++ y GM	330
Tabla 217: Tiempo invertido Space Invasors	331
Tabla 218: Tiempo instalación sujetos estudio.....	341
Tabla 219: Tiempo instalación sujetos Construct 2	342
Tabla 220: Datos instalación tiempo CTF.....	342
Tabla 221: Tiempo instalación GM encuestas	343
Tabla 222: Valoración sencillez proceso de instalación Gdevelop	344
Tabla 223: Valoración guia proceso de instalación Gdevelop	344
Tabla 224: Valoración sencillez proceso de instalación Construct 2.....	345

Tabla 225: Valoración guía proceso de instalación Constrcut 2	345
Tabla 226: Valoración sencillez proceso de instalación CTF	346
Tabla 227: Valoración guía en el proceso de instalación CTF.....	346
Tabla 228: Game Maker sencillez proceso de instalación encuestas.....	347
Tabla 229: Valoración guía proceso de instalación GM encuestas	347
Tabla 230: Valoración Interfaz Gdevelop	348
Tabla 231: Gdevelop resultado encuesta interfaz.....	349
Tabla 232: Valoración Interfaz CTF	350
Tabla 233: Valoración Interfaz GM encuestas	351

1. Justificación y Objetivos

1.2. Objetivos

1.2.1. Objetivos Generales

Objetivo General: Conocer los principales Frameworks de creación de videojuegos 2D para no programadores.

La principal motivación de este proyecto es analizar y conocer los Frameworks y SDK's que permiten la creación de videojuegos sin tener conocimientos a priori de lenguaje hombre-máquina.

Dada la gran popularidad que han experimentado los videojuegos en los últimos 20 años en el mercado existen una gran cantidad de programas que permiten (o al menos lo intentan) realizar esta función, se pretende encontrar los más utilizados o los que mejor consiguen realizar esta función prestando atención a los resultados profesionales que se pueden llegar a conseguir con ellos.

Conocer accesoriamente los juegos comerciales más populares que se han desarrollado con estas herramientas, así como los modelos de negocios de algunas de ellas.

Objetivo General: Analizar en profundidad los Frameworks y seleccionar un candidato adecuado para el estudio.

Una vez conocidos los SDK's y Frameworks de este tipo se va a seleccionar uno de ellos en función de su importancia en el mercado, de los resultados y se va analizar todos sus aspectos y funcionalidades.

Para ellos se van a comparar los distintos programas mediante unos criterios que se determinaran para que el seleccionado se ajústate a los objetivos del presente proyecto.

Objetivo General: Estudiar la viabilidad de grandes proyectos y la escalabilidad de estas herramientas.

Mediante el análisis en profundidad de las herramientas se pretende comprobar su viabilidad como plataformas para desarrollar grandes proyectos comerciales y/o profesionales.

Será importante comprobar las posibilidades de escalabilidad de las mismas y si existen proyectos profesionales realizados con las mismas.

1.2.2. Objetivos Específicos.

Comparación de juegos en programación clásica y con un framework.

Para tener una visión global de lo que este tipo de programas que la presente memoria intenta analizar se necesita comprar los resultados, consumo de recursos, tiempos,

curvas de aprendizaje o dificultad de uso entre otros aspectos en proyectos realizados con este tipo de software y la programación más clásica de videojuegos usando motores de programación pura como podría ser SFML o OGRE 3D.

Para conseguir esta comparación de ambas formas de crear juegos se va a realizar el mismo juego tanto en SFML como en el programa seleccionado y hacer una comparación de todos los elementos comentados anteriormente

2. Introducción.

2.1. ¿Por qué este trabajo?

Desde que comencé Ingeniería Multimedia noté una gran animadversión a los programas de creación de videojuegos como Game Maker, RPG Maker o Construct 2. Yo que vengo de una carrera de letras no entendía este sentimiento de rechazo a este tipo de herramientas.

Ahora con mis estudios casi finalizados y dedicándome profesionalmente a la ingeniería quiero ver hasta que punto ese rechazo es comprensible y responder a una serie de cuestiones sobre ella que siempre he tenido.

En los últimos tiempos estas herramientas han avanzado mucho, pero ¿se pueden considerar ya una alternativa a la programación clásica de videojuegos?, ¿Puede un usuario sin experiencia en programación a través del uso de estas herramientas convertirse en un desarrollador de videojuegos?, ¿Existen casos de éxito en el mundo de los videojuegos con herramientas de este tipo?

Estas son las preguntas fundamentales que he tenido siempre relativas a estos frameworks y ahora pretendo darles respuesta.

Durante la preparación del proyecto se ha visto que no existen apenas trabajos ni estudios acerca de las posibilidades de estas herramientas y las diferencias entre todas las opciones disponibles, por lo que parece que existen un vacío en el mundo académico que se pretende cubrir con este estudio.

Por último, quiero resaltar el carácter generalista del estudio, pese a estar realizado desde el punto de vista de un Ingeniero Multimedia se pretende que pueda ser consultado y usado por usuarios de todos los niveles y pueda suponer un punto de partida al uso de estas herramientas.

3. Marco teórico.

En este apartado del proyecto se pretende describir los conceptos fundamentales con los que se va trabajar, así que será el punto de partida para todos los desarrollos teóricos y prácticos posteriores. También sentará las bases para lo que se realizará en los siguientes puntos.

El marco teórico se divide en tres apartados:

1. En la primera parte se van a presentar tres estudios relacionados con el proyecto así los conceptos que se van a manejar durante el desarrollo del trabajo.
2. La parte central del marco teórico se dedicará a definir unos criterios de selección en base a los cuales se van a puntuar los **frameworks** que se analizarán en el cuerpo del trabajo.
3. Con los criterios ya definidos, se va realizar una primera selección de **frameworks**. En esta preselección se van a reducir todo lo posible el número de programas a analizar en el cuerpo del trabajo y excluir aquellos que tengan incompatibilidades con la finalidad del proyecto o impidan la consecución de los objetivos.

3.1. Estudios previos

3.1.1. Dev Master.

Dev Master es una base de conocimiento web de motores de videojuegos, herramientas, librerías accesorias, compañías y profesionales dedicados al sector de la programación de videojuegos. En ella se encuentran más de 350 motores.

Cada sección dedicada a un motor gráfico cuenta con todos sus datos (sitio web, propietarios, licencia y fecha de lanzamiento entre otros), sus datos técnicos (por ejemplo su lenguaje de programación, motores de físicas o shaders que usa), los libros escritos sobre el mismo relacionados y un apartado para que los usuarios y profesionales puedan dejar sus comentarios y análisis sobre el programa.

La principal diferencia con el estudio del presente proyecto radica en los siguientes aspectos fundamentales:

- Dev Master trata únicamente de motores gráficos y no también de frameworks.
- No compara los motores entre si como se pretende hacer en este proyecto, sino que se limita a listar todas sus características y opiniones de los usuarios.

Información sobre Dev Master	
Tipo	Proyecto Web
Año	2017
Autor/es	Colaborativo
Link	http://devmaster.net/

Tabla 1: Dev Master

3.1.2. Comparison and evaluation of 3D mobile game engines.

Se trata de una tesis realizada por Akekarat Pattrasitidecha para la universidad de Gotemburgo y que trata de encontrar el mejor programa para desarrollar videojuegos en tres dimensiones para dispositivos móviles.

Al buscar y comprar entre motores 3D centra la mayoría del trabajo en los aspectos técnicos de los motores de juego como las físicas 3D, los efectos de partículas o el video en 3D dejando los aspectos como la usabilidad y la facilidad de uso en un segundo plano al contrario que en el este proyecto. Además se limita únicamente a

motores de juego para dispositivos móviles y no abarca todos aquellos para escritorio y consolas como si se valora en este estudio.

Tipo	Tesis
Año	2014
Autor/es	Akekarat Pattrasitidecha
Link	http://publications.lib.chalmers.se/records/fulltext/193979/193979.pdf

Tabla 2: Comparison and evaluation of 3D mobile game engines

3.1.3. Comparación de motores de videojuegos para la creación de juegos serios.

Proyecto de fin de grado presentado por dos estudiantes de la Universidad Tecnológica de Pereira (Colombia), que trata de buscar y definir unos criterios tanto teóricos como tecnológicos que tiene que tener un de videojuego para poder desarrollar un videojuego profesional.

También tiene un apartado dedicado a pequeño análisis y comparación de motores de juego aunque el objeto de estudio solo son tres motores: JMonkey, Unreal Engine y Unity 5.

Por limitarse a solo el estudio de tres motores y estar centrado en definir los requisitos de un motor para crear un videojuego profesional, cuando en el caso del presente proyecto se desea abarcar proyectos de todo tipo, se diferencia bastante del este estudio.

Tipo	Tesis
Año	2015
Autor/es	Cristian Steven Chavarro Rico Mauricio Valencia Arbelaez
Link	http://repositorio.utp.edu.co/dspace/bitstream/handle/11059/6016/00669CH512.pdf?sequence=1

Tabla 3: Comparación de motores de videojuegos para la creación de juegos serios

3.2. Conceptos previos.

3.2.1. Framework.

Un **framework** en esencia es un patrón o esqueleto para el desarrollo de una aplicación, juego o contenido multimedia. Una definición más académica o formal podría ser que un **framewok** es un conjunto estandarizado de conceptos, prácticas y criterios para desarrollar y crear aplicaciones.

Los **frameworks** suelen incluir:

- Soporte de programas.
- Librerías.
- Lenguaje de scripting.
- Software accesorio para desarrollar partes específicas del programa.
- Un lenguaje de programación o interpretado.

Su uso implica un coste de aprendizaje aunque a largo plazo facilita el desarrollo, mantenimiento de la aplicación y permiten evitar trabajar en los detalles de bajo nivel permitiendo concretar esfuerzo en las partes esenciales de programa.

Desde el punto de vista del desarrollo de software, un **framework** es una estructura de soporte definida, en la cual otro proyecto de software puede ser organizado y desarrollado.

3.2.2. SDK (Kit de desarrollo).

Un **SDK o Kit de desarrollo** es un conjunto de herramientas y programas de desarrollo de software que permite al programador crear aplicaciones para un determinado paquete y estructura de software, plataforma de hardware, sistema operativo de computadora o máquina.

Un **SDK** no es más que una **API¹** (interfaz de aplicaciones) que permite el uso de uno o varios (aunque esta opción es menos habitual ya que suele trabajar en exclusividad con un lenguaje) lenguajes de programación.

Los **SDK** suelen llevar incluidas herramientas que ayudan al desarrollador a crear sus aplicaciones. Entre ellas destacan las siguientes:

¹ conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

- Debug.
- Soporte para la detección de errores de programación.
- Códigos de ejemplo y documentación.
- Entorno de programación **IDE**².

3.2.3. Game Engine (Motor de videojuegos).

El **Game Engine o Motor de videojuegos** proporciona un entorno de desarrollo integrado para la creación de juegos, no es más que un framework diseñado para la creación y desarrollo de juegos.

Un motor de juego suele incluir diferentes componentes que se corresponden con los elementos de un videojuego. Los elementos que suele incluir entre otros son los siguientes:

- **Programa del juego principal.**
- **Motor de render.**
- **Motor de audio.**
- **Motor de físicas.**
- **Motor de IA.**
- **Motor de Red.**

Antes de los Game Engine se escribían como entidades singulares y no reutilizables. El término motor de juego apareció a mediados de los 90 ligado al crecimiento de los juegos 3D especialmente shooters en primera persona, ya que permitía en juegos muy complejos técnicamente no comenzar desde cero en cada nuevo desarrollo.

² Aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software, como pueden ser editor de código fuente, herramientas de construcción automática o compilador.



Figura 1 : Wof3d Pimer motor de juegos 3D

3.2 Criterios de selección

Uno de los objetivos fundamentales del presente proyecto es analizar y buscar el mejor framework de creación de videojuegos para todo tipo de usuarios. Para seleccionar el programa se deben definir previamente unos criterios que permitan que esta elección sea lo más correcta posible ya que de ella depende en gran medida el éxito del proyecto.

Posteriormente a la determinación de estos criterios se le otorgará a cada uno de ellos una puntuación máxima y mínima. La suma de estas puntuaciones determinará cuál el mejor framework de carácter general.

Se va usar una escala de 100 para definir la puntuación ya que da más facilidades a la hora de visualizar las mismas y comparar los distintos programas entre si. No todos los criterios tienen la misma puntuación ya que no todos tienen la misma importancia, ni los mismos elementos.

En la siguiente tabla se encuentran definidas las distintas puntuaciones que pueden tener los elementos de cada criterio y su valoración en cuanto a su importancia y denominación.

Puntuación	Importancia	Explicación
0-1	Mínima	Elemento sin mucha importancia pero que hay tener en cuenta y valorarlo.
0-2	Poca	Apartado con cierta importancia pero que se desea que no influya demasiado en el resultado final
0-4	Normal	Los criterios en este apartado tienen una importancia standard, puntuación por defecto
0-6	Alta	Elementos bastante importantes y que deben contar mucho a la hora de seleccionar un programa
0-8	Clave	Elementos clave y fundamentales.
>10	Especial	Se detallará en cada caso si hay alguna puntuación de este tipo.

Tabla 4: Clasificación de puntuaciones

Criterio 1: Requisitos e instalación.

El primer criterio que hay tener en cuenta a la hora de analizar los diferentes motores es el requisito de funcionamiento e instalación.

En cuanto a los requisitos de instalación, dado el carácter generalista del presente proyecto, se valorará con mayor puntuación a aquellos programas En cuanto a los requisitos de instalación, dado el carácter generalista del presente proyecto, se valorará con mayor puntuación a aquellos programas que necesiten menos recursos del sistema para funcionar y que además funcionen en más sistemas operativos.

Es importante que el proceso de instalación sea tenido en cuenta, para comprobar la sencillez del mismo, la posible necesidad de instalación de otros programas o librerías accesorias y una buena guía al usuario durante todo el proceso. En este apartado será también considerado el tiempo que dura el proceso.

A continuación se muestran los apartados que compone este criterio y su importancia.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Requisitos de instalación.	Normal	0-4
Instalación.	Poca	0-2

Tabla 5: Criterio: Instalación y requisitos

Medición de la puntuación: Instalación.

El apartado instalación se medirá con la realización de encuestas en las que se solicitará a cada participante que mida el tiempo que tarda en todo el proceso de descarga e instalación del programa en su equipo y posteriormente se le hará las siguientes preguntas:

1. **Tiempo total de instalación.**
2. **Valoración de la complejidad del proceso:** de instalación en una escala de 10 donde 10 es sencillo y cero es complicado.
3. **Valoración del proceso de instalación:** Donde se va tratar de medir si los pasos en la instalación están bien explicados, donde 1 es que es poco o nada explicados y 10 que el programa guía al usuario de forma correcta y explica todos los pasos que va a realizar.
4. **Campo adicional: Librerías adicionales.** Donde el usuario deberá indicar si el programa le ha solicitado la instalación e alguna librería o programa adicional

para funcionar.

5. **Campo adicional: Fallo de instalación.** Si el programa falla durante la instalación el usuario deberá indicarlo en este campo e indicar el error aparecido para posteriormente intentar determinar que ha podido ocurrir y si esto puede restar puntuación al candidato.

Una vez se tiene la información recopilada de las encuestas, los dos puntos correspondientes a este apartado se van a distribuir de la siguiente forma. Un punto va a corresponder al tiempo de instalación medio y se va a ponderar según la siguiente tabla.

TIEMPO MEDIO DE ENCUESTAS	PUNTUACIÓN ASIGNADA
Tiempo menor de 5 minutos	1
Tiempo entre 5 y 7 minutos	0.66
Tiempo entre 7 y 9 minutos	0.33
Más de 9 minutos	0

Tabla 6: Ponderación tiempo de instalación

El otro punto del apartado se va otorgar realizando una media de los puntos 2 y 3 de la encuesta, que corresponden a la valoración de proceso y valoración de su complejidad. En la siguiente tabla se especifica que puntuación se otorgará en función de la media obtenida.

PUNTUACIÓN MEDIA	PUNTUACIÓN ASIGNADA
Más de 9	1
Entre 6 y 9	0.66
Entre 3 y 6	0.33
Menos de 3	0

Tabla 7: Ponderación valoración del proceso de instalación

Si se encuentran librerías adicionales y fallos en la instalación, puntos 4 y 5 de la encuesta, se valorará en cada caso la penalización a aplicar.

Criterio 2: Documentación, tutoriales y comunidad.

Como el presente proyecto pretende ser de alcance mayoritario, sin necesidad de conocimientos previos de ingeniería o programación, es fundamental que el programa elegido cuente con una buena documentación, tutoriales y una gran comunidad de usuarios.

En este apartado se incluyen toda la documentación oficial proporcionada por los propietarios del software y su soporte oficial, los tutoriales tanto los proporcionados de forma oficial como los realizados por usuarios en plataformas de video online, como You Tube o Vimeo, en blog especializados y las comunidades o foros de desarrolladores que trabajan con el programa.

El pilar fundamental de este criterio es el análisis de los tutoriales oficiales. Estos tutoriales son la primera aproximación de los usuarios al programa y por lo tanto deben contar con todos los elementos para que se puedan llevar a cabo de manera correcta, rápida y entendible. Se necesita un programa con unos tutoriales que aporten conocimiento e inciten a seguir aprendiendo y desarrollando habilidades, ya que se pretende que el presente proyecto sea un punto de partida a un aprendizaje en profundidad del programa y por lo tanto es necesario contar con los recursos necesarios.

Se otorgará puntuación extra a aquellos programas que tengan documentación, foros y tutoriales en castellano.

Los subapartados que componente este criterio son los listados en la siguiente tabla.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Documentación, foros y tutoriales oficiales	Normal	0-4
Análisis de los tutoriales oficiales	Clave	0-8
Comunidades y foros de desarrolladores.	Normal	0-2

Tabla 8: Criterio: Documentación, tutoriales y comunidad

Medición de la puntuación: Documentación, tutoriales y comunidad.

Al tratarse de un apartado clave en la elección del framework , hay que tener muy claro cómo se va medir la puntuación del mismo.

Respecto al subapartado **documentación, foros y tutoriales oficiales** se va otorgar la puntuación en función de los recursos aportados por los propios desarrolladores del programa o por terceros pero que sean considerados oficiales por los propietarios del framework. Se tendrán en cuenta todos los elementos puestos a disposiciones a disposición de los usuarios, independientemente de su naturaleza o formato aunque se presentará especial atención al manual oficial, los tutoriales y si cuentan o no con un foro oficial.

En subcriterio anterior se media la cantidad de recursos que se ponen a disposición de los usuarios de forma oficial, el subcriterio siguiente, **análisis de los tutoriales oficiales**, se trata de comprobar la calidad de estos recursos. Para medir este apartado se realizará uno o varios tutoriales oficiales y se analizará para comprobar si tienen o no errores, si su implementación es correcta, clara e entendible o si consiguen el objetivo de formar al usuario en el uso del programa e incitarla a continuar usando el programa.

Por último y para completar a los dos subapartados anteriores, nos encontramos con el subcriterio **comunidades y foros de desarrolladores no oficiales**. Para concretar la puntuación en este subapartado, se va comprobar la presencia de comunidades activas en Reddit³ , ya que es foro con mayor cantidad de usuarios a nivel mundial, y en Taringa, el equivalente en idioma español a Reddit⁴.

³ Reddit es un sitio web de marcadores sociales y agregador de noticias donde los usuarios pueden dejar enlaces a contenidos web. Otros usuarios pueden votar a favor o en contra de los enlaces, haciendo que aparezcan más o menos destacados. Se trata de un mapa de discusión, como parte de un DDS global distribuido.

⁴ Taringa! es una red social de origen argentino, de uso masivo en países de habla hispana.

Criterio 3: Funcionamiento y dificultad de uso.

El funcionamiento general del programa trata de detectar fallos técnicos tanto en la fase de instalación como posteriormente durante el uso del programa tales como errores que impiden la ejecución, el manejo o la realización de las distintas acciones y la estabilidad del programa.

Se puede decir que es una forma de medir si el programa se adecua a los estándares que un usuario puede comprender y usar de una manera rápida y efectiva consiguiendo sus propósitos funcionales.

La curva de aprendizaje es un apartado importante para decidir el programa va ser objeto de las acciones de este estudio ya que pretende ser un proyecto que en algunas partes del mismo, como puede ser las guías de aprendizaje puedan ser realizados posteriormente por usuarios de todos los niveles.

Por esta razón se busca un programa en el que la curva de aprendizaje no sea demasiado alta y pueda ser asumida por un usuario medio. Respecto a este criterio no se trata de la curva para aprender a dominar todos y cada uno de los aspectos del programa, sino de poder empezar a trabajar con el programa y crear pequeños proyectos, es decir, el foco está en el comienzo de la curva de aprendizaje.

Se busca por lo tanto un programa que sea muy sencillo en los primeros momentos y de adaptación muy rápida por parte de los usuarios sin importar tanto la dificultad de dominar todos los aspectos del mismo.

Los criterios que componentes este apartado de análisis son los mostrados en la tabla siguiente.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Funcionamiento General	NORMAL	0-4
Primeros Pasos: Creación de un pequeño proyecto.	NORMAL	0-4
Funciones básicas	OTROS	0-12
Lógica de programación necesaria	NORMAL	0-4

Tabla 9: Criterio: Funcionamiento general y curva de aprendizaje

Medición de la puntuación: Funcionamiento y dificultad de uso.

Dado que este criterio tiene un carácter puramente práctico se han de establecer las siguientes reglas o patrones para que la comparación entre los distintos programas sea lo más ecuánime posible, justa y adecuada para el propósito general del programa.

Detectar la curva de aprendizaje de un programa es una tarea bastante complicada de por sí, ya que entran parámetros subjetivos de la persona que está intentando identificar dicha curva de dificultad. Además en el caso que ocupa el presente trabajo presenta otra dificultad añadida, al tratarse de frameworks que tiene un funcionamiento en muchos apartados similar no se va apreciar la misma dificultad en el primer programa analizado que en los siguientes viendo ésta continuamente disminuida. Para mejorar ésta criterio que es posiblemente uno de los más importantes dentro de la comparación se van a realizar las siguientes acciones:

- Para cada programa se va realizar uno o varios tutoriales a ser posible oficial que consistan en la creación de un pequeño juego hasta completar aproximadamente 4 horas de trabajo con ellos. Documentando las dificultades que se van encontrando (en el caso que se trate de tutoriales oficiales los fallos o malas explicaciones en estos ya que afectan a otros criterios del estudio). Con lo que el apartado de "Creación de un pequeño proyecto" estará cubierto.
- Una vez completado este tiempo se estará en disposición de indicar una puntuación para los primeros pasos donde cero es la máxima dificultad y cuatro es muy sencillo.
- Con los conocimientos adquiridos en las horas dedicadas a la realización de un pequeño proyecto vía tutoriales se podrán definir como se hacen las funciones básicas de un juego y su dificultad, las funcionalidades básicas que se tendrán en cuenta son las siguientes: Movimiento de personaje por el escenario, mecánica básica de disparo, salto, colisión con otro objeto y desaparición de objeto colisionado.
- La puntuación del apartado "Creación de un pequeño proyecto" se asignará en función del nivel de complejidad conseguido el tiempo dedicado a la realización de los tutoriales oficiales, también influirá la cantidad de contenido creado o si han podido realizar diferentes tipos de juegos.
- A las puntuaciones del apartado **funciones básicas** se les va aplicar un factor corrector incremental que reducirá las puntuaciones en función de la posición en que se han realizado. **FACTOR multiplicador de 1 que se irá reduciendo en 0.05 en cada programa.** Estos frameworks pese a ser

distintos tienen bastantes aspectos en común en su funcionamiento por lo que la percepción de dificultad se reducirá conforme se vayan analizando programas, por lo que es fundamental contar con el factor de corrección para equilibrar las puntuaciones y que los programas analizados en los últimos lugares no se vean beneficiados por el conocimiento adquirido en el uso este tipo de herramientas.

Criterio 4: Editor Gráfico/Visual.

El contar con un editor visual para poder posicionar los objetos, personajes en el mundo y con el que poder crear relaciones entre ellos sin necesidad de *escribir código* y disponiendo solo de una edición gráfica es un punto esencial para seleccionar un framework u otro.

Se valorará no solo que tenga un editor gráfico sino que éste sea accesible, sencillo e intuitivo para los usuarios. Se tendrá en cuenta también el funcionamiento general, es decir, que no tenga bugs ni errores que impidan su funcionamiento o que dificulten el proceso de creación y desarrollo de los proyectos.

La usabilidad⁵ en el funcionamiento de la interfaz y de sus elementos también será un aspecto a tener en cuenta.

La tabla siguiente muestra la puntuación del presente apartado.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Editor/Gráfico visual	Clave	0-8

Tabla 10: Criterio: Editor gráfico/visual

Medición de la puntuación: Editor Gráfico/Visual.

Este criterio tiene en la mayoría de sus aspectos un carácter subjetivo, por lo que la mejor forma de valorarlo es a través de realización de encuestas, una por cada programa, más una extra para comparación.

La muestra que van a realizar el estudio la compondrán personas con conocimientos en distintos campos tanto de la ingeniería informática y multimedia, como del diseño y profesionales dedicados al sector tecnológico.

La composición profesional de la muestra es la que muestra la siguiente tabla:

⁵ La Usabilidad es la medida de la calidad de la experiencia que tiene un usuario cuando interactúa con un producto o sistema

Profesión/formación	Número de individuos.
Ingenieros Multimedia (Rama videojuegos)	2
Ingenieros Multimedia (Rama web)	2
Ingenieros Informáticos	1
Profesionales del sector tecnológico.	2
Arquitectos.	1

Tabla 11: Participantes en la encuesta

El proceso que se va a seguir para la valoración de estos criterios por la muestra escogida será el siguiente y tendrá las siguientes características:

1. El estudio tendrá en cuenta solo las encuestas de los usuarios que completen las 5 encuestas que componen el estudio, esto se hace para mantener una coherencia y cohesión en los datos, los mismos usuarios valoran distintas interfaces.
2. El proceso tendrá una duración aproximada de un mes enviándose a los participantes una encuesta cada semana.
3. Cada encuesta tendrá un encabezado para que el usuario se identifique con un nombre, apodo o de cualquier forma que una vez completado todo el estudio se pueda recolectar los resultados tanto del conjunto como de cada encuestado individualmente, además es requisito para cumplir el punto número 1.
4. En la primera de las encuestas se integrará un apartado para que el usuario indique las características del equipo en el que va a realizar las pruebas y que va ser el mismo para todas las pruebas.
5. Los apartados que los encuestados deben valorar para determinar la puntuación de cada programa tienen una puntuación de entre 1 y 5 siendo cinco el máximo.
6. Para el apartado **editor visual/intuitivo** se pedirá a los usuarios que valoren:
 -A priori parece un programa sencillo de utilizar.
 -Los iconos representan bien los elementos.

En el caso de valorar si el editor es visualmente atractivo, con un aspecto cuidado y actual los apartados a puntuar son los siguientes:

- La interfaz es atractiva.
- Tiene un aspecto cuidado y moderno.
- Puntúa el diseño del programa.

7. El buen funcionamiento del programa se medirá con los siguientes apartados:

- Rapidez al iniciarse.
 - Consumo de recursos del programa.
 - El programa se muestra rápido.

Existen dos apartados no obligatorios y no cuantificables, donde el usuario puede añadir información adicional que posteriormente si son completados habrá que valorar, son los siguientes apartados:

- Un apartado destinado a que el usuario indique si el programa no ha funcionado bien los errores que ha tenido. Si se produce algún caso de este tipo influirá negativamente en la valoración del funcionamiento general.
- Un campo para que el usuario valore con palabras la interfaz, que después habrá que interpretar.

Criterio 5: Escalabilidad.

El presente proyecto en su apartado de guía de uso pretende ser una iniciación al uso de un framework de videojuegos tanto para usuarios expertos en programación como para usuarios medios.

Pero los conocimientos adquiridos durante la realización del mismo deben servir tanto para realizar pequeños proyectos como los que aquí se proponen como para futuros juegos más grandes y ambiciosos. De esta forma el programa seleccionado debe tener una gran escalabilidad que permita afrontar problemas mayores y seguir progresando en cuanto a la complejidad de los proyectos creados.

Los dos apartados que componente este criterio son los mostrados en la siguiente tabla y su puntuación es la siguiente.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Posibilidad de crear funciones y scripts propios y/o lenguaje de programación propio.	Clave	0-8
Add-ons y plug-ins	Alta	0-6

Tabla 12: Criterio: Escalabilidad

Criterio 6: Resultados profesionales.

Los resultados que otros desarrolladores y/o creadores han conseguido con los diversos frameworks también es un criterio importante a la hora de seleccionar aquél sobre el que se va realizar las acciones del presente proyecto.

Elementos como la implantación de dichos motores en el mercado de juegos profesionales y los acabados gráficos y/o estéticos que se pueden conseguir con ellos son esenciales para que el motor sea adecuado para este estudio, ya que como se ha comentado pretende fomentar posteriormente un uso más profesional de dicha herramienta.

Un aspecto muy importante en este apartado es la diferenciación de los juegos profesionales o la no repetición de patrones visuales. Esto hace referencia a que los proyectos puedan ser tan diferentes tanto visualmente como funcionalmente de forma que no se parezcan entre ellos y se note la procedencia de los mismos. Es fundamental que el programa evite en la medida de lo posible que el usuario pueda detectar con un vistazo rápido con qué herramienta ha sido desarrollado el juego. Se habla aquí de la primera sensación y de carácter usuario ya que un desarrollador experto puede detectar el motor gráfico.

En la tabla mostrada a continuación se encuentran los subapartados de este criterio, su importancia y su puntuación.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Principales juegos desarrollados	Clave	0-8
Diferenciación	Normal	0-4

Tabla 13: Criterio: Resultados profesionales

Criterio 7: Estado actual.

A la hora de seleccionar un framework, se debe tener en cuenta el estado actual del mismo. Se trata de encontrar un framework que permita un uso prolongado en el tiempo y para ello debe estar activo, es decir, no abandonado por sus desarrolladores. Se valorará también la frecuencia de actualizaciones ya que permite detectar que los bugs y problemas que surgen en todo programa con su uso se van solucionando, además que estas nuevas versiones permiten incluir nuevas funcionalidades.

Los elementos que componen el estado actual son los siguientes:

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Estado actual	Normal	0-2
Frecuencia de actualizaciones.	Mínima	0-1
Fecha desde la última actualización.	Mínima	0-1

Tabla 14: Criterio: Estado actual

Criterio 8: Sistemas operativos soportados y requerimientos.

El carácter mayoritario y generalista ya comentado del presente trabajo, aconseja tener en cuenta para la selección de la herramienta de desarrollo los requisitos y requerimientos del sistema para hacer funcionar los juegos. Estos deben ser adecuados para llegar al mayor número posible de usuarios teniendo en cuenta tanto el hardware como el software y sistema operativo.

En la puntuación de este criterio tendrán una gran influencia positiva aquellos programas aquellos programas que posibiliten la exportación de sus juegos a un mayor número de dispositivos posibles. En este apartado no se valora si esta exportación a los diversos dispositivos y sistemas operativos es una función propia del programa base y viene incluida en su licencia principal o necesita módulos especiales de pago o gratuitos para llevarla a cabo, ya que esto ya será tenido en cuenta en el criterio de licencia.

La siguiente tabla muestra la importancia y puntuación de este criterio.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Posibilidades de exportación: Sistemas y plataformas	Clave	0-8

Tabla 15: Criterio: Sistemas operativos soportados

Medición puntuación posibilidades de exportación.

Se van a evaluar las posibilidades de exportación de los juegos de cada framework en 10 sistemas/plataformas, véase la tabla 16, por lo que se obtendrá una puntuación entre 10.

Para que un sistema o plataforma sea puntuado basta con que para uno de los subsistemas o plataformas sea posible realizar la exportación. Por ejemplo si el programa permite la exportación a Wii ,aunque no lo sea posible para Wiiu , su puntuación será completa en ese sistema general.

Posteriormente está se ponderará sobre la importancia del criterio, 8 puntos. Si bien pueden existir factores que aumenten o disminuyan esta puntuación que se tratarán y justificarán en cada caso.

SISTEMA/PLATAFORMA	INCLUYE
Windows	Sistemas operativos Widnows
Linux	Sistemas operativos basados en el kernel de Linux.
iOs	Para sistemas de escritorio y móvil.
Android	
HTML 5	
Flash	
Windows Store (Mobile y Store escritorio)	Móvil y tienda de escritorio.
Sistemas PlayStation (PS3, PS-VITA, PS4)	PS3, PS-Vita y PS4.
Sistemas Xbox (XBOX360, XBOX ONE)	Xbox 360. Xbox One.
Sistemas Nintendo (3DS, Wii, WiiU)	3ds , Wii, Wiiu.

Tabla 16: Sistemas de exportación a evaluar.

Criterio 9: Licencia.

El precio del software necesario y el tipo de licencia de cada programa a analizar también es un punto importante a la hora de seleccionar el mismo.

Al pretender ser un proyecto abierto y generalista tendrán mejor puntuación aquellas herramientas con un precio menor o aquellas que sus versiones gratuitas cuenten con unas menores limitaciones de uso.

Pese a ello, este criterio tiene menos importancia que otros ya que no se debe desvirtuar el objeto del estudio, ni que por seleccionar un software de menor precio o licencia con menos limitaciones no se llegue a la consecución de determinados objetivos o no se completen de manera satisfactoria.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Precio de Licencias	Normal	0-4
Modo de prueba o free	Poca	0-2
Precio módulos de exportación	Poca	0-2

Tabla 17: Criterio: Licencias

Puntuación total.

A continuación, se presenta la tabla con el resumen de las puntuaciones de todos los criterios.

CRITERIO	PUNTUACIÓN MIN-MAX
Requisitos e instalación	0-6
Documentación, tutoriales y comunidad	0-16
Funcionamiento y dificultad de uso	0-24
Editor Gráfico/Visual	0-8
Escalabilidad	0-14
Resultados profesionales	0-12
Estado actual	0-4
Sistemas operativos soportados y requerimientos.	0-8
Licencias	0-8
TOTAL	0-100

Ilustración 1: Puntuación Total

3.2 Primera selección de framework.

Antes de proceder a un análisis de los posibles candidatos, se ha realizado una primera lista de posibles motores, que pueden cumplir los criterios referidos en el punto anterior y ser el medio con el que se van a cumplir los objetivos del presente trabajo.

Esta primera lista parte de la búsqueda en general de motores gráficos y frameworks para desarrollar juegos 2D, sin entrar a valorarlos con los criterios anteriormente citados.

De ellos se descartaran automáticamente aquellos que posean características incompatibles con los objetivos planteados, aquellos que se vea sin necesidad de un estudio completo que no cumplen los requisitos buscados o aquellos que por otros motivos justificados no pueden ser una opción.

Los siguientes programas mostrado en la tabla son los que más se ajustan a priori al objeto de estudio del proyecto y por lo tanto se consideran a priori candidatos a ser el seleccionado para el mismo.

Click Team Fusion 2.5	Torque 2D
RPG-Maker	Gdevelop
Unreal 4 2D	Game Maker
Cocos2D	GODOT
Construct 2	Marmalade
Eclipse Origins	Unity 2D
Adventure Game Studio	Stencyl
Angel 2D	

Tabla 18: Primera selección de motores

3.2.1. Motores descartados y justificación

De la tabla mostrada anteriormente se han descartado una serie de programa a continuación se explican las razones de la exclusión de cada uno de ellos.

Torque 2D

Torque 2D o Torque Game Builder es la adaptación a 2D del motor gráfico Torque 3D creado por la compañía GarageGames para su juego Tribes 2 en 2001. Esta versión 2D del motor fue lanzada en 2013 bajo licencia MIT⁶ lo que le confiere un punto a favor al tratarse de software libre.

Pese a ser a contar con una licencia libre Torque 2D tiene la posibilidad de comprar una interfaz de desarrollo Torque Studio.

Después de una primera aproximación se ha podido comprobar que la instalación y configuración es demasiado compleja para el usuario medio, la creación de proyectos es excesivamente tediosa y es prácticamente indispensable contar con conocimientos de programación para poder crear contenido por lo que se ha descartado.



Figura 2: Torque 2D

Unreal 4 2D

Unreal 4 es la evolución del motor desarrollado en 1998 para el juego del mismo nombre y que bajo él y sus constantes evoluciones se han desarrollado juegos tan populares como Deus Ex, Bioshock, Batman: Arkham Asylum o Mass Effect.

Como pasara en su día que evolucionó desde un motor exclusivo de juegos en primera persona o shooters a abarcar también juegos tercera persona ahora en sus últimas versiones intenta también dar soporte a juegos 2D.



Figura 3: Unreal Engine

Aunque su potencia es muy grande y se pueden realizar grandes proyectos con él toda su interfaz está pensada para elementos tres dimensiones con lo que en algunos aspectos resulta bastante complejo el manejo de las dos dimensiones, además es un motor bastante exigente respecto a los requerimientos del sistema tanto para trabajar con el cómo en los proyectos resultantes.

⁶ Esta licencia es una Licencia de software libre permisiva lo que significa que impone muy pocas limitaciones en la reutilización y por tanto posee una excelente Compatibilidad de licencia

Cocos2D.

Cocos2D es un framework para diseñar y crear juegos 2D inicialmente basado en Phyton aunque diferentes versiones lo han adaptado a diferentes lenguajes de programación como por ejemplo la versión X o la versión Js que dan soporte a C++ y Javascript respectivamente.

Cocos2D se descarta en esta primera fase ya que aunque posee una creación rápida de proyectos, un manejo fácil y rápido de Sprites es imprescindible programar ya sea en Phyton con la versión original o en los lenguajes propios de otras versiones.

Cuenta además con la desventaja de no contar con una interfaz visual de manejo, creación y posicionamiento de los elementos, su funcionamiento es similar a motores como SFML, en los cuales tienes que llamar desde código a las funciones propias del mismo para realizar diversas acciones, pero añadiéndole su propio entorno de programación.



Figura 4: Cocos2D 2D

GODOT Engine.

A GODOT Engine es un motor de videojuegos multiplataforma y de código abierto. Su funcionamiento basado en nodos y escenas lo diferencia en gran medida de los motores y frameworks analizados en el presente proyecto.

Uno de los objetivos de este proyecto es que usuarios no programadores se acerquen al desarrollo de videojuegos. Puedan crear sus primeros proyectos, para posteriormente con la base adquirida poder aumentar su conocimiento, esto incluye poder utilizar cualquier otro motor o framework que no sea el seleccionado, por lo que uno que trabaje de forma tan distinta y diferencia del resto no es un buen candidato.

A parte de su funcionamiento diferente nos encontramos también en un caso como el de Cocos 2D, donde necesitamos nociones de programación para poder realizar proyectos en GODOT.

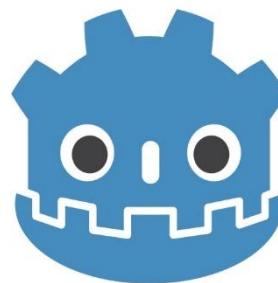


Figura 5: GODOT Engine

Marmalade.

Marmalade es un SDK y motor gráfico basado en C++. Es uno de los SDK's de desarrollado más utilizados por los grandes estudios para crear juegos 2D y 3D , juegos como Lara Croft and the Guardian of Light, Plants vs Zombies , Cut the Rope o el modo Zombies 2D de Call of Duty: World at War. Ha sido premiado en los Games Industry Awards 2015 en la categoría Engine and Middleware, Tools and Tech.



Figura 6: Marmalade

Pese al gran potencial de Marmalade y las cotas de calidad que pueden alcanzar sus juegos, su entorno basado en el lenguaje C++ que se debe conocer y dominar para crear contenido y su alta curva de aprendizaje hacen que sea descartado.

Eclipse Game Engine.

Eclipse Origins es un motor gráfico y una interfaz o suite para la creación de MMORPG⁷ 2D. Su uso es sencillo y su interfaz visual hace bastante sencillas las tareas de creación de escenarios y colocación de Sprites.

Su limitación a un solo género, la escasa documentación y comunidades activas unido a que es motor que prácticamente ha quedado en desuso hacen que no sea un buen candidato.

Unity 2D

Unity es uno de los motores de videojuegos más utilizados hoy en día tanto por grandes equipos como por pequeños estudios independientes. Su gran escalabilidad, gran interfaz visual, carácter multiplataforma y el potente sistema de scripts lo convierten un motor apto para cualquier nivel usuarios.



Figura 7: Unity

Pero el problema principal que hace que sea descartado es que Unity está pensado principalmente para desarrollo de juegos 3D y aunque sus últimas versiones incorporan la posibilidad de desarrollar juegos 2D todavía no tiene toda la potencia que sí tiene para juegos 3D. Aunque no se duda de que en futuras versiones sigan mejorando en ese sentido.

Al igual que con Unreal 2D, al ser un motor 3D adaptado a 2D, es pesado y consume excesivos recursos.

⁷ son videojuegos de rol que permiten a miles de jugadores introducirse en un mundo virtual de forma simultánea a través de internet e interactuar entre ellos.

Stencyl

Stencyl es un programa de creación de videojuegos 2D para computadoras y dispositivos móviles. Es un software gratuito con la opción de comprar módulos de pago.

Stencyl funciona dividiendo cada ciclo de ejecución en tres pasos: creación, dibujado y actualización de datos. Para las físicas hace uso del motor Box2D.

El principal inconveniente de Stencyl es que usa Flash para la exportación de sus juegos y las vulnerabilidades de Flash han hecho que la mayoría de navegadores hayan dejado de dar soporte a dicha tecnología. Aunque es conveniente seguir usando las aplicaciones en Flash incluyendo Flash Player en el sitio Web es conveniente tratar de usar otros programas que hagan uso de otras tecnologías.

Tabla resumen de programa descartados

La siguiente tabla muestra los motores descartados y las razones fundamentales para cada uno de ellos, algunos aparecen en varios motivos.

MOTIVO	CANDIDATO EXCLUIDO
Complejidad	Torque2D, Unreal 2D, Marmalade.
Alto consumo de recursos	Unreal 2D, Unity2D
Necesidad de conocimientos en programación	Cocos2D, GODOT Engine. Marmalade
Diferencia excesiva con el resto	GODOT Engine.
Limitaciones	Eclipse Game Engine.
Escasa documentación	Marmalade.
Motivos de Seguridad	Stencyl
Programa en desuso	Eclipse Game Engine, Stencyl

Tabla 21: Exclusiones de Frameworks

3.2.2. Motores preseleccionados.

Después de los programas excluidos han quedado candidatos posibles para el objeto de nuestro proyecto. Es a estos frameworks a los que se les hará el estudio completo basado en los criterios expuestos.

Gdevelop

Gdevelop es un software para la creación de juegos 2D y 3D, si necesitas de utilizar ningún lenguaje de programación. Gdevelop es un programa de código abierto y totalmente gratuito.



Figura 8: Gdevelop

El programa está escrito en C++ y trabaja fundamentalmente con la librería gráfica SFML , añadiéndole por encima una interfaz basada en WxWidgets⁸ y utiliza también TinyXML para la gestión de mapas y Tiles.

Gdevelop trabaja en base a varios editores que abarcan todo el proceso de creación de un videojuego. Los editores con los que cuenta Gdevelop son los siguientes:

- **Gestor del proyecto:** Es el editor principal del programa en el cual se indican los datos básicos del juego como pueden ser nombre o autor o pantallas de cargas. A través de este editor se accede al resto de editores. Funciona como entorno principal de trabajo.
- **Editor de escenas:** Permite organizar cada escena, pantalla o nivel del juego de manera individual. En él se realiza por ejemplo el posicionamiento de objetos o enemigos. Cuenta con un depurador para comprobar el correcto funcionamiento de la escena y un *player* que posibilita probar la escena.
- **Editor de objetos en la escena:** Posibilita la creación de los distintos tipos de objetos que pueden encontrarse en una escena. Permite crear distintos tipos de objetos como podrían ser Sprites, Textos u objetos 3D. En este editor se añaden todos los efectos tanto de físicas como de partículas o comportamientos predefinidos.
- **Editor de las capas de la escena:** Gdevelop puede trabajar con diferentes capas dentro de una escena, lo que ayuda a organizar y mejorar el trabajo pudiendo por ejemplo tener una capa para enemigos, otra para objetos de fondo, otros para objetos con los que poder colisionar,
- **Editor de eventos:** Gestionan los eventos de la escena y permiten realizar las funciones propias del juego. Este editor compone toda la capa lógica del juego y en la que se va definir toda la jugabilidad del mismo.
- **Banco de imágenes:** Gestiona todas las imágenes, videos y gráficos que se van a utilizar en un proyecto.

⁸ bibliotecas multiplataforma y libres, para el desarrollo de interfaces gráficas programadas en lenguaje C++

GDEVELOP	
Fecha de lanzamiento	2014
Programado en	C++
Lenguaje de programación	Javascript
Licencia	Libre

Tabla 22: Gdevelop Datos

Clickteam Fusion.

Click Team Fusion, apartir de ahora (CTF) es un framework para la creación de videojuegos especializada en juegos 2D lanzado en el año 2013 que actualiza la herramienta de creación y programación de juegos Multimedia Fusion que fue uno de los primeros programas que permitían crear juegos de manera visual aunque de una manera limitada.

Pero con la actualización de Multimedia Fusion a Click Team Fusion se añadieron toda una serie de herramientas que hicieron del programa una potente herramienta para crear videojuegos de manera rápida y visual. Algunas de estas características fueron por ejemplo la integración del motor de físicas Box2D que le daba a las mismas un acabado profesional, un nuevo sistema de objetos que hacía más sencilla la tarea de creación de los mismos y una nueva interfaz que llevaba la programación gráfica a un nuevo nivel no visto en las versiones anteriores.

Lo más destacable de CTF es su sistema de gestión de eventos, que constituye el núcleo central de todo el programa, en la Figura 9 se observa un ejemplo de eventos de CTF.

	All the events	All the objects																
1	• Start of Frame																	
2	• Start of Frame																	
3	Set Alterable Value A																	
4	• Always																	
5	Fade In																	
6	• Timer equals 01"-00																	
7	• : internal flag 0 is on																	
8	• Timer equals 03"-00																	
9	• : internal flag 0 is on																	
10	• Timer equals 05"-00																	
11	• : internal flag 0 is on																	
12	• Timer equals 07"-00																	
13	• : internal flag 0 is on																	
14	• Timer equals 09"-00																	
15	• : internal flag 0 is on																	
16	• Timer equals 11"-00																	
17	• : internal flag 0 is on																	
18	• Timer equals 14"-00																	
19	• New condition																	

Figura 9: Sistema de eventos Click Team Fusion

CLICK TEAM FUSION	
Fecha de lanzamiento	2012
Programado en	C, C++
Lenguaje de programación	C, C++, Propio
Licencia	Software Propietario

Tabla 23: Click Team Fusion Datos

Construct 2

La primera versión de Construct de código abierto fue creada por un grupo de estudiantes, cuya ultima build fue lanzada en 2012 dando por finalizado en ese momento el ciclo de vida del programa.

Construct 2 fue una evolución comercial del primer Construct, creada por el mismo grupo de desarrolladores ya constituidos en la empresa Scirra, y sacada al mercado en Febrero de 2011.

Pensando como un motor de creación de videojuegos para usuarios sin conocimiento de programación, está basado basando en dos aspectos claves: el drag and drop de elementos y en la otorgación de patrones de comportamiento de manera visual.



Figura 10: Construct 2

Su funcionamiento principal es a través de las hojas de comportamiento y/o evento, en las que a partir del cumplimiento de una serie de condiciones el programa ejecuta unas sentencias previamente definidas por el usuario. Además, a estas condiciones se les pueden añadir subcondiciones, o agregaciones para hacer todo lo complejo que uno quiera el comportamiento de los objetos del juego. En la Figura 11 se puede ver un ejemplo de una hoja de comportamiento de Construct 2.

Figura 11: Hoja de Eventos Construct 2

CONSTRUCT 2

Fecha de lanzamiento	2011
Programado en	C++, Javascript
Lenguaje de programación	Javascript
Licencia	Software Propietario

Tabla 24: Construct 2 Datos

Game Maker

Fundamentalmente divide su funcionamiento en dos grandes apartados, por un lado en el sistema de eventos y comportamientos, que se compone de hojas de eventos en la que cada hoja tiene una serie de eventos y sus sentencias condicionales y una vez que se cumpla se realizará el comportamiento asignado. Y por otro lado en evitar la selección de instancias mediante la selección de los objetos por medio del menú contextual que especifica el creador en las hojas de comportamiento.

Game Maker Studio, (a partir de ahora GM) es un software y kit de desarrollo para la creación de videojuegos creado en **Delphi**. Nació en el ámbito académico a finales de los noventa en Países Bajos. El profesor de la universidad de Utrecht, Mark Overmars, empezó a desarrollar una herramienta para ayudar a los estudiantes en la animación de personajes tanto de juegos como de otros ámbitos del dibujo, pero el proyecto fue creciendo y se convirtió en un software para que los estudiantes pudieran crear videojuegos en los primeros momentos de su carrera con pocos conocimientos de programación. En 1999 GM dejó de ser únicamente un programa para los alumnos de Overmars y fue lanzado al mercado.



Figura 12: Game Maker

La principal característica de Game Maker es que se intenta adaptar a todo tipo de usuarios, ya que permite realizar la implementación de las funcionalidades de dos formas completamente distintas: una desde su interfaz visual haciendo uso del Drag and Drop sobre objetos y otra mediante su lenguaje de programación GML.

En 2012 Yoyo Games (compañía propietaria de GM) lanzó en paralelo a la versión estándar de Game Maker otra versión llamada Game Maker: Studio, que incorporaba cambios profundos el programa tanto a nivel visual como a bajo nivel e incluía nuevas posibilidades de exportación a plataformas y sistemas operativos. En 2015 la versión Game Maker fue retirada por Yoyo Games y la rama Game Maker Studio se convirtió en la única versión del programa.

GAME MAKER

Fecha de lanzamiento	Noviembre 1999
Programado en	Delphi
Lenguaje de programación	GML (Propio)
Licencia	Software Propietario

Tabla 25: Game Maker Datos

4. Metodología

El presente proyecto cuenta con varios apartados bien diferenciados entre si donde algunas fases son de análisis e investigación y en otras es implementación y creación de contenidos, por lo tanto, es difícil nombrar una metodología que se adapte a todo el proyecto.

Para suplir este déficit se van a implementar diferentes metodologías o formas de trabajar para cada una de las partes que componen el trabajo. Si bien se contará con una metodología común para todo el proyecto para llevar la gestión de tiempos y de las funciones propias de la administración.

4.1. Metodología de trabajo en general.

Para las tareas de organización y administración del trabajo se va seguir una metodología por tiempos, prioridades e importancia con una **pila de tareas** que nos va indicar que tareas debemos realizar en cada momento y cuál es la siguiente a realizar. Al comienzo del proyecto se han delimitado todas las tareas que hay que realizar para completarlo, justo a esta delimitación cada una de ellas cuenta con su tiempo de aproximado de realización, su prioridad y su escala de importancia.

El tiempo aproximado de realización va medir el tiempo que debido a un análisis previo de la tarea en cuestión se ha determinado se va tardar en completar la misma. Este campo va servir para posteriormente comprobar si el análisis previo de los tiempos de las tareas se hizo correctamente, así como para realizar las mediciones de las desviaciones de tiempos de los distintos apartados del proyecto.

Estos tiempos no son tiempos fijos ya que al ser un trabajo en el a priori mucha de la tecnología que se va usar se desconoce dar un tiempo aproximado a la realización de las tareas sería tremadamente osado y daría lugar a desviaciones muy grandes, por lo que estas estimaciones se van a ir realizando conforme se tengan datos más o menos fiables de los tiempos que se va tardar en realizar dicha tarea. Por lo tanto, la tabla de tiempos va ser dinámica y va estar evolucionando junto con el proyecto.

La escala de importancia mide la importancia de la tarea dentro del proyecto. Da una visión de la tarea en relación a las demás y al proyecto. Esta escala sirve para delimitar las partes del proyecto y/o tareas que resultan fundamentales para el mismo y sin las cuales el mismo pierde razón de ser y aquellas que tienen carácter más accesorio o secundario. Dentro de esta escala de creación ad hoc para el proyecto se ha establecido los siguientes niveles:

1. **Imprescindible:** Con este grado de la escala indicamos aquellas tareas o partes del proyecto sin los cuales el mismo pierde su razón de ser, por lo tanto, prevalecen sobre todas las demás y si no se realizan el proyecto no puede continuar. Son las primeras que se deben desarrollar y sin ellas completadas el proyecto se considerará a todos los efectos inconcluso. La característica principal de este tipo de tareas y que sirve para identificarlas es que tienen

valor por si mismas pueden ser entendidas sin necesidad del resto.

2. **Importantes:** Las tareas dentro de esta escala son necesarias para que el mismo se pueda entender, sin ellas el trabajo carecerá en parte de sentido y quedará incompleto. Son casi tan importantes como las Imprescindibles pero necesitan de estas para ser entendidas.
3. **Secundarias:** Aquí se trata de elementos del proyecto que ayudan y complementan a las anteriores, pero sin ellas el proyecto puede ser entendido. Incluyen elementos que dan más sentido y refuerzan los análisis y/o desarrollos realizados en las tareas Imprescindible y Muy Importantes. Un ejemplo claro de este tipo de apartados podría ser el aumentar el número de motores gráficos a analizar, el número de ejemplos de juegos analizados o el número de funciones recurrentes del manual de uso de Game Maker.
4. **Accesorias:** Esta categoría trata de aquellas tareas que no son propias del proyecto, ni complementan a las anteriores, sino que son tareas que sirven a modo de anexos o complementos para dotar al conjunto de una serie de elementos a su alrededor. Por ejemplo, la inclusión de juegos o proyectos anexos de Game Maker o la realización de una recopilación accesoria de manuales, cursos y videos para aprender a programar con el tipo de programas que son objeto de estudio son tareas de este tipo.

CATEGORIA EXTRAORDINARIA: ADMINISTRATIVAS: Existen tareas que no forman parte del proyecto en sí, son partes administrativas, propias de la estructura o del formato seguido para la realización del mismo. Además el carácter institucional e educativo implica ciertas partes que hay cumplimentar necesariamente. Se trata de una categoría excepcional ya que esta fuera de las propias y/o accesorias del trabajo y no se puede establecer una jerarquía de realización con el resto. En general este tipo de tareas se deben realizar en el momento que se tengan los datos necesarios para realizarlas prevaleciendo sobre el resto.

La prioridad va dar información de la importancia de la tarea dentro de la escala de importancia del punto anterior en el momento actual del proyecto. Una tarea con prioridad alta va ser clave para que el trabajo pueda continuar y por lo tanto tiene prevalencia sobre otras tareas con menor prioridad (siempre que se encuentren dentro del mismo grado de la escala de importancia). Las prioridades en el proyecto se clasifican en alta, media y baja.

Cola de tareas por escala y prioridades: Para saber que tarea se va a realizar a continuación se utilizará una cola de tareas clasificadas por escala de importancia y prioridad en la que estas las tareas están ordenadas por orden que se van a realizar. Esta cola no es una cola inamovible y fija ya que puede ir modificándose con el paso del proyecto por los siguientes motivos:

1. Aparecen tareas con el tipo "Categoría extraordinaria: Administrativas" que como se ha comentado anteriormente se deben

realizar en el momento que se cuente con los elementos necesario para hacerlo por lo que en ese momento pasaran al principio de la cola.

2. La prioridad de una tarea no fue medida correctamente porque no se conocía en profundidad el campo de conocimiento que la componen o surgen elementos nuevos que cambian la importancia de la misma.
3. Surgen tareas nuevas no previstas y que se convierten en claves para poder continuar con otra tarea.
4. Aparecen subdivisiones de tareas ya existentes.

Cola de tareas por prioridades

Escala de Importancia	Prioridad	Tarea	Fecha de inicio	Fecha de finalización	% completado	Confirmación Tutor
CE-Administrativo		Objetivos	08/09/2016	27/09/2016	100%	
CE-Administrativo		Tipo de metodología	22/09/2016		80%	
CE-Administrativo		Técnicas y procedimiento	22/09/2016		0%	
Imprescindible	Alta	Estudio de Motores Gráficos			0%	
Imprescindible	Alta	Elección de motor GM			0%	

Figura 13: Cola de tareas por prioridades

En figura anterior podemos ver la Cola de tareas por prioridades y esta cuenta con los siguientes campos:

1. **Escala de importancia.**
2. **Prioridad dentro de la escala.**
3. **Nombre de la tarea.**
4. **Fecha de inicio.**
5. **Fecha de finalización.**
6. **Porcentaje de realización:** El porcentaje de realización va a servir para poder realizar en determinados momentos una revisión general del estado de las tareas, ya que aunque sea una cola en la que las tareas en la parte superior se deben realizar antes que las que están por debajo de la misma, esto no implica que no puedan realizarse varias tareas en paralelo o que algunas de ellas se encuentren en estado bloqueado o la espera de algún elemento o recurso necesario para continuar con ella.

4.2. Procedimiento y selección de tareas.

Como hemos comentado en el punto anterior la metodología del trabajo de la parte general y administrativa va funcionar mediante una cola de tareas basada en una escala de importancia y una prioridad. Para determinar la posición en esta cola el proceso que se debe de seguir es el siguiente:

1. **Determinar la escala de la tarea:** Mediante un estudio previo del proyecto y de las tareas se debe clasificar todas las tareas del proyecto dentro de la escala.
2. **Determinar prioridad de la tarea:** Una vez que las tareas cuente con su escala se determina para cada una de ellas su prioridad.
3. **Posicionar las tareas:** Para posicionar las tareas tenemos que tener en cuenta en primer lugar la escala de importancia, cuánto más alta sea la tarea se posicionará más arriba dentro de la cola. Cuando encontramos , tareas que se encuentran dentro del mismo grupo de la escala, atenderemos a la prioridad para determinar cuál se realizan primero. Pero existe una excepción a esta regla general de posicionamiento, esta excepción se aplica a las tareas de la categoría extraordinaria administrativas, que como se ha explicado anteriormente son propias del formato y del carácter del proyecto, que en cuanto las circunstancias permitan realizarlas pasaran automáticamente a la primeras posiciones de la cola y tendrán que realizarse en primer lugar.

4.3. Tutorización del proyecto.

El presente proyecto va contar con una supervisión y control por parte de un tutor. La función del tutor son las de ser un apoyo al que poder acudir en caso de dudas respecto al proyecto, su organización o la redacción del presente documento.

El tutor tiene acceso en todo momento a todo el material de trabajo actualizado para poder visualizar y supervisarlo si así lo desea. Además, se mantiene una comunicación constante y fluida con el mismo para que sepa en la situación que se encuentra el proyecto y si surgen problemas durante el desarrollo.

Toda esta labor de tutorización vía comunicación email y con acceso a disposición de todo el material de desarrollo se complementa con reuniones presenciales constantes. En general se mantendrán reuniones cada dos semanas para supervisar los avances y para detectar posibles desviaciones en la planificación del proyecto. Pero en épocas de mayor carga de trabajo o si surgen problemas puntuales se podrá aumentar la frecuencia de la misma o mantener reuniones excepcionales.

4.4. Organización y visión general vía Project.

Para tener una visión de todo el proyecto e ir llevando un seguimiento de todos los aspectos del proyecto es necesario contar con alguna herramienta que permita visualizarlo en su conjunto de una manera rápida y clara el estado que se encuentran las tareas, cuales están terminadas, retrasadas o bloqueadas y la gestión de su tiempo. Para esta función se utiliza el software Microsoft Project *DEFINICION.

Escala de Importancia	Prioridad	Nombre de tarea	Duración	Tiempo estimado	Tiempo empleado	Diferencial de tiempo	Comienzo	Fin	Predecesoras
Secundaria	Baja	Citas		0	0	0			
Administrativa		Indice		0	0	0			
		▫ Cuerpo	106 días	0	0	0	0 mié 07/09/16	mié 01/02/17	
		▫ Introducción	106 días	0	0	0	0 mié 07/09/16	mié 01/02/17	
Administrativa		Importancia de la temática		1	0	1			
Administrativa		Problemas a investigar		1	0	1			
Administrativa		Propósito y finalidad		1	0	1			
		▫ Marco teórico	106 días	0	0	0	0 mié 07/09/16	mié 01/02/17	
		▫ Elementos conceptuales	106 días	0	0	0	0 mié 07/09/16	mié 01/02/17	
Importante	Media	Biblioteca: Libray		0	0	0			
Importante	Media	Framework		0	0	0			
Importante	Media	SDK		0	0	0			
Imprescindible	Alta	▫ SDK'S y Frameworks para el desarrollo de juegos	106 días	0	0	0	0 mié 07/09/16	mié 01/02/17	
Imprescindible	Alta	▫ GameMaker	106 días	0	0	0	0 mié 07/09/16	mié 01/02/17	
Imprescindible	Alta	¿Qué es?		2	0,5	1,5			
Importante	Media	Software y licencias		6	1,15	4,85			
Importante	Alta	Juegos desarrollados con Game Maker		6	0,5	5,5			
Imprescindible	Media	Funcionamiento		0	0	0			
Secundaria	Media	GML: Lenguaje de Programación propio		0	0	0			
Importante	Media	RPG Maker		0	0	0			
Importante	Media	Cocos2D		0	0	0			
Accesoria	Media	Otros		0	0	0			
Secundaria	Media	Estudios previos		0	0	0			
Importante	Alta	Fundamentos teóricos		0	0	0			
		▫ Objetivos	19 días	0	0	0	0 mié 07/09/16	lun 03/10/16	
		▫ O. General	19 días	0	0	0	0 mié 07/09/16	lun 03/10/16	

Con Project y el sistema de campos que se ha aplicado se puede apreciar las tareas principales y las pequeñas tareas dependientes de estas. Además fruto de la metodología que con la que se va a llevar a cabo el proyecto se han añadido una serie de campos que aumentan la información y que son necesarios para tener esa visión global que con esta herramienta se persigue conseguir.

Campos de Project:

1. Escala de Importancia.

2. Prioridad.

3. Nombre de la tarea.

4. Duración.

5. Tiempo estimado: Conforme se disponga de la información necesaria para realizar una aproximación del tiempo que se va a necesitar para desarrollar una

determinada tarea se incluirá en el documento.

6. **Tiempo empleado:** El tiempo que se ha tardado en completar la tarea. Este tiempo se va a medir gracias a la herramienta Toggl, que permite una cuantificación exacta del tiempo de realización de cada tarea.
7. **Diferencia de tiempo:** Muestra la diferencia entre el tiempo inicialmente previsto y el que posteriormente se ha tardado en desarrollar una tarea, siendo los valores los que indican que se ha tardado más de lo que inicialmente se ha previsto y los negativos a la inversa, que se ha tardado menos de lo que en la fase de estimación se pensó.
Este valor va permitir una vez terminado el proyecto medir lo correcta que fue la planificación y calcular diferentes índices que ayuden a entender cómo ha ido el proceso.

8. Comienzo.

9. Fin.

10. **Predecesora:** Indica si alguna tarea necesita ser dependiente directa de otra, es decir, que necesita que otra esté completa para poder empezar realizarse, si así fuera en este campo aparecería la tarea en cuestión.

4.5. Gestión de riesgos del proyecto.

4.5.1. Identificación de los riesgos.

En este apartado se va identificar los riesgos potenciales del proyecto que pueden llevar a la no consecución de los objetivos planteados.

Los riesgos potenciales pueden ser de los siguientes tipos o categorías:

1. **Herramientas:** Se encuentran en este tipo de riesgos aquello que afectan a los programas que van a dar apoyo al desarrollo durante el proyecto, se refiere a aquellos que se van a utilizar para la gestión administrativa del mismo, así como de los soportes físico o digitales donde se encuentran almacenados los elementos del proyecto.
2. **Personal:** Todos aquellos riesgos que afectan no al proyecto en sí mismo ni a ninguna de sus partes, sino a la persona que lo está realizando y que provienen de situaciones externas al desarrollo del proyecto.
3. **Requerimientos:** Los riesgos que afectan a los requerimientos hacen referencia a aquellos elementos previos, o relativos a la parte anterior del desarrollo en que se determinó cuáles iban a ser las funcionalidades, objetivos y finalidad del proyecto. También al resultado final del proyecto respecto a estos requerimientos.

4. **Estimación:** Todos aquellos riesgos relativos a los tiempos y a la planificación temporal son riesgos del tipo estimación.
5. **Organizacional:** Hacen referencia a la organización del trabajo en sí mismo, así como las relaciones entre los distintos agentes implicados de cualquier tipo de relación con el proyecto.
6. **Tecnología:** Hardware y software que se va a utilizar durante el desarrollo del proyecto. En este apartado entrarían todos aquellos riesgos que se refieren a los frameworks y programas utilizados para la creación de los contenidos.

TIPO DE RIESGO	RIESGO
Tecnología	Desconocimiento de algunas de las herramientas a usar
Personal	Baja por enfermedad
Herramientas	Fallos de GIT
Personal	Mala comunicación con el tutor
Requerimientos	Se proponen cambios en los requerimientos o en los objetivos que suponen rehacer partes del proyecto
Requerimientos	Objetivos o tareas poco claras
Estimación	Estimación de costes errónea
Requerimientos	Coherencia y cohesión de las diferentes partes del proyecto
Requerimientos	Falta de seguimiento continuado y permanente de tareas
Herramientas	Pérdida de documentación o archivos importantes
Herramientas	Rotura de equipos o programas
Estimación	Aparición de trabajos no programados
Requerimientos	Demoras en la toma de decisiones ante la necesidad de cambio de requerimientos
Personal	Baja moral, desmotivación

Requerimientos	Mala elección del motor gráfico sobre el que se va realizar la implementación
-----------------------	---

Tabla 26: Tipo de riesgos del proyecto

4.5.2. Análisis de riesgos.

Una parte importante de los posibles riesgos es determinar la probabilidad de aparición de cada uno de ellos y sus posibles efectos.

Las probabilidades de aparición de los riesgos se clasifican en:

1. **Muy altas:** Cuando la probabilidad de que el suceso de dicho riesgo ocurra excede el 75%.
2. **Alta:** Las posibilidades de que ocurra el riesgo están comprendidas entre un 50 y un 75%.
3. **Moderada:** El suceso tiene una posibilidad de ocurrir entre un 25 y un 50%.
4. **Baja:** Los riesgos con probabilidades bajas tiene entre un 10% y un 25% de posibilidades de que ocurran.
5. **Muy bajas:** Cuando la posibilidad de que el riesgo se cumpla es menor del 10%.

Los efectos están agrupados en las siguientes categorías:

1. **Insignificantes:** Riesgo que puede ser soportados sin afectar a la consecución de los objetivos finales planteados, bien porque su resolución es viable en poco tiempo bien porque su aparición ésta prevista dentro de los tiempos de desarrollo de las diferentes actividades.
2. **Tolerables:** Los efectos moderados provocan un retraso asumible en los tiempos del proyecto, siempre y cuando no sucedan muchos incidencias con dichos efectos, ya que la suma de varias de ellas pueden llevar a la no consecución de objetivos propuestos.
3. **Serio:** Los riesgos potencialmente serios comprometen seriamente la realización de las metas buscadas e impiden en gran medida completar el proyecto en tiempo y forma.
4. **Catastrófico:** Los riesgos con efectos catastróficos de producirse tendrán un efecto muy negativo en la consecución del proyecto y sería difícil cumplir los plazos y objetivos si esta circunstancia llega a producirse.

RIESGO	PROBABILIDAD	EFFECTOS
Baja moral, desmotivación	Muy baja	Catastrófico
Mala elección del motor gráfico sobre el que se va realizar la implementación	Media	Serio
Rotura de equipos o programas	Media	Serio
Demoras en una toma de decisiones ante la necesidad de cambio de requerimientos	Media	Serio
Baja por enfermedad	Baja	Serio
Mala comunicación con el tutor	Baja	Serio
Objetivos o tareas poco claras	Baja	Serio
Coherencia y cohesión de las diferentes partes del proyecto	Baja	Serio
Falta de seguimiento continuado y permanente de tareas	Baja	Serio
Pérdida de documentación o archivos importantes	Baja	Serio
Estimación de costes errónea, fuera de la que se puede conseguir	Alta	Tolerable
Aparición de trabajos no programados	Alta	Tolerable
Fallos en el repositorio GIT	Media	Tolerable
Se proponen cambios en los objetivos que	Baja	Tolerable

suponen rehacer el diseño		
Desconocimiento de herramientas	Muy alta	Insignificantes

Tabla 27: Riesgos, probabilidades y efectos

4.5.3. Planificación de prevención.

Establecer unos planes que permitan minimizar, controlar o no posibilitar la aparición de estos riesgos.

RIESGO	ESTRATEGIA
Baja moral, desmotivación	Realizando las tareas de forma organizada y siguiendo un orden para tener en todo momento sensación de avance. Teniendo una visión siempre general del proyecto que ayude a ver el cumplimiento de los objetivos. Centrar los esfuerzos en aquellas tareas que más motiven. Intentar en la medida de lo posible no venirse abajo ante imprevisto o fallos puntuales en la realización de tareas.
Mala elección del motor gráfico sobre el que se va realizar la implementación	Realizar correctamente y exhaustivamente la fase de estudio y selección del motor gráfico para que la elección esté justificada y sea la mejor posible para la consecución de los objetivos propuestos.
Rotura de equipos o programas	Realizando un mantenimiento periódico de los equipos a utilizar, intentando que los mismos no sufran desperfectos por un mal uso, tratando que no se produzcan incompatibilidades entre programas instalados en los mismos.
Demoras en una toma de decisiones ante la necesidad de cambio de requerimientos	Planificando de una manera clara y bien estructurada para reducir al mínimo las necesidades de cambios en los requerimientos, si llegan a producirse actuar de manera bien estudiada para

	que afecte al menor número de tareas relacionadas o dependientes.
Baja por enfermedad	No es elemento que se pueda manejar y que pueda ser controlado, su aparición es aleatorio.
Mala comunicación con el tutor	Realizando una comunicación periódica y constante con el tutor y realizando reuniones para mantener un clima de sensación de apoyo constante por parte del mismo.
Objetivos o tareas poco claras	Analizando y pensando bien los objetivos que se pretenden conseguir con la realización del proyecto. Respecto a las tareas realizar un pequeño análisis antes de desarrollar cada una de ellos, para tener una idea más clara de que se quiere implementar o realizar.
Coherencia y cohesión de las diferentes partes del proyecto	Relacionando correctamente las partes de análisis y estudio con las de implementación, justificando en todo momento las actuaciones de ambos apartados para la consecución de los objetivos finales.
Falta de seguimiento continuado y permanente de tareas	Haciendo un uso óptimo de las herramientas que se disponen para la gestión del proyecto, teniendo claro en cada momento la tarea se tiene que realizar. Teniendo siempre presente la pila de tareas que se ha creado para el proyecto.
Pérdida de documentación o archivos importantes	Usando herramientas de almacenamiento en la nube y control de versiones que nos permitan tener registrados y guardados todos los documentos y archivos de funcionalidades realizados para evitar pérdidas.

Estimación de costes errónea	Maximizando la productividad en las horas de trabajo, buen uso de recursos que nos ayuden a mejorar y entender mejor las herramientas a utilizar. Haciendo un buen estudio de lo que deseamos hacer en cada momento y centrando el trabajo en pequeñas subtareas sencillas para posteriormente a través de ellas llegar a los objetivos marcados.
Aparición de trabajos no programados	Desglosando la planificación de cada una de las tareas lo máximo posible, realizando un buen estudio de las funcionalidades, dejando para cada tarea un tiempo de más por las posibles apariciones de estos trabajos.
Fallos en el repositorio GIT	Teniendo claro el manejo básico del funcionamiento de las herramientas para manejar el repositorio. Realizando correctamente las diferentes operaciones que se realizan en el repositorio, comentando todos los commits y realizando una buena resolución de conflictos cuando surjan.
Se proponen cambios en los objetivos que supone rehacer el diseño	Intentando que estos cambios sean los mínimos posibles y si se producen que afecten lo menos posible al resto de objetivos, tratando que los cambios se produzcan en pequeños lugares y no en extensas áreas del proyecto que supongan un alto grado de cambio.
Desconocimiento de herramientas	Realizando pequeñas pruebas de la misma accesoria al proyecto para aprender su funcionamiento. La fase de análisis debe estar correctamente realizada ya que un fallo en esta puede implicar la aparición de este riesgo.

Tabla 28: Planificación de riesgos

4.5.4. Monitorización de riesgos.

Además de la identificación, análisis y planificación hay que monitorizarlos, es decir, determinar identificadores potenciales para los riesgos dependiendo de su tipo.

RIESGO	IDENTIFICADORES POTENCIALES
Baja moral, desmotivación	Retraso general, no aprovechamiento del tiempo de trabajo, no realización de tareas o abandono del proyecto.
Mala elección del motor gráfico sobre el que se va realizar la implementación	Los objetivos finales no se pueden cumplir ya que el motor seleccionado impide la aplicación de la función final para la que el proyecto ha sido desarrollado.
Rotura de equipos o programas	Retraso en la entrega, perdida de información, imposibilidad para trabajar.
Baja por enfermedad	Retraso en la entrega de funcionalidades, sobrecarga para el resto del equipo de trabajo, imposibilidad de comunicación con la persona afectada.
Mala comunicación con el tutor	Falta de organización, pérdida de visión general del proyecto, retraso en tareas, desviación de objetivos.
Objetivos o tareas poco claras	Objetivos o tareas no conseguidos correctamente o cambiados respecto de su función original, tareas que no se realizan ante la imposibilidad por la poca claridad de las mismas.
Falta de coherencia y cohesión de las diferentes partes del proyecto	Falta de consecución objetivos por no estar bien relacionadas las partes del programa, mala memoria final, proyecto final no entendible, no consecución de objetivos. Proyecto no entendible y carente de sentido.

Falta de seguimiento continuado y permanente de tareas	Pérdidas de tiempo, falta de organización y de que tarea debe realizar cada miembro del equipo, fallos en la coordinación, retraso de entrega por mala planificación.
Pérdida de documentación o archivos importantes	Retraso en la entrega, volver a realizar tareas ya realizadas, funciones que no se entregan, falta de partes importantes del proyecto.
Estimación de costes errónea	Retrasos en la entrega, funcionalidad que se pierden o se retrasan por falta de tiempo, desmotivación y frustración por no cumplir lo planificado, retraso general del proyecto.
Demoras en una toma de decisiones ante la necesidad de cambio de requerimientos	Retraso general de proyecto, cambios importantes en las funcionalidades que puede hacer que algunas se pierdan, posibilidad de pérdida por no servir de parte del trabajo ya realizado.
Aparición de trabajos no programados	Funcionalidades retrasadas, imposibilidad de cumplir los plazos, pérdida de tiempo en la realización de los mismos lo que implica retraso general del proyecto.
Fallos en el repositorio GIT	Pérdida de elementos, fallos de consistencia en el proyecto, sobre escritura de elementos realizados anteriormente.
Se proponen cambios en los objetivos que supone rehacer el diseño	Objetivos no conseguidos o realizados de forma diferente a lo planificado, retraso por pérdida de tiempo en rehacer el proyecto.
Desconocimiento de herramientas	Retraso en la entrega, descontrol en la planificación, funciones imposibles de implementar.

Tabla 29: Monitorización de riesgos

5. Cuerpo del trabajo

5.2. Análisis de Frameworks

5.2.1 Consideraciones previas.

Equipo de trabajo.

Las pruebas tanto de instalación, realización de proyectos y prueba de los juegos se va a realizar en el mismo equipo para que la comparación entre los diferentes candidatos sea lo más justa posible. El equipo de prueba cuenta con las siguientes características:

Equipo de pruebas	
Sistema Operativo	Windows 10.
Procesador	I7 4750HQ
Memoria	8GB de RAM
Tarjeta Gráfica	GTX 950
Otros	Conexión internet 100Mb.

Tabla 30: Datos equipo de pruebas

Versiones de los programas.

Para realizar la implementación de las funcionalidades, la creación de pequeños proyectos y todos aquellos análisis que requieran el uso de los frameworks se va usar siempre que sea posible la versión gratuita de los mismos.

Precio medio de las licencias y de los módulos de exportación.

Para poder evaluar los apartados precio de licencias y de los módulos de exportación es necesario tener un baremo con el que determinar si estos precios son altos o bajos. Para ello se va usar el precio medio, tanto de las licencias como de los módulos, del conjunto de los programas. En la siguiente tabla se muestran los precios medios.

En la siguiente tabla se detallé el precio de las licencias más avanzadas de los programas y su precio medio.

Precio medio licencias	
Gdevelop	0 €
Contract 2	405,99€
Click Team Fusion	211.99€
Game Maker	99€

PRECIO MEDIO	179,25€
---------------------	---------

Tabla 31: Precio medio de licencias

Para delimitar la puntuación de licencia se va usar el siguiente baremo.

Baremo puntuación precio de licencias.	
Precio medio +/-20€	50% del total de la puntuación.
Precio medio hasta +50€	25% del total de la puntuación.
Precio medio hasta - 50€	75% del total de la puntuación.
Precio medio + de 50€	0% del total de la puntuación.
Precio medio – de 50€	100% de la puntuación.

Tabla 32: Baremo puntuación precio licencias

En el caso de los módulos de exportación para calcular la media se van utilizar el precio medio de todos los módulos disponibles para cada programa, pero solo de las versiones Android, HTML5 e Ios, ya que si se incluyen precios de módulos de exportación de consolas que suelen ser los más costosos, que solo algunos de los frameworks tienen, estaríamos desvirtuando el precio medio y penalizando a aquellos programas que posibilitan este tipo de exportación.

Precio medio módulos de exportación.	
Gdevelop	0€
Construct 2	0€
Click Team Fusion	76.3€
Game Maker	83.32€
TOTAL PRECIO MEDIO CONJUNTOS	39.9€

Tabla 33: Precio medio módulos de exportación

Al igual que para el precio de las licencias, el precio de los módulos de exportación también cuenta con un baremo que determinará la puntuación del programa en este apartado.

Baremo puntuación precio de módulos.	
Precio medio +/-35€	50% del total de la puntuación.
Precio medio hasta +40€	25% del total de la puntuación.
Precio medio -40€	100% del total de la puntuación.
Precio medio + de 80€	0% del total de la puntuación.

Tabla 34: Baremo precio módulos de exportación

5.2.2. Análisis Gdevelop.

Criterio 1: Requisitos e instalación.

Los requisitos para poder desarrollar en Gdevelop son bastante asequibles y la mayoría de equipos pueden trabajar con el programa ya que consume pocos recursos del sistema, como se puede observar en la siguiente tabla.

REQUISITOS GDEVELOP	
Sistema Operativo	Mínimo Windows XP SP3, soportados Windows Vista, 7, 8 y 10 Ubuntu Otros Linux (Versiones no oficiales) Mac (En fase beta)
Procesador	200Mhz o mayor
Memoria	512 MB de RAM
Tarjeta Gráfica	Mínimo Tarjeta Gráfica con soporte para Direct3D 9
Otros	Conexión internet necesaria para la instalación, actualización y descarga del software.

Tabla 35: Requisitos Gdevelop

Cuenta con la ventaja de que no solo se puede desarrollar en Windows sino que también tiene disponible un entorno de desarrollo en Ubuntu y en otras distribuciones de Linux como pueden ser Fedora⁹ o Arch¹⁰(aunque éstas no son versiones oficiales han sido creadas por la comunidad) y una versión para Mac que por el momento se encuentra en estado beta y con una funcionalidad bastante limitada.

Gdevelop tiene una versión app-web, llamado **Gdevapp**, véase Figura 1, que, aunque de forma más limitada que la versión de escritorio, permite trabajar con el programa desde cualquier dispositivo conectado a Internet, aunque se manejan de forma independiente. Gracias a su sistema de importación y exportación, desde la App web

⁹ distribución Linux para propósitos generales basada en RPM, que se caracteriza por ser un sistema estable, la cual es mantenida gracias a una comunidad internacional de ingenieros, diseñadores gráficos y usuarios que informan de fallos y prueban nuevas tecnologías. Cuenta con el respaldo y la promoción de Red Hat.

¹⁰ Arch Linux es una distribución GNU/Linux independiente, de propósito general, desarrollada para x86-64. Se esfuerza en ofrecer las últimas versiones estables de la mayoría del software, siguiendo un modelo rolling-release (lanzamiento continuo). La instalación por defecto deja un sistema de base mínima, que el usuario configurará posteriormente agregando lo que necesite.

podemos trabajar con proyectos generados por el programa de escritorio en la aplicación web y a la inversa.

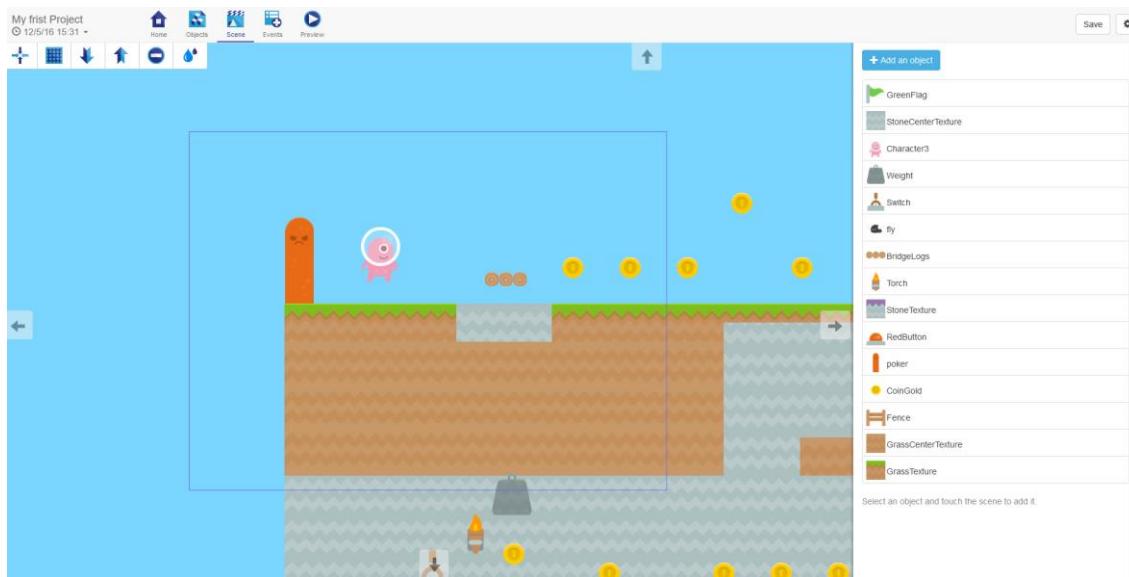


Figura 14: Gdevapp

Puntuación subcriterio requisitos

Como se ha comprobado los requisitos de funcionamiento de Gdevelop son asumibles por casi la totalidad de los usuarios y además no limita su uso a sistemas operativos Windows, teniendo alternativas tanto en sistema Ios como versión accesible desde cualquier sistema y dispositivo. Por estos motivos la puntuación en este apartado que muestra la tabla siguiente es la máxima posible.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Requisitos de instalación.	0-4	4

Tabla 36: Gdevelop puntuación instalación

Una vez se ha comprobado que el equipo dónde se va a instalar GDevelop cumple los requisitos mencionados en la Tabla 1, el proceso de instalación es bastante sencillo y tiene los siguientes pasos:

1. Descargar el paquete de instalación para el sistema en el que se vaya a desarrollar. Desde la web oficial (<http://compilgames.net>), la descarga tiene un peso de 109 Mb.

2. El paquete de instalación está en castellano y siguiendo las instrucciones que el programa indica se completa la instalación. Hay que indicar que en un momento del proceso se pregunta si se desea instalar no solo el programa, sino también junto a él toda una serie de ejemplos y recursos, como podrían ser imágenes y sprites, véase Figura 2.

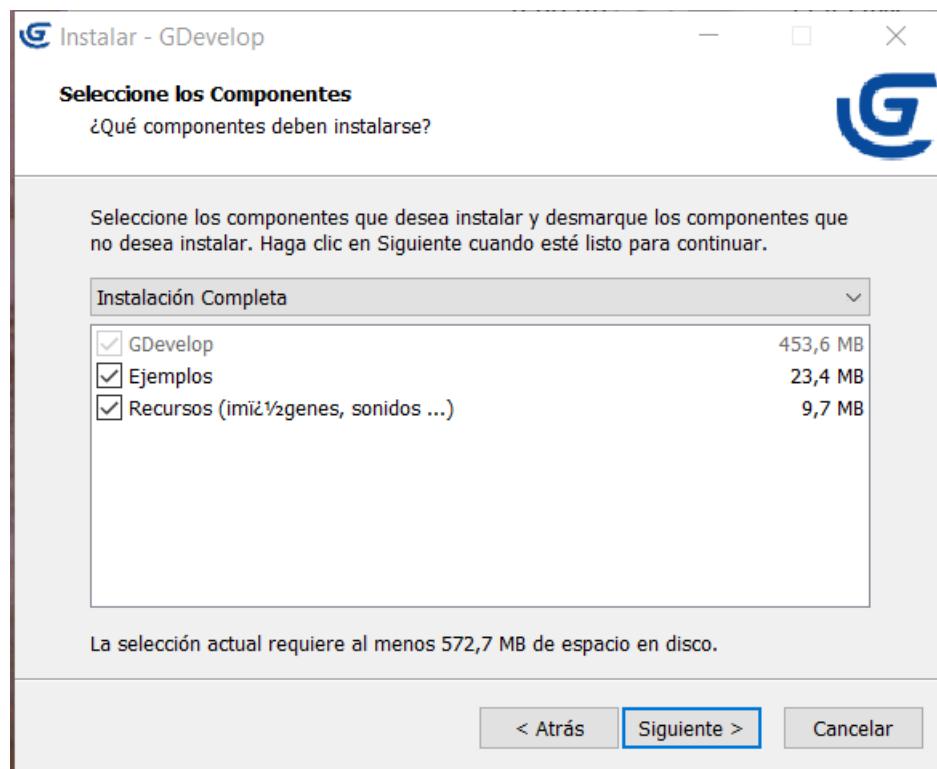


Figura 15: Gdevelop instalación de recursos accesorios

3. El proceso de instalación en el equipo de prueba ha tenido un tiempo total de ocho minutos y 26 segundos. La instalación del programa con los ejemplos y los recursos adicionales que se indican en el paso anterior tiene un peso total en el equipo de 572Mb.

Resumen proceso de instalación

La tabla siguiente muestra el resultado de la instalación y el resumen de los datos obtenidos en el equipo de pruebas.

Gdevelop Instalación en el equipo de pruebas	
Tamaño de descarga	107 Mb
Requiere programas adicionales	No
Instalación de módulos o recursos adicionales durante la instalación	Si
Tiempo total	8 minutos 26 segundos.
Tamaño posterior a la instalación	572 Mb

Tabla 37: Resumen proceso de instalación Gdevelop

Datos de instalación encuesta.

En la siguiente tabla se muestra un resumen de resultados del proceso de instalación de los 8 participantes en el estudio vía encuestas realizado. Los datos completos se encuentran en el Anexo I: Resultados estudio de instalación e interfaz.

Gdevelop datos estudio instalación	
Tiempo medio	4 minutos 9 segundos
Valoración media sencillez del proceso de instalación (0 a 10)	9.25
Valoración media de la funcionalidad del proceso de instalación (0 a 10)	9
Puntuación media funcionalidad y sencillez	9.125
Fallos en instalación.	No

Tabla 38: Gdevelop datos encuesta instalación

Como se puede comprobar en la tabla el tiempo de instalación es bastante pequeño, la valoración tanto de la funcionalidad y sencillez del proceso de instalación son muy altas y los sujetos del estudio no han tenido problemas ni fallos de instalación.

Puntuación subcriterio instalación.

Gdevelop ha tenido un tiempo de instalación menor de 5 minutos y su valoración media de funcionalidad y sencillez del proceso de instalación es mayor de 9, concretamente 9.125. Conforme a la ponderación definida en el marco teórico le corresponde 2 puntos en este apartado.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Instalación.	0-2	2

Tabla 39: Gdevelop subcriterio instalación

Resumen puntuación requisitos e instalación.

En la siguiente tabla se resumen la puntuación en el criterio requisitos e instalación.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Requisitos de instalación.	0-4	4
Instalación	0-2	2
TOTAL		6

Tabla 40: Gdevelop criterio requisitos e instalación

Criterio 2: Documentación, tutoriales y comunidad.

Gdevelop cuenta con una wiki colaborativa, con la que se pueden aprender todos los aspectos del programa, de su instalación y puesta en funcionamiento así como de los distintos tipos de editores con los que cuenta y cómo funciona cada uno de ellos.

Esta wiki colaborativa está fundamentalmente en inglés, aunque algunos artículos y apartados también se encuentran disponible en castellano y los responsables de la misma permiten que los usuarios colaboren traduciendo las partes de la misma por lo que el número de textos traducidos va en aumento.

La documentación está escrita de manera que usuarios de todos los niveles pueden entender y realizar la mayoría de las funcionalidades que en ella se explican.

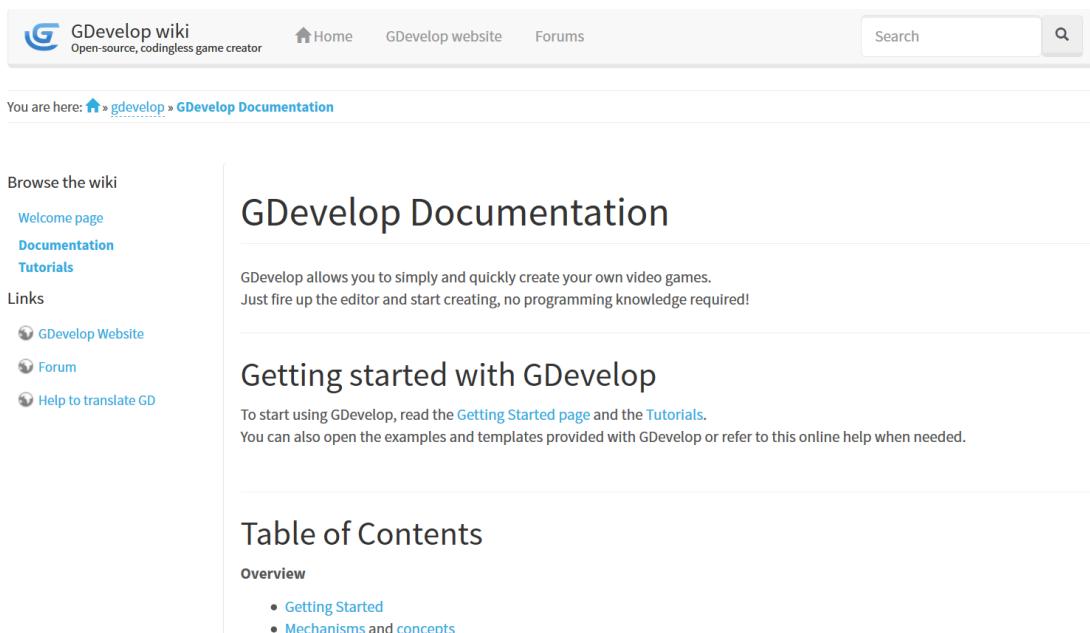
A screenshot of the GDevelop Documentation wiki homepage. The header includes the logo 'GDevelop wiki' and the tagline 'Open-source, codingless game creator'. Navigation links for 'Home', 'GDevelop website', and 'Forums' are at the top right, along with a search bar. Below the header, a breadcrumb trail shows the user's location: 'You are here: Home > gdevelop > GDevelop Documentation'. On the left, a sidebar titled 'Browse the wiki' lists links for 'Welcome page', 'Documentation', 'Tutorials', 'Links', 'GDevelop Website', 'Forum', and 'Help to translate GD'. The main content area features a large heading 'GDevelop Documentation' with a subtext: 'GDevelop allows you to simply and quickly create your own video games. Just fire up the editor and start creating, no programming knowledge required!'. Below this is a section titled 'Getting started with GDevelop' with a note: 'To start using GDevelop, read the Getting Started page and the Tutorials. You can also open the examples and templates provided with GDevelop or refer to this online help when needed.' At the bottom, there is a 'Table of Contents' section with an 'Overview' link and a bulleted list: 'Getting Started' and 'Mechanisms and concepts'.

Figura 16: Portada de la wiki de Gdevelop

Gdevelop cuenta con una wiki colaborativa, con la que se pueden aprender todos los aspectos del programa, de su instalación y puesta en funcionamiento, así como de los distintos tipos de editores con los que cuenta y cómo funciona cada uno de ellos.

Esta wiki colaborativa está desarrollada fundamentalmente en inglés, aunque algunos artículos y apartados también se encuentran disponible en castellano y los responsables de la misma permiten que los usuarios colaboren traduciendo distintas partes de la misma por lo que el número de textos traducidos va en aumento.

La documentación está escrita de manera que usuarios de todos los niveles pueden entender y realizar la mayoría de las funcionalidades que en ella se explican.

The screenshot shows the GDevelop wiki homepage. At the top, there's a navigation bar with the GDevelop logo, "GDevelop wiki Open-source, codingless game creator", links for "Home", "GDevelop website", and "Forums", and a search bar. Below the header, a breadcrumb trail says "You are here: gdevelop > GDevelop Documentation". The left sidebar has sections for "Browse the wiki" (Welcome page, Documentation, Tutorials), "Links" (GDevelop Website, Forum, Help to translate GD), and "Overview" (Getting Started, Mechanisms and concepts). The main content area features a large h1 "GDevelop Documentation" and a brief introduction: "GDevelop allows you to simply and quickly create your own video games. Just fire up the editor and start creating, no programming knowledge required!". Below this is a section titled "Getting started with GDevelop" with a note about reading the Getting Started page and Tutorials, and links to examples and templates. A "Table of Contents" section follows, listing "Overview" and "Getting Started" under "Overview".

Figura 17: Portada de la wiki de Gdevelop

La wiki tiene un apartado específico dedicado a tutoriales. En el primero, recomendado para principiantes, se enseñen las funciones básicas del programa, el movimiento y control del personaje y la gestión básica del proyecto mediante la realización de un ejemplo básico de un juego de plataformas. Después de este primer tutorial que se recomienda hacer a usuarios principiantes, se encuentran una serie de tutoriales que explican determinadas funcionalidades y que están clasificados de la siguiente forma:

- **Para principiantes:** En este apartado se encuentran todos aquellos tutoriales que son necesarios para poder hacer un uso básico del programa. Entre ellos se encuentra el citado tutorial de plataformas, así como otros relativos a la exportación de proyectos y otros de elementos sencillos. Este apartado incluye 5 tutoriales.
- **Otros tutoriales:** Engloba funciones más avanzadas que expanden y completan a los tutoriales para principiantes. Por ejemplo, el uso de variables, la distribución de los juegos, creación de un menú de juego o dos tutoriales de juegos más avanzados para aprender más características del programa. Este apartado lo componen 10 tutoriales.
- **Consejos y pequeñas explicaciones:** Recomendaciones para trabajar correctamente y aprovechar al máximo las funciones del programa y una serie de pequeños dudas que suelen aparecer recurrentemente a los usuarios. Hay cuatro tutoriales disponibles en esta sección.

- **Tutoriales avanzados:** Enseñan el uso de las posibilidades superiores del programa. Son tutoriales para los que se necesita un gran conocimiento del programa y/o conocimiento de Javascript o C++, ya que en este aparto se explica el uso de estos lenguajes para crear eventos personalizados y nuevos comportamientos dentro de los proyectos. Encontramos 6 tutoriales en este apartado.
- **Tutoriales de la comunidad:** Los responsables de Gdevelop van recopilando los tutoriales que creen que pueden ayudar a los usuarios en el manejo del programa, los que consideran más completos o los que por razones de utilidad de lo explicado o por temática que ha realizado la comunidad en sus foros oficiales y les da visibilidad. Los tutoriales de la comunidad son 36.

La mayoría de tutoriales oficiales de Gdevelop contienen además los sprites, backgrounds y todos los recursos que van a hacer falta para seguir la explicación y completarlo, es un punto importante y que ayuda a reducir tiempo en esta fase de aprendizaje ya que no se debe dedicar tiempo a buscar estos elementos.

Los foros oficiales otro de los puntos fuertes de Gdevelop, su comunidad es bastante activa y se puede desde reportar bug detectados en el programa, realizar consultas sobre el programa, mostrar los juegos desarrollados, así como preguntar dudas respecto a la implementación de determinadas funciones. Los foros oficiales están disponibles en inglés y no existe versión en castellano por el momento.

Puntuación subcriterio documentación, foros y tutoriales oficiales.

Por todos los aspectos relativos a manuales, tutoriales, y foros que componen este criterio descritos anteriormente, la puntuación en este criterio es bastante alta, ya que el único punto negativo que se puede encontrar es la falta de más proyectos de ejemplo y videos explicativos.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Documentación, foros y tutoriales oficiales	0-6	5

Tabla 41: Gdevelop puntuación Documentación, foros y tutoriales oficiales

Tutoriales oficiales realizados.

[How to make a platformer game?](#)

Es el primer tutorial que desde la página oficial se recomienda hacer para empezar a entender el funcionamiento general del programa. De hecho, en su primer apartado explica el proceso de instalación del mismo.

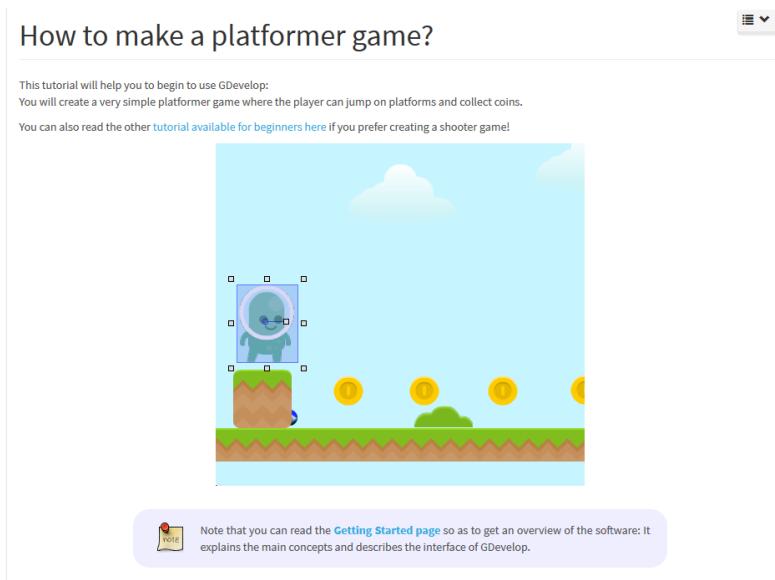


Figura 18: Portada del tutorial: How to make a platformer game?

En líneas generales el tutorial es muy completo, ofreciendo una guía muy detallada de todos los pasos que se deben ir realizando en el mismo. Destaca sobre todo en los siguientes aspectos:

1. Cuenta con numerosas imágenes que ayudan a visualizar todos los pasos que se deben realizar.
2. Como se ha comentado anteriormente tiene todos los recursos necesarios para completarlo (Sprites y elementos de la escena). En estas fases de aprendizaje donde lo importante que el usuario se familiarice con el entorno de desarrollo , es muy adecuado contar con los recursos, para maximizar el tiempo dedicado al aprendizaje y no a buscar recursos accesorios.
3. El apartado que incluye los eventos, está muy bien explicado, lo que permite hacerse una idea bastante buena de cómo funciona a nivel lógico.
4. Se facilita tanto el archivo fuente del proyecto por si en algún momento un usuario no sabe realizar alguna acción o quiere comparar su implementación con la del tutorial pueda acudir a este recurso como un ejemplo compilado en

versión web del resultado del tutorial

5. El tutorial va escalando su dificultad y no repite conceptos. La primera vez que enseña al usuario una nueva función la explica detalladamente con diversas imágenes y textos explicativos, en las sucesivas veces que aparece esa misma acción el tutorial no la detalla en profundidad e indica que clase de acción es y en qué parte del tutorial la hemos realizado. Todo esto ayuda al usuario a intentar realizar las cosas por sí mismo, progresar en su manejo del programa y hace el manual más ameno.

Pero también tiene una serie de inconvenientes de apartados que no están correctos o podría ser mejorables:

1. El tutorial está en inglés y pese a que a primera vista esto no supone ningún problema como para considerarlo un inconveniente o elemento que reduce la valoración general del tutorial, sí lo hace el hecho de que el programa se encuentre en castellano, con lo que la totalidad de las acciones que se deben llevar a cabo deben ser traducidas por el usuario o identificadas por los iconos.
2. Algunas imágenes del tutorial no son correctas. El tutorial consiste en la creación de un videojuego de plataformas, en el que en un apartado explica la creación de monedas en el escenario y su posterior colisión con el personaje, pero las imágenes que aparecen en este apartado no se corresponden con lo explicado.
3. Tiene partes desactualizadas. El tutorial parece realizado con una versión anterior del programa y hay apartados que ahora son distintas. En concreto se ha detectado este problema en los apartados que explican la colisión y como crear capas de texto.
4. El tutorial contiene la creación de un enemigo que se mueve por el escenario. Pero la creación de este movimiento está mal explicada y estructurada y es complicado entenderla correctamente.

Begginner's Tutorial

Junto al anterior, este tutorial constituye la primera aproximación al programa y son los que se recomienda realizar en un primer momento para aprender las funciones básicas y a manejarse con él mismo.

En este tutorial se explica cómo crear un juego de disparos donde el jugador tiene que destruir los tanques enemigos que se mueven hacia él apuntando y disparando con el ratón.



The screenshot shows a step-by-step tutorial for GDevelop. On the left, there's a sidebar with links like 'Welcome page', 'Documentation', 'Tutorials', 'Links', 'GDevelop Website', 'Forum', and 'Help to translate GD'. The main content area has a title 'Step by step tutorial for GDevelop' and a sub-section 'This tutorial will help you to begin to use GDevelop: You will create a very simple game where enemy are going toward the player, who will be able to shot them so as to make them explode.' It also mentions another tutorial for beginners. A large image in the center shows a game scene with two green enemy turrets and a player character. One enemy is hit and exploding. A note at the bottom says: 'Note that you can read the [Getting Started page](#) so as to get an overview of the software: It explains the main concepts and describes the interface of GDevelop.'

Figura 19: Tutorial Gdevelop Begginers

El tutorial comparte con anterior todos sus aspectos positivos, pero tiene los siguientes inconvenientes o errores:

1. Al igual que el anterior el tutorial esta en inglés y el programa en castellano, por lo que se ha comentado anteriormente esto produce problemas de coherencia.
2. La parte de la animación de la explosión no ésta bien explicada y si se implementa como indica el tutorial aparecerá en un lugar distinto al que debería aparecer, para corregirlo se pueden realizar dos acciones: ajustar restando a la posición de X e Y para situarla en el lugar correcto; o bien crear un nuevo punto en el objeto y que sea este el genere la colisión. En esta prueba se ha optado por la primera solución.

También se han detectado posibles mejoras y modificaciones del tutorial. Son las siguientes:

1. Toda la parte del fin de partida ha sido implementada de forma diferente a la explicada en el tutorial, ya que no era la forma correcta de realizarla en términos de gestión de recursos y un mejor flujo de juego.
En el manual, cuando un enemigo golpea a un jugador aparece el mensaje de Game Over sobre la pantalla y no sucede nada más y para volver a jugar hay que detener el juego y volver a ejecutarlo.
Para mejorar el proceso se ha creado otra escena a la que se cambia después del mensaje de "Game Over" y un botón que permite volver a iniciar el juego.
2. Se ha añadido una variable *puntuación* que se imprimirá en la parte superior de la pantalla y que se incrementará cada vez que un enemigo sea destruido.
3. Se ha añadido una serie de sub- eventos (véase Figura 7) que conforme la puntuación aumente también lo haga la velocidad de los enemigos, para aumentar la dificultad progresivamente.

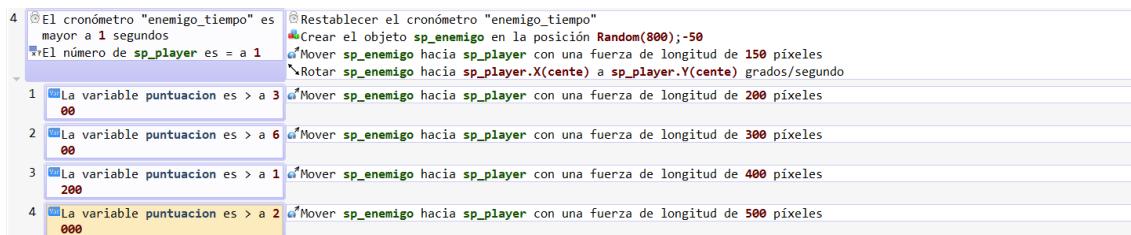


Figura 20: Eventos puntuación

4. La creación de las balas tiene una nueva condición y es que exista un jugador en la escena. Esto impide que después de destrucción del jugador (véase Figura 8) se sigan generando balas, como si ocurría con la implementación original.



Figura 21: Destrucción del jugador

Resumen de tutoriales realizados.

La tabla siguiente contiene el resumen de los tutoriales realizados , el tiempo invertido y las funciones aprendidas durante su realización.

Resumen de tutoriales realizados.	
Tiempo dedicado.	7 horas 4 minutos
Tutoriales realizados	2
Tutoriales completados	2
Juegos completos	1
Errores detectados en los tutoriales	Pocos
Modificaciones realizadas	Pocas y accesorias
Funcionalidades aprendidas en los tutoriales	<ul style="list-style-type: none"> • Creación de proyectos. • Añadir y gestionar Sprites. • Añadir fondo a la escena. • Movimiento en 2D. • Salto. • Movimiento en 3D. • Creación de objetos y enemigos en posiciones aleatorias.

	<ul style="list-style-type: none"> Movimiento unidireccional de enemigo, con persecución y cambio de dirección. Rotar objetos con el ratón. Disparo de balas. Colisión entre objetos. Destrucción de objetos tras colisión.
--	--

Figura 22: Gdevelop resumen tutoriales realizados

Puntuación subcriterio análisis de tutoriales oficiales.

Como se puede observar en la tabla del apartado anterior, el número de funcionalidades aprendidas es muy alto, los errores detectados son pocos y las modificaciones realizadas escasas y accesorias, solo algunos pequeños fallos de explicaciones, de correlación entre imágenes y explicación y coherencia impiden que tenga la máxima puntuación.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Análisis de tutoriales oficiales	0-8	7

Tabla 42: Subcriterio Gdevelop Análisis de tutoriales oficiales

A continuación, se detalla la presencia de Gdevelop en Reddit y Taringa.

Reddit

La presencia de Gdevelop en Reddit es meramente anecdótica, como muestra la Figura 10, con tan solo 8 temas abiertos y el último mensaje en uno de ellos fue escrito hace más de 4 meses. Además, en todos de ellos los desarrolladores del programa remiten a los foros oficiales.

Análisis de Herramientas para la creación de videojuegos | Óscar Sánchez

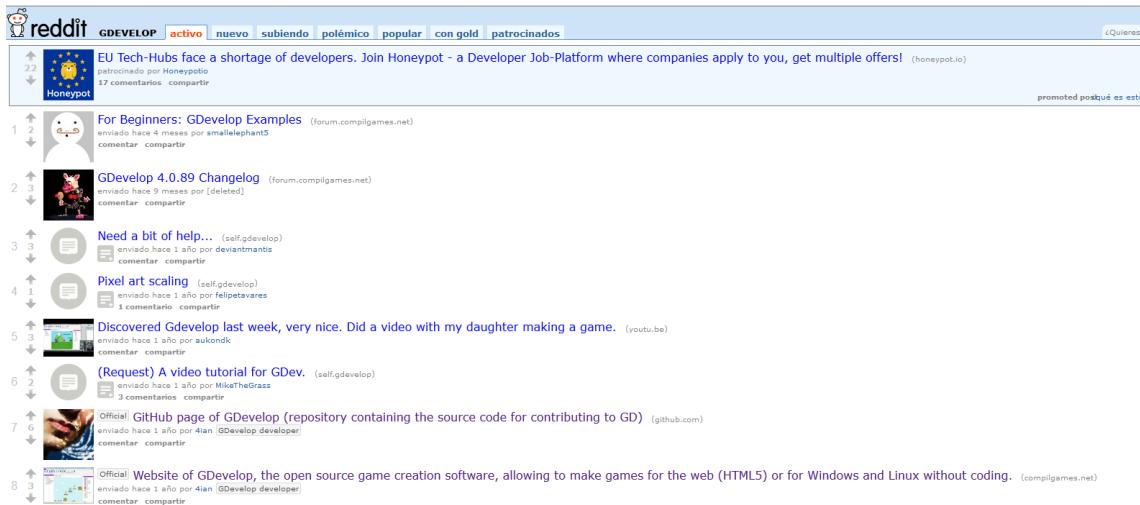


Figura 23: Gdevelop en Reedit

Taringa.

Existe una comunidad, pero solo tiene dos miembros y un tema activo por lo que no se puede considerar como tal.

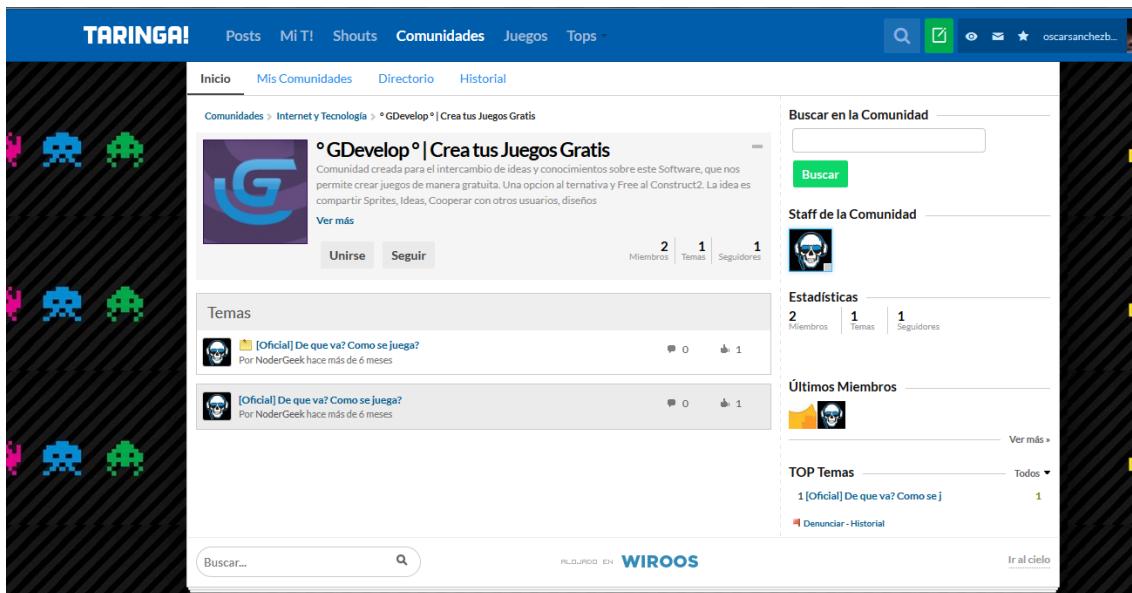


Figura 24: Gdevelop en Taringa

Puntuación subcriterio comunidades y foros de desarrolladores.

Como se ha podido comprobar la presencia de Gdevelop en Reedit y Taringa es meramente anecdótica y además remite a los foros oficiales, por lo que la puntuación en este apartado es mínima.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Foros y comunidades de desarrolladores	0-2	0.5

Tabla 43: Subcriterio *foros y comunidades de desarrolladores*

Resumen de puntuación Documentación, tutoriales y comunidad.

La tabla siguiente resume la puntuación del este criterio.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Documentación, foros y tutoriales oficiales	0-4	5
Análisis de tutoriales oficiales	0-8	7
Foros y comunidades de desarrolladores	0-4	0.5
TOTAL		12.5

Tabla 44: Gdevelop criterio documentación, tutoriales y comunidad

Criterio 3: Funcionamiento y dificultad de uso.

Durante el desarrollo de los tutoriales no se han detectado fallos en el programa y ha funcionado en todo momento de forma fluida, tampoco se han reportado errores en este sentido por los participantes en las encuestas.

Puntuación subcriterio funcionamiento.

Por lo comentado anteriormente la puntuación en este apartado es la máxima posible, como muestra la siguiente tabla.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Funcionamiento General	0-4	4

Tabla 45: Gdevelop subcriterio funcionamiento general

Puntuación primeros pasos: creación de un pequeño proyecto.

Gdevelop por su concepción, funcionamiento y por los distintos elementos que se comentarán a continuación hace que los primeros pasos con el programa sean bastante asequibles y sencillos para todo tipo de usuarios.

Un punto importante de Gdevelop que ayuda a conocer sus fundamentos básicos es que cuenta con una serie de plantillas o tipos de juego preestablecidos que aceleran la tarea de comprensión y aprendizaje. Estas plantillas se aplican al crear proyecto nuevo de forma que primero se pregunta la clase de juego a desarrollar. Así Gdevelop prepara todos los elementos necesarios para crear un juego de ese tipo además de crear una pequeña escena a modo de ejemplo,

Es un buen comienzo para los usuarios utilizar estas plantillas para conocer cómo funcionan los distintos elementos y eventos de Gdevelop, para una vez comprendidos poder utilizar el modo sin plantilla donde se crea todo desde cero.

Por lo visto en la fase de tutoriales en menos de 8 horas se ha podido completar la realización de un juego y se han sentado las bases para realizar otro por lo que el periodo que necesita un usuario para empezar a crear juegos es bastante corto.

Por todo lo anteriormente comentado la puntuación de en este apartado es la máxima posible.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Primeros pasos: Creación de un pequeño proyecto.	0-4	4

Tabla 46: Gdevelop subcriterio funcionamiento general

Funciones básicas

A continuación se va a explicar cómo implementar las funciones básicas descritas en la definición de los criterios de selección y se va a puntuar en función de su dificultad.

Creación de un proyecto.

La creación de un proyecto nuevo es bastante sencilla, se puede observar en la Figura 11, mediante desde la interfaz con la siguiente sucesión de pasos:

Ruta/comandos	Archivo->Creación de nuevo proyecto.
Variables	<p>Plantilla del proyecto: Para facilitar la creación de proyectos nuevos, Gdevelop permite que se use una de las plantillas disponibles que corresponden a distintos tipos de géneros o seleccionar proyecto vacío.</p> <p>Tipo de proyecto: Existen dos tipos de proyectos, nativo (es decir para el sistema operativo en el que se está trabajando) y HTML5</p>

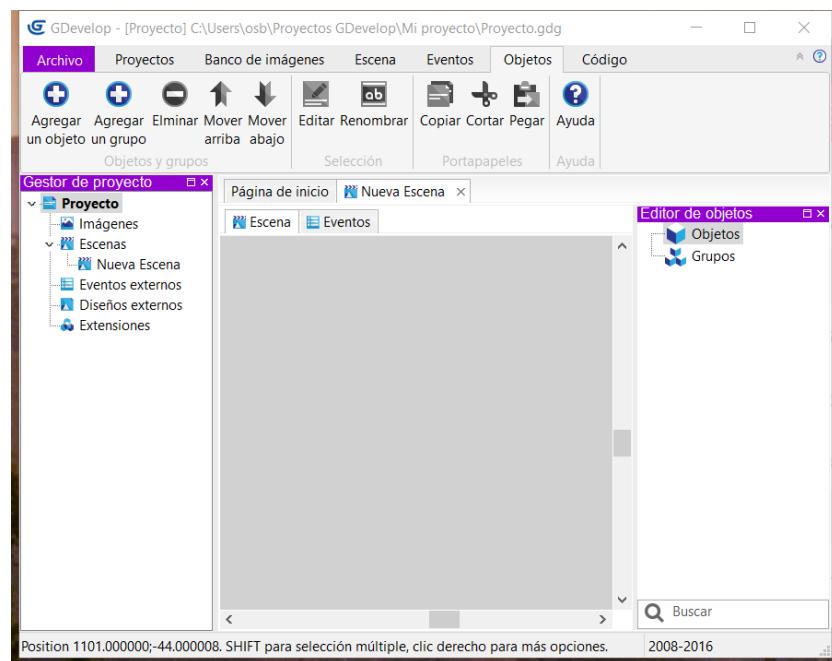


Figura 25: Gdevelop creación de un nuevo proyecto

Tabla resumen funcionalidad.

Funcionalidad	Valoración	Puntuación
Creación proyecto.	Muy fácil	10

Insertar fondo en la escena.

Para dotar a la escena de un fondo el proceso que hay que seguir es el siguiente:

1. Crear Objeto Sprite con la imagen de fondo.
2. Arrastrar el objeto desde la capa de editor de objetos a la escena.
3. Botón secundario sobre el objeto fondo que se acaba de arrastrar a la escena.
4. Se ajusta al tamaño a la escena, o a la parte de la misma donde se quiera que aparezca el fondo.
5. Cambiar el valor del Plano Z del objeto para que permanezca atrás del resto de los objetos.

Ruta/comandos	Doble clic sobre el objeto en la escena-> Propiedades.
Variables.	Plano(Z): Cambiar el valor Plano(Z) por el cero o un número negativo para que éste por debajo del resto de objetos. Este valor siempre tiene que ser menor que el Plano (Z) de todos los objetos si se quiere que se mantenga siempre como fondo.

Tabla resumen funcionalidad.

Funcionalidad	Valoración	Puntuación
Insertar fondo	Media	5

Movimiento de personaje en dos direcciones.

Para dotar a un personaje controlable de movimiento en dos direcciones, como sería por ejemplo en un juego plataformas donde el personaje se mueve de izquierda a derecha, Gdevelop cuenta con tipos de comportamientos prestablecidos que facilitan enormemente la implementación de dicha funcionalidad.

En Gdevelop este tipo de movimiento se basa en dos elementos o patrones pre establecidos:

1. Objeto que se mueve sobre plataformas: Este es el patrón de comportamiento que debe tener el personaje que se desea controlar y realizar con el movimiento, con las variables de configuración que indica la Figura 12. Para dotar de dicho comportamiento a un objeto el proceso es el siguiente :

Ruta/comandos	Doble clic sobre el objeto en la escena->Propiedades->Comportamiento->Aregar->Seleccionar "Objeto que se desplaza sobre plataformas"->Aceptar
Variables	<p>Aceleración: Velocidad con la que el personaje va empezar a moverse y a mantener hasta alcanzar su velocidad máxima.</p> <p>Can grab platform ledges:</p> <p>Controles por defecto: Si se marca esta opción no se deberá especificar las teclas con las que se va controlar el personaje, de forma que el personaje se moverá con las flechas izquierda y derecha del teclado y con espacio se produce el salto.</p> <p>Desaceleración: Velocidad de frenado del personaje.</p> <p>Grab offset on Y axis:</p> <p>Grab tolerance on X axis:</p> <p>Gravedad: Fuerza con la que el personaje es atraído hacia abajo de la escena.</p> <p>Velocidad de caída máxima: Tope de velocidad tras caída de plataforma que puede alcanzar.</p> <p>Velocidad de salto: Rapidez con la que un personaje salta.</p>

	<p>Velocidad máx: Velocidad que puede alcanzar el Sprite que se controla.</p> <p>Ángulo máx. de pendiente: Ángulo a partir del cual el personaje comenzará a caer y a desplazarse.</p>
--	--

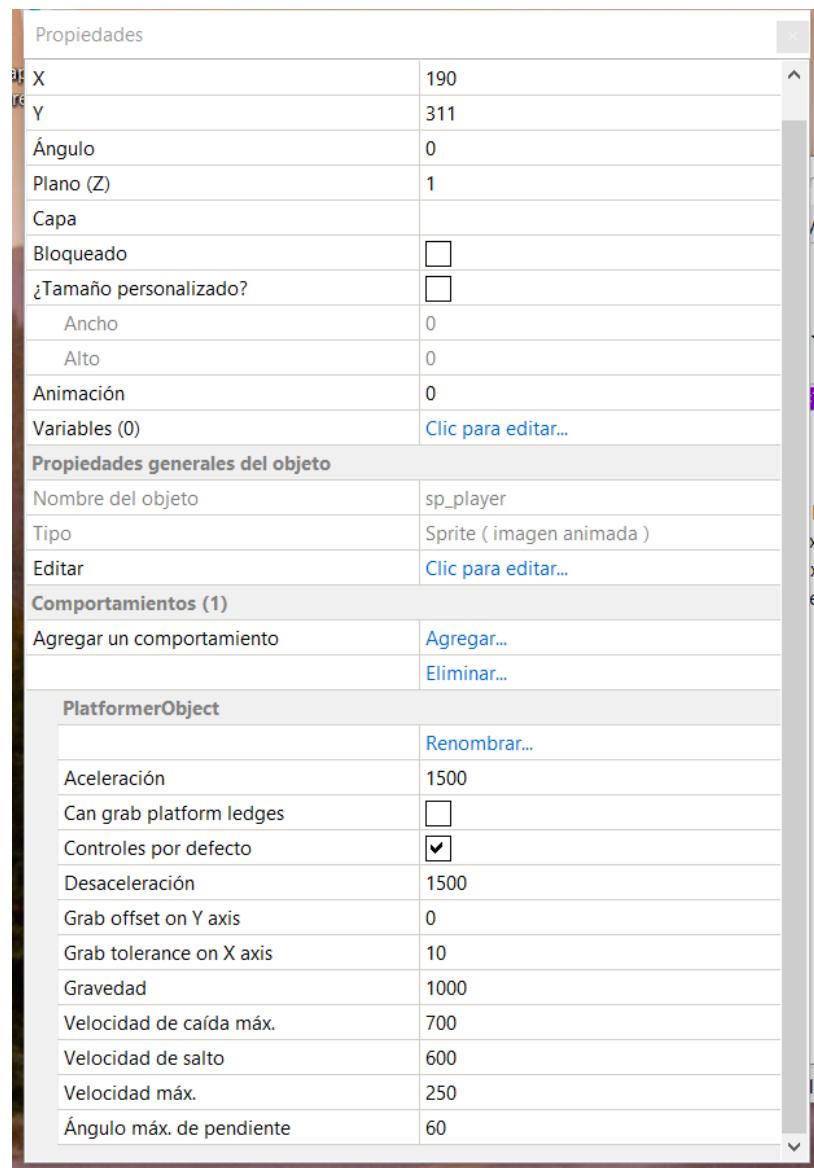


Figura 26: Gdevelop: Variables movimiento en 2D

2. Objeto plataforma: Pero si a un objeto se le indica que se trata de un *objeto que se desplaza sobre plataforma*, tiene que estar acompañado de otros que

hagan la función de plataforma ya que si no caerían infinitamente hacia debajo de la escena.

Ruta/comandos	Doble clic sobre el objeto en la escena->Propiedades->Comportamiento->Aregar ->Seleccionar "Plataforma"->Aceptar
----------------------	--

Una vez se tiene el *Objeto que se mueve sobre plataformas* y una o varias *plataformas* basta con colocar al Sprite sobre la plataforma en la escena.

Tabla resumen funcionalidad.

Funcionalidad	Valoración	Puntuación
Movimiento en dos dimensiones	Muy fácil	10

Salto de personaje que se mueve en dos direcciones.

El salto del personaje en 2D mediante la pulsación de una tecla, en este caso espacio, es automática en Gdevelop si se trata de un objeto que se mueve por plataformas. Se puede variar su altura, y/o velocidad desde el editor de comportamiento de *objeto que se mueve por plataformas* e incluso desactivar o anular dicha posibilidad.

Tabla resumen funcionalidad.

Funcionalidad	Valoración	Puntuación
Salto.	Muy fácil	10

Movimiento personaje en 8 direcciones.

El movimiento en 8 direcciones, es decir, izquierda, derecha, arriba, abajo y sus respectivas diagonales se hace de forma muy parecida a la implementación del movimiento en dos direcciones pero seleccionando en este caso el patrón de comportamiento "movimiento en 4-8 direcciones".

Ruta/comandos	Doble clic sobre el objeto en la escena->Propiedades->Comportamiento->Aregar->Seleccionar "Movimiento de arriba abajo (en 4-8 direcciones)->Aceptar
Variables	<p>Aceleración: Velocidad con la que el personaje va empezar a moverse y va mantener hasta alcanzar su velocidad máxima.</p> <p>Controles por defecto: Si se marca esta opción no se deberá especificar las teclas con las que se va controlar el personaje, en esta opción el personaje se mueve con las flechas izquierda y derecha del teclado y con espacio se produce el salto.</p> <p>Desaceleración: Velocidad de frenado del personaje.</p> <p>Desplazamiento angular:</p> <p>Objeto a rotar:</p> <p>Permitir diagonal: Define si el personaje se va a poder mover en diagonal o solo en las 4 direcciones principales.</p> <p>Velocidad de rotación:</p> <p>Velocidad máx: Velocidad que puede alcanzar el Sprite que se controla.</p>

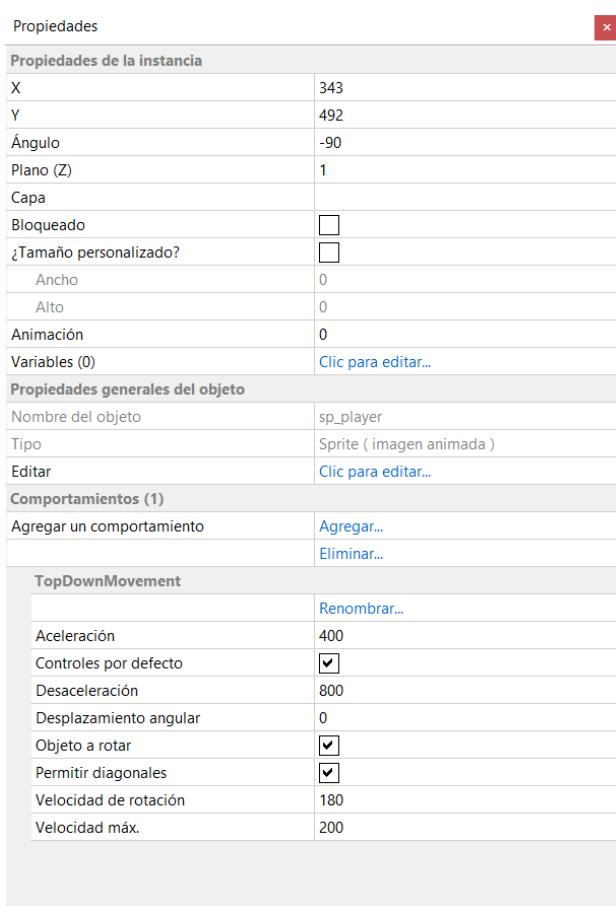


Figura 27: Gdevelop: variables movimiento en 8 direcciones

Tabla resumen funcionalidad.

Funcionalidad	Valoración	Puntuación
Movimiento 8 direcciones	Muy fácil	10

Disparo de bala.

Para que el personaje controlado pueda realizar disparo de balas debe seguir el siguiente proceso:

1. En primer lugar y antes de crear toda la lógica que va permitir la creación de balas y su movimiento en una dirección, hay que crear en el Sprite un punto adicional, los que tiene por defecto (centro y origen). Así podemos precisar el lugar exacto donde se quiere que las balas se creen, por ejemplo en el caño si se trata de un tanque o en los laterales si se trata de un avión.

Ruta/comandos	Ratón sobre el Sprite origen->Botón secundario->Editar ->En el menú visual se hace clic sobre la cruz "Editar los
----------------------	---

puntos de la imagen" ->Crear nuevo punto-> Hacer clic en la imagen y situar el punto en el lugar se quiere que sea origen de la bala ->renombrarlo para poder usarlo después.

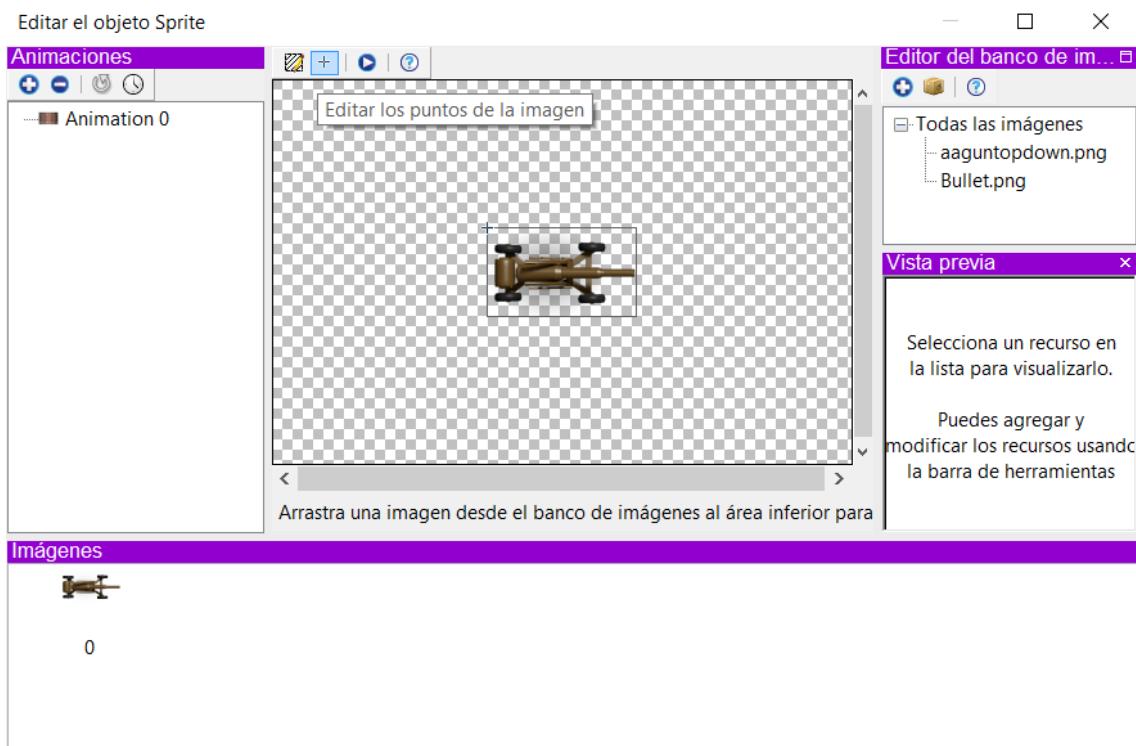


Figura 28: Gdevelop puntos de imagen

2. Creación del evento que se encargará de la creación de la bala y toda la lógica asociada. Un evento en Gdevelop consta de dos partes: la primera son **condiciones**, donde se van a especificar aquellas acciones, situaciones o estados de la escena que van a provocar que el evento comience a ejecutar, y la segunda las **acciones**, aquellos sucesos que se van a producir en la escena una vez que las condiciones especificadas suceden.

Ruta/comandos	Eventos->Creación de evento.
---------------	------------------------------

3. Se agrega la condición para que al pulsar una tecla determinada él evento de creación de la bala comience.

Ruta/comandos	Condición -> Ratón y Táctil -> Botón de ratón presionado o toque
Variables	Botón a evaluar: Botón que va desencadenar el evento.

- Para complementar la condición anterior, se tiene que añadir una condición que evite que se creen infinitas balas y prácticamente al mismo tiempo, introduciendo una restricción temporal para que las balas aparezcan en intervalos de tiempo. Para ello se agrega al evento la siguiente condición.

Ruta/comandos	Condición-> Cronómetro y tiempo -> Valor de un cronómetro.
Variables	<p>Tiempo en segundos: El evento se va poder producir cuando el cronómetro haya superado el tiempo introducido en este campo</p> <p>Nombre del cronómetro: Dar un nombre al cronómetro es necesario para gestionar la creación de las balas ya que una vez que este supere el valor en la variable anterior la condición se cumplirá siempre. Pero con este nombre se pueden crear modificaciones en los pasos siguientes si se necesita reiniciarlo.</p>

- Ahora hay que crear el disparo de la bala que se ejecutará una vez se cumplan las dos condiciones creadas en los pasos anteriores. Es aconsejable que dicha opción se encuentre siempre en la última posición. Antes de realizar el disparo lo primero que habrá que hacer es restablecer el cronómetro.

Ruta/comandos	Acciones ->Cronómetro y tiempo->Restablecer un cronómetro.
Variables	<p>Nombre del cronómetro: En este campo se introduciría el nombre dado en el paso anterior.</p>

6. Ahora que ya se tienen las condiciones hay que crear la bala.

Ruta/comandos	Acciones->Todos los objetos -> Objetos -> Crear un objeto
Variables	<p>Objeto: El objeto que se va crear, en este caso la bala.</p> <p>Posición X: Lugar en el ancho de pantalla donde se va crear la bala. En este campo se debe introducir el objeto generador y el punto creado en la instrucción uno, de la siguiente forma: <code>-nombre_del_objeto.X(punto_creado_instrucción_1);</code></p> <p>Posición Y: Lo mismo, pero para seleccionar a qué altura aparece la bala. <code>nombre_del_objeto.Y(punto_creado_instrucción_1);</code></p>

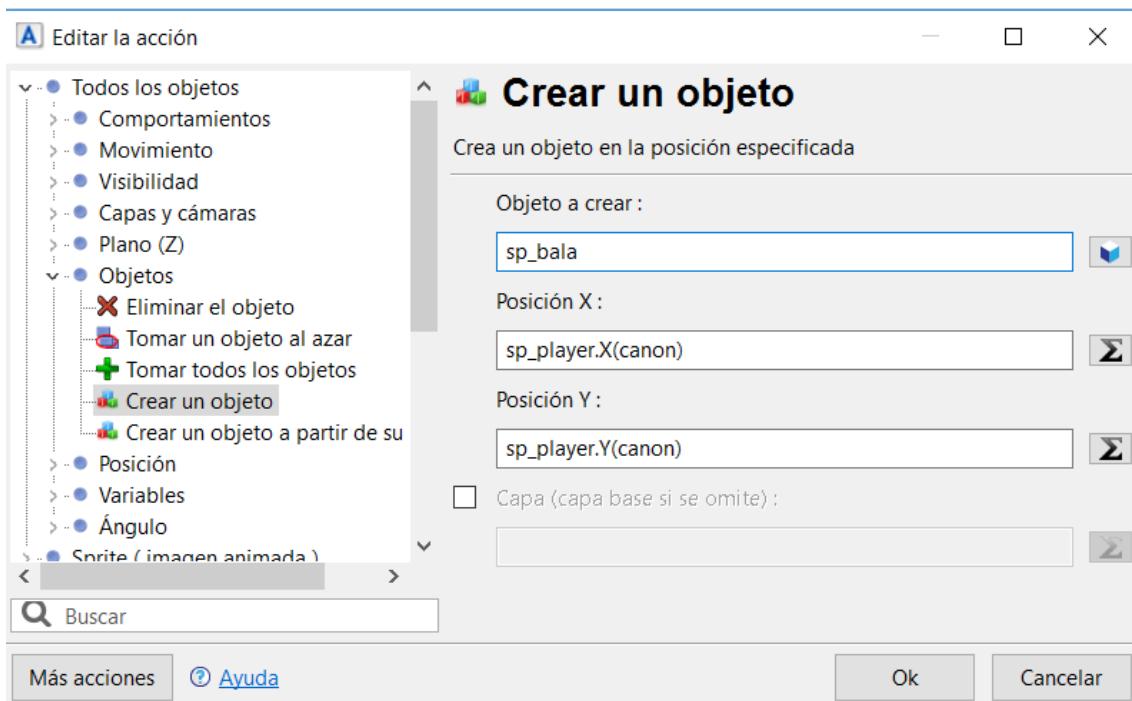


Figura 29: Gdevelop: creación de bala

7. En la instrucción anterior se ha creado la bala. En este paso se le va dar una velocidad de desplazamiento y dirección. Introduciendo la acción a continuación de la anterior.

Ruta/comandos	Acciones->todos los objetos -> movimiento -> agregar una fuerza (ángulo)
Variables	<p>Objeto: Al que se quiere dar movimiento, en este caso la bala.</p> <p>Ángulo: Hacia qué dirección se va a mover el objeto desde su creación.</p> <p>Velocidad: Se indica la velocidad en pixeles del objeto.</p> <p>Amortiguamiento: Simula la resistencia que posee el escenario a la bala. El valor 0 significa resistencia total, y cuanto mayor sea la cantidad menor será la resistencia que tendrá. Es un factor multiplicador de la</p>

	velocidad.
--	------------

8. Para una mejor gestión de la memoria y del juego en general se van a destruir las balas una vez que se encuentren fuera de la pantalla.

Ruta/comandos	Ratón sobre el objeto bala->clic derecho del ratón ->otras propiedades->agregar comportamiento ->seleccionar destruir fuera de la pantalla

Tabla resumen funcionalidad.

Funcionalidad	Valoración	Puntuación
Disparo	Difícil	2.5

Colisión con objeto.

El proceso para implementar una colisión entre objetos es el siguiente:

Ruta/comandos	Condición->Sprite -> Colisión ->Colisión(Por píxel/ Pixel Perfect).
Variables	<p>Objeto 1: Primer Sprite de la colisión.</p> <p>Objeto 2: Segundo Sprite de la colisión.</p>

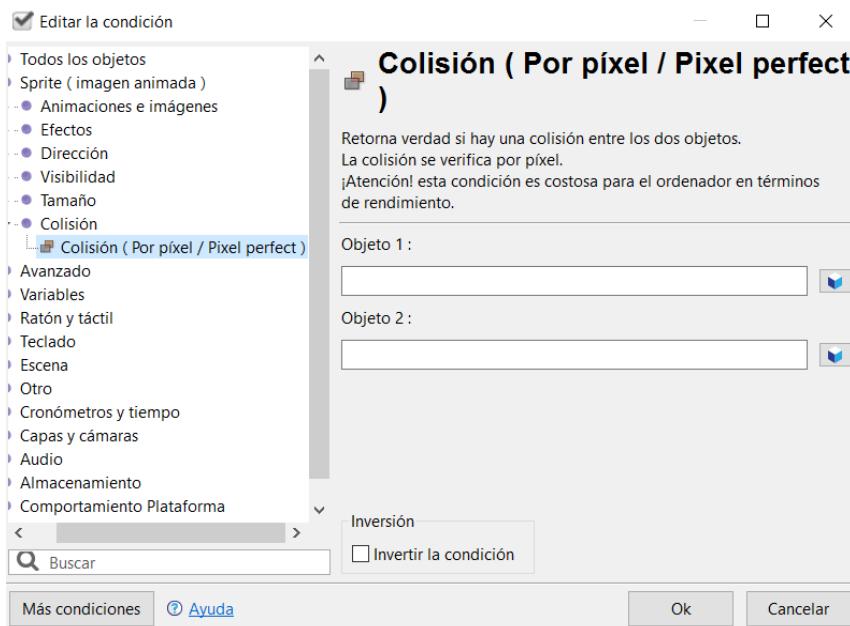


Figura 30: Gdevelop colisión por píxel

Tabla resumen funcionalidad.

Funcionalidad	Valoración	Puntuación
Colisión	Muy fácil	10

Destrucción de objeto tras colisión.

1. Crear el evento con la condición colisión como se ha explicado en el punto anterior. Los objetos de la colisión serán por ejemplo Objeto 1, y Objeto 2.
2. Añadir la acción en el evento que va a destruir uno de los objetos implicados en la condición anterior.

Ruta/comandos	Acción->Todos los objetos->Objetos->Eliminar el objeto
Variables	Objeto: El objeto que se quiere eliminar.

Tabla resumen de funcionalidad.

Funcionalidad	Valoración	Puntuación
Destrucción tras colisión	Muy fácil	10

Resumen de funciones básicas y puntuación del subcriterio.

La siguiente tabla contiene todas las funcionalidades y la dificultad y puntuación de cada una de ellas.

Función	Dificultad	Puntuación
Creación de un proyecto	Fácil	10
Insertar fondo a una escena.	Media.	5
Movimiento del personaje en 2 direcciones	Fácil	10
Salto de personaje en 2 direcciones	Muy fácil	10
Movimiento de personaje en 8 direcciones	Muy fácil	10
Disparo de bala.	Difícil	2.5
Colisión entre objetos.	Muy Fácil.	10
Destrucción de Objeto tras colisión.	Muy Fácil.	10
PUNTUACIÓN FINAL		67.5
FACTOR DE CORRECIÓN		1
PUNTUACIÓN CON FACTOR DE CORRECIÓN		67.5

Tabla 47: Gdevelop resumen de funcionalidades

Resumen subcriterio funcionalidades.

En la tabla siguiente se expone la puntuación escalada sobre 12 obtenida por Gdevelop en las funciones básicas.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Funciones básicas	0-12	10.15

Tabla 48: Gdevelop puntuación funciones básicas

Puntuación subcriterio lógica de programación necesaria

Durante el desarrollo de los tutoriales, ejemplos y pequeños proyectos, se han detectado algunos elementos que hacen referencia directa a la programación tradicional si bien el programa cuenta con ayudas visuales para facilitar que sea el usuario quien escriba estas funciones de programación. Los elementos detectados son los siguientes:

1. Para generar números aleatorios por ejemplo, para la aparición en un lugar sin predeterminar del mapa de enemigos, se tiene que introducir la función Random (número) que da un número dentro del rango introducido.
2. Para mostrar números por pantalla, estos deben ser convertidos a cadena de texto y para ello Se debe usar la función ToString que está presente en la mayoría de lenguajes de programación.

Estas son las dos únicas referencias a los lenguajes de programación que se han encontrado. Así la puntuación en este apartado es alta ya que no son funciones muy difíciles de aprender por un usuario y ni requieren un conocimiento muy amplio de programación.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Lógica de programación necesaria	0-4	3

Tabla 49: Gdevelop: subcriterio lógica de programación

Resumen puntuación funcionamiento y dificultad de uso

La siguiente tabla contiene las puntuaciones de todos los apartados que componen el criterio puntuación y dificultad de uso.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Funcionamiento General	0-4	4
Primeros Pasos: Creación de un pequeño proyecto.	0-4	3
Funciones básicas	0-12	10.15
Lógica de programación necesaria	0-4	3
TOTAL		20.15

Tabla 50: Gdevelop criterio funcionamiento y facilidad de uso

Criterio 4: Editor Gráfico/Visual.

En la siguiente tabla se especifican las puntuaciones que ha obtenido Gdevelop en los distintos apartados de la encuesta dedicada a valorar la interfaz gráfica y visual.

Concepto	Puntuación Máxima	Puntuación minima	Media (Sobre 5)
Rapidez al iniciarse	5	3	4,125
El programa se muestra rápido	5	3	4,25
La interfaz es atractiva	5	2	3,375
Tiene un aspecto cuidado y moderno	5	2	3,625
Puntuación diseño del programa	5	2	3,625
Los iconos representan bien los elementos Y/o funciones	5	3	4,25
A priori parece sencillo de usar	5	2	3,625
Consumo de recursos	5	3	3,75
MEDIA TOTAL		3,85/5	
PUNTUACIÓN PONDERADA		6.15/8	

Tabla 51: Resultado encuestas sobre interfaz

Una vez realizada la ponderación de la puntuación para adaptarla a la importancia del criterio la puntuación es la que muestra la siguiente tabla.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Editor Gráfico/Visual	Clave	6.15

Tabla 52: Criterio: Editor gráfico/visual

Criterio 5: Escalabilidad.

Puntuación subcriterio posibilidad de creación de funciones, scripts y/o lenguaje de programación propio.

Gdevelop no dispone de un lenguaje de programación propia y no provee, como si hacen otros frameworks, de un sistema para introducir código dentro del programa. La única posibilidad de aumentar las funcionalidades base de Gdevelop consiste en la modificación directa de los archivos de código para modificar su funcionamiento o crear copias de los plugins existentes para crear los propios. Es una tarea compleja y con ninguna documentación al respecto con lo que la mayoría de usuarios no sabrán realizar esta tarea.

Por lo expuesto anteriormente las posibilidades de creación de funciones y scripts propios son bastante limitadas y complejas por lo que como indica la tabla siguiente la puntuación de este apartado es muy baja.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Posibilidad de crear funciones y scripts propios y/o lenguaje de programación propio.	0-8	1

Tabla 53: Gdevelop: Subcriterio Posibilidad de creación de Scripts propios

La distribución de Gdevelop incluye una serie algunas extensiones que incluyen funciones tales como un sistema de partículas, incluir Box3D en juego o un motor de red. Pero fuera de estas extensiones Gdevelop no dispone de add-ons oficiales que puedan extender las funcionalidades que el programa tiene de serie.

Al ser de código abierto y posibilitar que usuarios modifiquen, añadan o mejoren las funciones que el programa trae de serie por lo que existen algunos plug-ins no oficiales desarrollados por la comunidad, pero por la dificultad instalación para el usuario, la no garantía de que funcionen correctamente con la versión se tiene instalada del programa y las incompatibilidades entre los mismos, hacen que no se incluyan en esta sección.

La puntuación en este apartado por lo que acabamos de ver es cero ya que no se encuentran plug-ins sencillos de instalar, confiables y compatibles entre ellos.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Add-ons y plugins	0-6	0

Tabla 54: Gdevelop: Subcriterio Add-ons y plugins

Resumen puntuación Escalabilidad.,

La tabla siguiente muestra el resumen de la puntuación del criterio escalabilidad.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Posibilidad de crear funciones y scripts propios y/o lenguaje de programación propio.	0-8	1.5
Add-ons y plug-ins	0-6	0
TOTAL		1.5

Tabla 55: Gdevelop: puntuación criterio escalabilidad

Criterio 6: Resultados profesionales.

El carácter de software libre y abierto de Gdevelop hace que sea muy complicado determinar que juegos comerciales se han realizado con Gdevelop. Derivado de este carácter de software libre no es necesaria ningún tipo de atribución. Además el tipo de licencia permite la modificación total del programa, por lo que los desarrolladores que trabajan con Gdevelop lo hacen con sus propias versiones modificadas y adaptadas a sus desarrollos por lo que es prácticamente imposible determinar que juegos comerciales han sido realizados con Gdevelop.

Aún si se consiguiera una lista de juegos creados con Gdevelop nos encontraríamos con una nueva dificultad todavía mayor. Ya que habría que examinar caso por caso las dimensiones de las modificaciones para determinar si se puede considerar todavía Gdevelop o estamos ante una versión tan customizada que ya estamos en presencia de un motor nuevo basado en Gdevelop .

Por lo comentado anteriormente solo se pueden tener en cuenta los juegos que la propia web de Gdevelop indica que han sido desarrollados con el programa. Respecto a estos están lejos de tener un resultado profesional y se tratan de pequeños juegos de navegador y que no se pueden catalogar de grandes proyectos, por lo tanto, se puede concluir que con el programa original sin modificaciones no existen juegos comerciales.

Visualmente los juegos se alejan mucho de ser excesivamente atractivos y no resultan potentes ni gráficamente ni artísticamente, quedando en ese aspecto bastante limitados y no profesional.

Puntuación subcriterio principales juegos desarrollados

Como se ha comentado anteriormente es prácticamente imposible determinar que juegos comerciales han sido desarrollados con Gdevelop y los que indica en la web oficial del programa que han sido creados usando el programa están lejos de ser juegos verdaderamente profesionales y comercialmente atractivos, con lo que la puntuación en este apartado es la mínima posible.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Principales juegos desarrollados	0-8	0

Tabla 56: Gdevelop: subcriterio Principales juegos desarrollados

Puntuación subcriterio diferenciación

Por el contrario, pese a que visualmente los resultados no son buenos y no son juegos comercialmente potentes, si hay cierta diferenciación entre ellos y no se detectan patrones visuales comunes.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Diferenciación	0-4	2

Tabla 57: Gdevelop subcriterio principales juegos desarrollados

Resumen puntuación resultados profesionales.

En la siguiente tabla se muestra el resumen de la puntuación el criterio resultados profesionales.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Principales juegos desarrollados.	0-8	0
Diferenciación	0-4	2
TOTAL		2

Criterio 7: Estado actual.

Gdevelop se encuentra en estado activo, en constante desarrollo y con equipo responsable trabajando en todo momento en la creación de nuevas funcionalidades.

Desde la web oficial se puede acceder al panel de trabajo interno en Trello¹¹, Figura XX, donde se puede ver las funciones y mejoras en las que están trabajando actualmente, las ya implementadas, las que van a desarrollarse próximamente e ideas sobre próximas actualizaciones.

Tabla 58: Trello desarrollo Gdevelop

Con lo que no solo tenemos un flujo constante de actualizaciones, mejoras y corrección de errores, sino que también tenemos toda la información de las inminentes y futuras.

Estado	Activo
Última versión	4.0.94 Noviembre 2016
Frecuencia de actualizaciones	Constantes.

Tabla 59: Gdevelop: estado y actualizaciones

¹¹ Trello es un software de administración de proyectos con interfaz web, cliente para iOS y android para organizar proyectos

Resumen puntuación criterio Estado actual

Como se ha visto, las actualizaciones son constantes y los desarrolladores están trabajando continuamente en agregar funcionalidades al programa, por lo que la puntuación en los subapartados que componen este criterio es máxima. El programa se encuentra dentro de su ciclo de vida y activo.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Estado actual	0-2	2
Última versión	0-1	1
Fecha desde la última actualización.	0-1	1
TOTAL		4

Tabla 60: Gdevelop: estado actual

Criterio 8: Sistemas operativos soportados y requerimientos.

Las posibilidades de exportación de Gdevelop son bastante limitadas, no por el número de sistemas soportados, ya que aunque son menos que otros programas cubren los principales sistemas, sino porque se ven condicionadas por el sistema operativo en el que estemos desarrollando.

Gdevelop cuenta con las siguientes opciones de exportación:

- **Exportar a Web:** Exporta el juego para ser reproducido en la web bajos los standards de HTML5.
- **Exportar a ejecutable nativo:** Exporta el proyecto para que pueda funcionar en el sistema en el que el juego ha estado desarrollándose, es decir si se está trabajando con la versión de Windows el juego que con esta opción solo funciona en Windows.

Hay que añadir también que la exportación a Android e iOS de forma oficial ésta en fase de prueba y aunque se puede realizar, primero realizando una exportación a Web y posteriormente con programas externos como Intel XDK* crear los paquetes Apk e iOS. No se trata de una creación del paquete directa, sino la conversión de un contenido web a otro adaptado para dispositivos móviles. Además, el proceso resulta bastante difícil para el usuario medio y puede provocar errores e incompatibilidades que no puede controlar.

La siguiente tabla muestra las posibilidades de exportación de Gdevelop a los diferentes sistemas y plataformas.

Sistema/Plataforma	Exportación.
Windows	Si
Linux	Si
iOS	Si ¹²
Android	Si ¹³
HTML 5	Si
Flash	No
Windows Store (Mobile y Store escritorio)	No

¹² Siempre que el juego se haya desarrollado en dicha plataforma

¹³ A través de programas externos

Sistemas PlayStation (PS3, PS-VITA, PS4)	No
Sistemas Xbox (XBOX360, XBOX ONE)	No
Sistemas Nintendo (3DS, Wii, WiiU)	No
RESUMEN Y PUNTUACIÓN PONDERADA	
TOTAL SISTEMAS	4 DE 10
PUNTUACIÓN PONDERADA	4

Resumen puntuación sistemas operativos soportados y posibilidades de exportación.

Como se ha podido comprobar Gdevelop se puede exportar a multitud de sistemas y plataformas, pero las limitaciones a la hora de realizar dicha exportación, ya que obliga a que el proyecto haya sido realizado con el sistema en el que se desea que el juego funcione, hace que la puntuación este sentido no sea excesivamente alta, como muestra la siguiente tabla.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Posibilidades de exportación: Sistemas y plataformas	0-8	4
TOTAL		4

Tabla 61: Gdevelop puntuación exportación

Criterio 9: Licencia.

GDevelop es un software de código abierto por lo que no tienen ningún coste para el usuario y además puede ver o modificar si tiene los conocimientos necesarios el código fuente del mismo.

Al programa se le aplican varios tipos de licencia dependiendo de la parte del mismo:

- **la IDE:** Se le aplica una licencia GPL v3, que permite usar, estudiar, compartir y modificar libremente el software siempre que estas versiones modificadas conserven la misma licencia que el programa principal.
- **Librería del código fuente, la de los módulos de exportación nativa y a HTML:** Estas partes del programas están protegidas con la licencia LGPL v3*, que permiten su libre distribución y copias pero no su modificación.
- **Extensiones:** La licencia de las mismas es la licencia zlib/libpng, que permiten modificar y utilizar el software para cualquier fin sea comercial o no, pero el software no se debe falsificar la autoría de las partes originales y las versiones alteradas tienen que estar identificadas como tal dentro del código.

Gdevelop no tiene módulos externos para la exportación a otros sistemas como el resto de programa a la venta.

Resumen puntuación licencias.

Gdevelop es un software libre que usa diferentes tipos de licencia, todas libres, para cada parte del mismo. Por lo que pone a disposición de los usuarios el programa, todos los módulos de exportación de manera gratuita y sin ninguna restricción y limitación por lo que la puntuación en este apartado es la máxima posible como muestra la tabla siguiente.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Precio de Licencias	0-4	4
Modo de prueba o free	0-2	2
Precio módulos de exportación	0-2	2
TOTAL		8

Tabla 62: Gdevelop: puntuación licencias

Puntuación final

Después de un análisis de Gdevelop bajo los criterios definidos el marco teórico obtiene una puntuación de 66,8 sobre 100. En la tabla siguiente se detalla la puntuación en criterios evaluables.

Estos resultados serán analizados y comparados con el resto de frameworks en el apartado conclusiones del proyecto.

Criterio	Puntuación máxima	Puntuación obtenida	Puntuación porcentual
Requisitos e instalación	6	6	6
Documentación, tutoriales y comunidad.	16	12,5	16
Funcionamiento y dificultad de uso	24	20,15	24
Editor Gráfico/Visual	8	6,15	8
Escalabilidad	14	1,5	14
Resultados profesionales	12	2	12
Estado actual	4	4	4
Sistemas operativos soportados y requerimientos.	8	4	8
Licencias	8	8	8
TOTAL	100	64,3	

Tabla 63: Gdevelop puntuación total

5.2.3. Análisis Construct 2.

Criterio 1: Requisitos e instalación.

Los requisitos son bastante elevados en comparación con otros frameworks de creación de videojuegos, sobre todo en cuanto a la potencia del proceso y la memoria. Además no es posible usar Construct 2 en sistemas operativos distintos de Windows aunque ésta anunciada una versión compatible con Mac, Linux e IOs aún no ha sido lanzada.

REQUISITOS GDEVELOP	
Sistema Operativo	Mínimo Windows XP
Procesador	Procesador 1 Ghz
Memoria	2GB de RAM
Tarjeta Gráfica	Mínimo Tarjeta Gráfica con soporte para Direct3D 9
Otros	Conexión internet necesaria para la instalación, actualización y descarga del software.

Tabla 64: Gdevelop Requisitos de instalación

Puntuación subcriterio requisitos

Con lo expuesto anteriormente y los datos de la tabla la puntuación en este apartado para Construct 2 es bastante baja y se detalla en la siguiente tabla.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Requisitos de instalación.	0-6	2

Tabla 65: Construct 2: subcriterio requisitos

Para la instalación de Construct 2 el proceso es el siguiente:

1. Descargar el paquete de instalación desde el apartado *download* de la web de Scirra.¹⁴
2. Ejecutar el instalador y seguir los pasos para completar la instalación. En un momento determinado de la instalación el programa, el usuario debe indicar la si su ordenador es de 32bits o 64bits para facilitar este paso el instalador tiene la opción de auto detectar este parámetro, véase Figura .

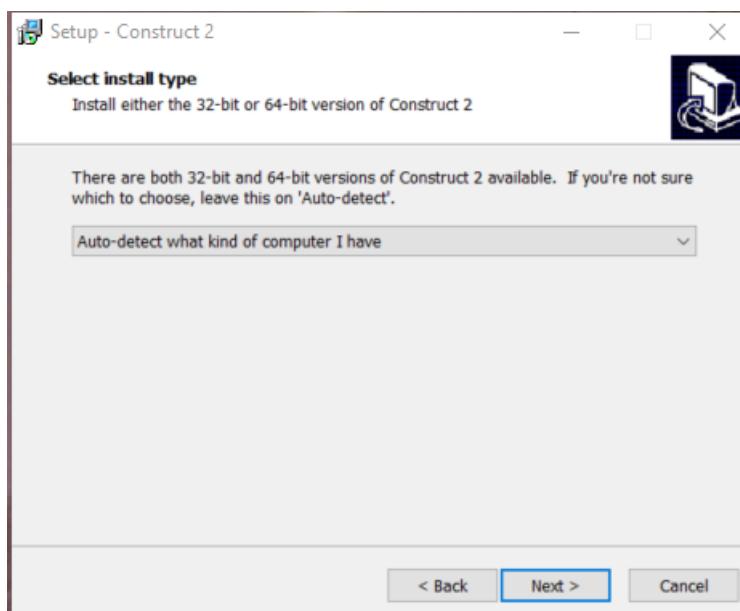


Figura 31: Construct 2 Instalación

3. El programa en el equipo de prueba ha tenido un tiempo de instalación de 1 minuto y 55 segundos y no ha necesitado ninguna librería adicional para su funcionamiento.

Construct 2 se puede instalar también a través de Steam¹⁵.

Resumen proceso de instalación

La tabla siguiente muestra el resultado de la instalación y el resumen de los datos obtenidos.

¹⁴ <https://www.scirra.com/construct2/releases/r239/download>

¹⁵ Steam es una plataforma de distribución digital, gestión digital de derechos, comunicaciones y servicios multijugador desarrollada por Valve Corporation.

Construct 2 Instalación en el equipo de pruebas	
Tamaño de descarga	57 Mb
Requiere programas adicionales	No
Instalación de módulos o recursos adicionales durante la instalación	No
Tiempo total	1 minutos 54 segundos.
Tamaño posterior a la instalación	97 Mb

Tabla 66: Resumen proceso de instalación Gdevelop

Datos de instalación encuesta.

Los resultados de la encuesta de instalación de Construct 2 se resumen en la tabla siguiente. Si se desea visualizar los datos en profundidad se pueden encontrar en el Anexo I del presente documento.

Gdevelop datos estudio instalación	
Tiempo medio	2 minutos 12 segundos
Valoración media sencillez del proceso de instalación (0 a 10)	7.75
Valoración media de la funcionalidad del proceso de instalación (0 a 10)	8.375
Puntuación media funcionalidad y sencillez	8.06
Fallos en instalación.	No

Tabla 67: Gdevelop datos encuesta instalación

Puntuación subcriterio instalación.

Con los baremos de puntuación establecidos en el marco teórico para el criterio de instalación y con la tabla presentada anteriormente se puede establecer la puntuación de Construct en este apartado.

El programa ha tardado menos de 5 minutos en completar su descarga e instalación, por lo que le corresponde un punto.

La media sencillez y funcionalidad del proceso de instalación es de 8.06 con lo que le corresponde 0.66 puntos.

No ha necesitado paquetes adicionales y el proceso de instalación ha funcionado correctamente para todos los usuarios con lo que no tiene penalizaciones en ese sentido.

La siguiente tabla muestra el resultado de la puntuación de subcriterio instalación.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Instalación.	0-2	1.66

Tabla 68: Gdevelop subcriterio instalación

Resumen puntuación requisitos e instalación.

La siguiente tabla muestra el resumen de la puntuación de Construct 2 en el criterio requisitos e instalación.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Requisitos	0-6	1.5
Instalación	0-2	1.66
TOTAL		3.16

Tabla 69: Construct 2 criterio requisitos e instalación

Criterio 2: Documentación, tutoriales y comunidad.

En la web oficial de Scirra hay disponible un extenso manual de uso del programa, que además se puede descargar para poder visualizarlo sin necesidad de disponer de una conexión a internet.

En este manual se intenta explicar todos los aspectos del programa y quiere ser una herramienta de consulta ante dudas o problemas surgidos durante el desarrollo de proyectos con Construct.

El manual ésta dividido en las siguientes secciones.

1. **Installing:** Se explica el proceso de descarga, instalación y configuración inicial de Construct. Este apartado tiene un total de 5 subsecciones.
2. **Overview:** Este apartado da a los usuarios que quieran iniciarse en el programa todas las explicaciones necesarias para desenvolverse en la interfaz, entender la estructura de los proyectos, la exportación de los juegos creados o la una guía de buenas prácticas uso, cuenta con 10 subapartados.
3. **Interface:** En el apartado anterior se daba una primera aproximación a la interfaz del programa, este se dedica totalmente a ella. Se explica en detalles todos sus elementos, sus funcionalidades y sus herramientas. En sección se puede encontrar subsecciones como la dedicadas a explicar la barra de propiedades de los objetos, el editor de imágenes, los atajos de teclado o el modo debug. El apartado interfaz es uno de los más extensos ya que cuenta con un total de 36 subsecciones.
4. **Project Primitives:** Las primitivas del proyecto, son los elementos básicos a partir de los cuales los juegos se van construyendo. Los puntos fundamentales de este apartado los constituyen los objetos y el sistema de eventos, cada uno de ellos con varios subapartados. Las subsecciones que componen este apartado del manual son 27.
5. **Behavior Reference:** Construct incorpora tanto para los objetos, como para los personajes que el jugador controla varios comportamientos y funcionalidades predefinidas. En esta sección se explican y se da una guía del uso de cada uno de ellas así como de sus parámetros de configuración. Se encuentran en este apartado elementos como la explicación del movimiento en 8 direcciones, el comportamiento perseguir de los enemigos o el scroll. Las subsecciones dedicadas a explicar los comportamientos y funcionalidades predefinidas son 26.

6. **Plugin Reference:** Al programa base se le pueden añadir plugins y complementos que aumentan o mejoran sus funcionalidades base. En el apartado Plugin Reference del manual se trata los plugins oficiales, es decir aquellos desarrollados directamente por Scirra o que por su gran uso o importancia tienen esta consideración. Entre estos añadidos oficiales se puede encontrar por ejemplo, algunos dedicados a la integración con plataformas de publicación de juegos como Facebook, Google Play o Windows Store, otros que permiten el uso de lenguajes de programación el desarrollo como XML o AJAX y por último los que aumentan las funcionalidades como un sistema de partículas o el soporte para multijugador. En manual tiene 48 subsecciones dentro de este apartado.
7. **System Reference:** Explicaciones de funciones avanzadas como el lenguaje de programación que usa Construct, los tipos de variables o las funciones del sistema. Solo tres subapartados se incluyen dentro de System Reference.

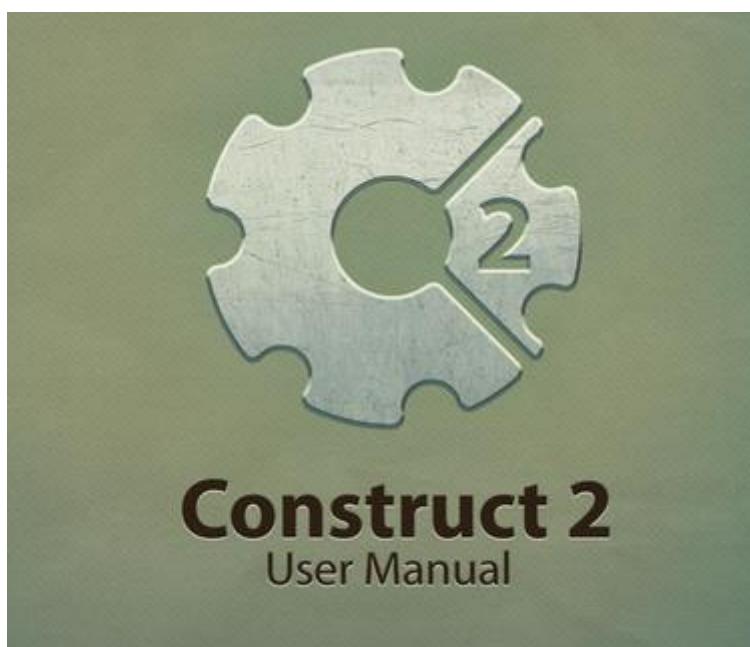


Figura 32: Construct 2 manual de usuario

La web de Scirra cuenta también con una sección dedicada a tutoriales y guías que se pueden filtrar por su dificultad, por el tipo de tutorial y por idioma. Cualquier usuario puede crear sus tutoriales e incluirlo en esta sección. La cantidad de manuales es enorme llegando a una cifra superior a 1200.

Entre esta gran cantidad de tutoriales disponibles hay muchos desarrollados por la propia Scirra y que por lo tanto se pueden considerar oficiales, son estos los que se analizarán en el punto siguiente.

No existe ningún filtro que permita separar estos del resto para saber que es un manual oficial hay que fijarse en el ícono que acompaña al manual o entrar en el

perfil del autor, véase la Figura 21, donde se indicará su condición de administrador y su pertenencia a Scirra.



Figura 33. Perfil oficial tutoriales Construct 2

En la web de Scirra tenemos disponible un completo foro oficial. En él, los desarrolladores y usuarios de Construct comparten tanto sus creaciones, como sus dudas a la hora de implementar funcionalidades concretas.

El foro tiene una gran cantidad de actividad, hilos y estos a su vez se encuentran organizados en categorías.

Una de las secciones más interesantes, tanto para usuarios que desean iniciarse como para usuarios expertos, es "How do I", véase Figura 22. En esta sección todos los hilos comienzan con una pregunta sobre cómo realizar una acción o implementar una funcionalidad y el resto de usuarios tratan de dar solución al requerimiento inicial, aportando distintas formas de llegar a la solución lo que hace esta sección tan interesante , no solo para usuarios novatos.

Forum	Topics	Posts	Last post
Beginner's Questions Just started using Construct 2? Post your questions here	2,408	11,977	Re: how to make a layer visible and... Last post by ramyaswetha
Post a new topic <input type="text" value="Search this forum..."/> <input type="button" value="Search"/>			
Announcements	Replies	Views	Last post
Dropbox has changed, what it means for our forums by Kyatric » Thu Mar 16, 2017 2:47 pm	8	1,087	Last post by SoldjahBoy Tue Mar 21, 2017 9:58 am
Topics	Replies	Views	Last post
[How do I]Frequently asked questions by Kyatric » Thu Oct 20, 2011 1:02 pm	181	1,256,535	Last post by thisisblogs Mon Mar 13, 2017 5:37 pm
C2 Academy Tutorial Videos by ArcadEd » Mon Feb 09, 2015 3:35 pm	33	31,663	Last post by imothep85 Fri Feb 03, 2017 10:19 am
Tips for posting in the "How do I?" forum by Tom » Sat Jan 10, 2011 1:09pm	4	201,791	Last post by lilfootsie Sun Sep 11, 2011 5:50 pm

Figura 34: Sección How do I, foro Construct 2

Puntuación subcriterio documentación, foros y tutoriales oficiales.

Como se ha visto, Construct 2 cuenta con un extenso manual de usuarios, que además podemos descargar, una cantidad prácticamente inagotable de manuales muchos de ellos en castellano. Además el foro oficial está constantemente activo y tiene secciones de todo tipo y para usuarios de todos los niveles. Por lo tanto la puntuación de Construct 2 en este criterio es máxima.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Documentación, foros y tutoriales oficiales	0-6	6

Tabla 70: Construct 2 subcriterio ducumentación oficial

Tutoriales oficiales realizados.

Tutorial How to make a Platform game.

Este tutorial se encuentra dentro de los de la categoría beginners en la web de Scirra. Es un de los tutoriales que se recomienda realizar en primer lugar. La propuesta es crear un prototipo de un juego de plataformas, véase Figura 23.



Figura 35: Resultado buscado tutorial How to make a Platform game

El tutorial se presenta en formato online y no proporciona los Sprites para poder realizarlo, lo que conlleva una pérdida de tiempo importante en buscarlos. En este tutorial sería importante tenerlos ya que se trata de un tutorial con muchos apartados dedicados a animaciones y manejo de tilesets.

El tutorial destaca en los siguientes aspectos:

1. El tutorial en la mayoría de sus partes explica muy bien todas las acciones, no limitándose a indicar que pongas un valor o función, sino que da las razones de esas acciones y de los valores que se dan. Esto es muy importante sobre todo en la parte que añade eventos relacionados con vectores. Ayuda al usuario a relacionarse con estos conceptos complejos y entenderlos.
2. Añade pequeños detalles que dan al prototipo un aspecto más profesional y que al usuario le aportan una sensación de estar creando algo divertido y jugable desde el primer momento. Un ejemplo de este tipo de pequeños elementos lo encontramos en el hecho de añadir un efecto rebote al matar a un enemigo, como hace la mayoría de plataformas.
3. Da al usuario mucha libertad creativa, presentándole los elementos, creación de personaje, de plataforma, de enemigos y dejando que este se ocupe del diseño de fase y de la distribución de estos elementos.
4. Al finalizar se añade el tutorial un apartado de conclusiones con un resumen de todo lo aprendido y de las cosas más importantes de las funcionalidades creadas.

Pese a tener todos aspectos positivos también algunos elementos negativos que hacen que tutorial no sea todo lo bueno que se podría esperar. Estos elementos negativos son los siguientes:

1. Algunas opciones son demasiado avanzadas como la creación de animaciones a través de tilesets para un tutorial del mínimo nivel.

2. Omite el paso de crear elementos tipo sólidos para el movimiento tipo plataforma con lo que no es posible completar esta parte si no se busca información o se conoce el funcionamiento del movimiento.
3. Para ser un tutorial para usuarios que empiezan a usar Construct demasiado enfocado a las animaciones algo que no es lo principal en el aprendizaje de la herramienta.
4. La parte que explica el movimiento tipo flash al realizar la colisión con el enemigo no funciona.
5. El tutorial acaba de una manera incompleta, no se refiere este punto a la parte de creatividad comentada en los aspectos positivos, sino a elementos como que el jugador no pueda morir o no haya finalización de fase.

Para subsanar algunos de estos aspectos negativos del tutorial y mejorar otros presentes se han realizado algunos cambios en el tutorial.

1. Se ha añadido un movimiento tipo Scroll centrado en el personaje, para que la cámara esté en todo momento centrada en él.
2. En el tutorial los enemigos no hacían ninguna acción sobre el personaje solo se movían y este podía destruirlos. Se ha creado las funciones para que estos eliminen al personaje. Para ello se ha creado un evento que determina que si se produce la colisión y que el personaje no esté cayendo, véase Figura 24, se elimine al jugador.



Figura 36: Funcionalidad enemigos destruyen a personaje

3. Como se ha comentado uno de los puntos negativos del tutorial es que no está completo. Se han realizado una serie de acciones para que lo esté:
 - a. Se ha creado un nuevo layout con el mensaje "Has ganado".
 - b. Se ha añadido al jugador una variable "enemigos" que al destruir un enemigo aumente en una unidad.
 - c. Se ha creado un objeto cuadrado al final del nivel que al colisionar con el personaje, hace que se compruebe la variable "enemigos" del jugador y si es igual al número de enemigos totales cambie de layout y al nuevo creado en el paso a.

- d. El jugador al ser destruido genera un evento que reinicia la pantalla.
- e. Se ha añadido un Sprite que hace de agua que al colisionar con el jugador lo destruye.

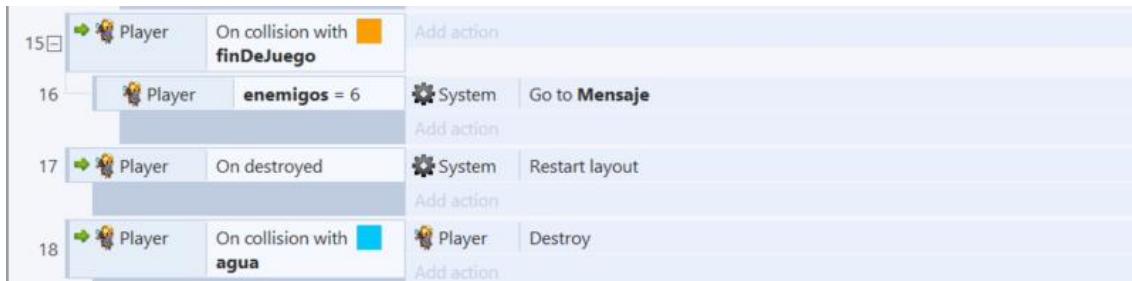


Figura 37: Nueva condiciones tutorial Construct 2

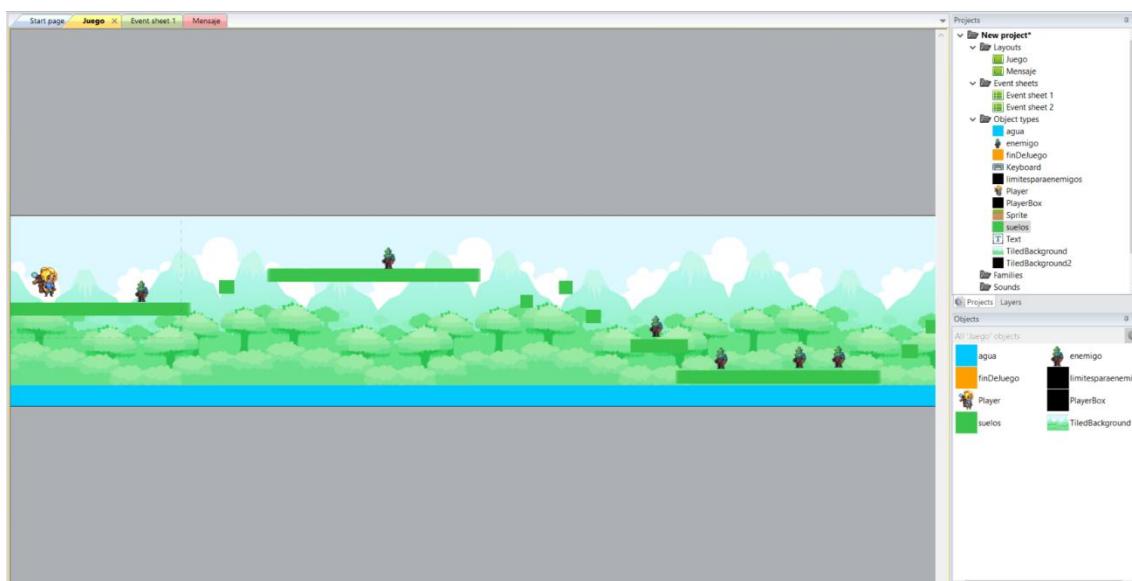


Figura 38: Gdevelop juego desarrollado tutorial 1

Tutorial Make a top-down shooter

Junto al tutorial anterior constituyen las dos recomendaciones básicas para tener una pequeña base con la que comenzar a trabajar con Construct 2.

Como sucedía con el tutorial anterior no contiene los sprites por lo que se han tenido que buscar unos compatibles, aunque por el tipo de juego que propone y la ausencia de animaciones la tarea ha sido sencilla y no tan costosa como el anterior. El tutorial pretende construir un shooter básico con vista cenital.

El tutorial es uno de los más completos, mejor explicados y sencillos para principiantes que se han realizado en este estudio. No contiene ninguna acción incorrecta o que no se pueda implementar ni tampoco funcionalidades que puedan ser dificultosas para los

primeros pasos con el programa. Es el ejemplo por perfecto de cómo debe de ser una guía de iniciación en este tipo de programas.

Al tutorial se le han añadido una serie de características para mejorarlo y hacer del mismo un juego completo, pero por la comentado anteriormente en el caso de este tutorial los cambios han sido mínimos.

Se han añadido condiciones para que las balas también se destruyan con el contacto con los enemigos o con los límites del escenario para que no se estén creando infinitas estancias de balas y sobrecarguen la memoria.

1. Para aumentar la dificultad del juego y que sea interminable, se han creado un sistema para que conforme la puntuación aumente la velocidad a la que se crean los enemigos también aumenta, véase Figura 26.



Figura 39: Sistema de eventos introducido para aumentar la dificultad

2. Se ha añadido un nuevo layout que aparece cuando el jugador muere y permite pulsando espacio volver a jugar. A su vez se reinicia la variable puntuación.

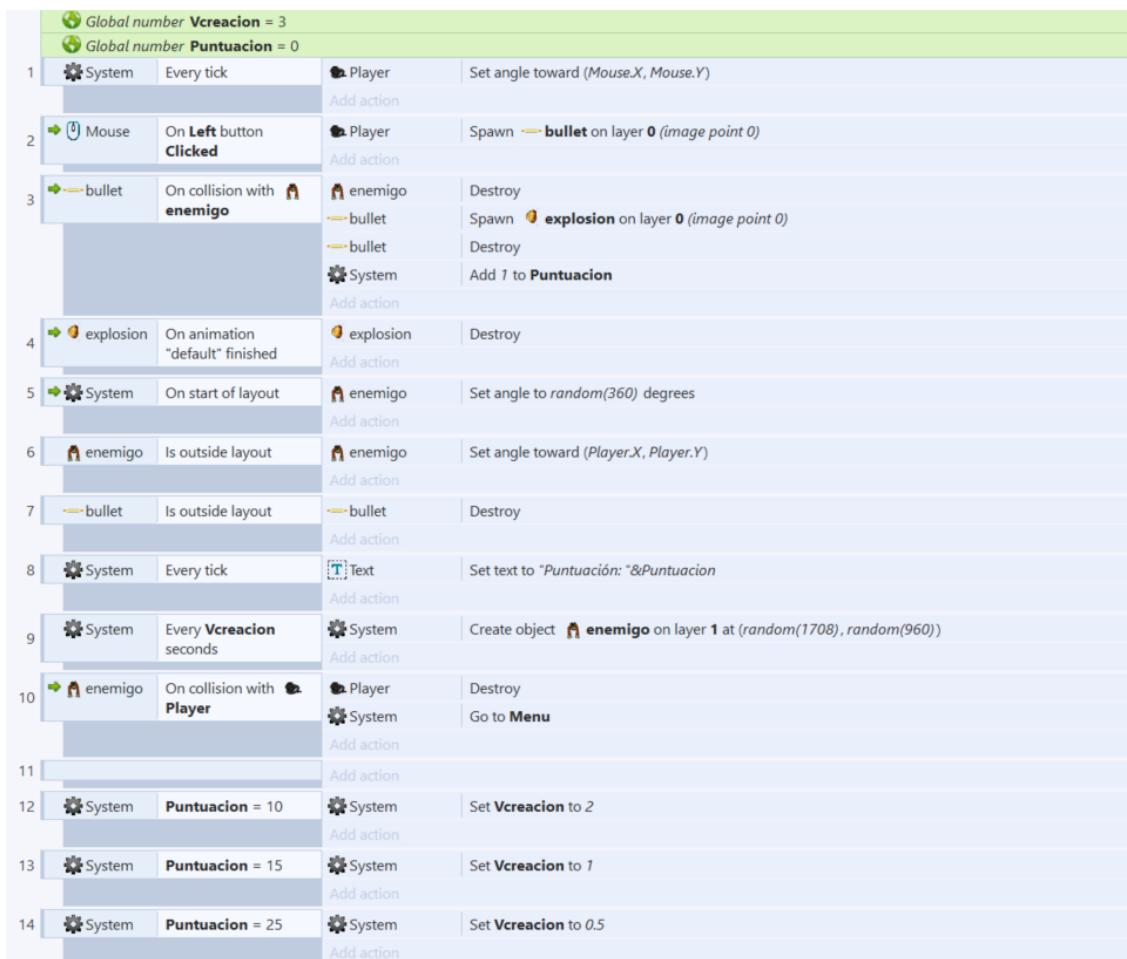


Figura 40: Eventos Gdevelop tutorial 2

Puntuación subcriterio tutoriales realizados.

La tabla siguiente contiene el resumen de los tutoriales realizados, el tiempo invertido y las funciones aprendidas durante su realización.

Tiempo dedicado.	8 horas 2 minutos
Tutoriales realizados	2
Tutoriales completados	2
Juegos completos	2
Errores detectados en los tutoriales	Muy pocos
Modificaciones realizadas	Pocas y accesoria

Funcionalidades aprendidas en los tutoriales	<ul style="list-style-type: none"> • Mover personaje en dos direcciones. • Movimiento en 8 direcciones. • Colisiones. • Insertar fondo a la escena. • Creación aleatoria en posición de objetos. • Movimiento de caída por la gravedad de objetos. • Mostrar puntuación en pantalla y sumar. • Crear movimiento de bala. • Movimiento del ángulo de personaje con el ratón. • Utilizar el ratón para funcionalidades del personaje como disparar. • Destruir enemigos con impacto de bala. • Explosión tras destrucción de objeto u enemigos. • Aumentos y disminuciones de vida del personaje y de los enemigos. • Gestión de puntuación y mostrarla por pantalla mediante un HUD. • Crear enemigos en posiciones aleatorias cada cierto tiempo.
---	--

Tabla 71: Gdevelop resumen tutoriales realizados

Puntuación subcriterio análisis de tutoriales oficiales.

Como se puede observar en la tabla 27, los dos tutoriales se han podido completar en su totalidad y canalizándose en dos juegos completos terminados, además la cantidad de funcionalidades aprendidas con los mismos ha sido enorme que permiten que el usuario pueda empezar a crear proyectos por sí mismo prácticamente desde el primer momento.

Todo lo anterior sumado a que entre los manuales básicos de Construct 2 se encuentra uno de los mejores que se han podido analizar en este informe hacen que la puntuación sea la más alta posible en este apartado.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Análisis de tutoriales oficiales	0-8	8

Tabla 72: Subcriterio Gdevelop Análisis de tutoriales oficiales

A continuación, se detalla la presencia de Construct 2 en Reedit y Taringa.

Reddit

Más de 1500 subscriptores tiene el hilo de Reddit de Construct 2. En él se pueden encontrar todos los perfiles posibles desde desarrolladores profesionales que trabajan con Construct hasta usuarios novatos, incluyendo a los propios desarrolladores del programa.

Taringa.

Construct 2 no tiene comunidad propia en Taringa

Puntuación subcriterio comunidades y foros de desarrolladores.

Por lo expuesto anteriormente Construct dos tiene la siguiente puntuación respecto a comunidades y foro de desarrolladores.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Foros y comunidades de desarrolladores	0-2	1

Tabla 73: Subcriterio foros y comunidades de desarrolladores

Resumen de puntuación Documentación, tutoriales y comunidad.

La tabla siguiente resume la puntuación de Construct 2 en el criterio documentación, tutoriales y comunidad.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Documentación, foros y tutoriales oficiales	0-4	6
Análisis de tutoriales oficiales	0-8	7
Foros y comunidades de desarrolladores	0-2	1
TOTAL		14

Tabla 74: Gdevelop criterio documentación, tutoriales y comunidad

Criterio 3: Funcionamiento y dificultad de uso.

El funcionamiento del programa destaca sobre todo en los siguientes puntos:

1. Gran gestión de tiles- por ejemplo, con solo indicarle el número de divisiones horizontales y verticales el programa automáticamente crea las animación.
2. Cuenta con una gran herramienta de autocompletado en la selección de objetos en eventos, comparaciones y acciones.
3. Si durante la ejecución de un juego se produce un error fruto de una mala implementación de una funcionalidad o de un uso incorrecto de algún elemento del programa se muestra un mensaje explicando el porque de ese fallo. Esto ayuda mucho a usuarios nuevos para no frustrar su avance sabiendo exactamente que se está realizando incorrectamente y también a usuarios avanzados con proyectos con un gran número de objetos y funcionalidades en la escena donde el identificar que ésta produciendo el fallo ahorra mucho tiempo de desarrollo.



Figura 41: Construct mensaje de error

Puntuación subcriterio funcionamiento.

El programa funciona correctamente sin que se produjera ningún fallo durante las horas que se ha estado trabajando con él. Además, como se ha visto tiene algunos aspectos que hacen su funcionamiento mejor para sus usuarios, por lo tanto, su puntuación en este aspecto es muy alta.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Funcionamiento General	0-4	3.5

Tabla 75: Gdevelop subcriterio funcionamiento general

Puntuación primeros pasos: creación de un pequeño proyecto.

Los comienzos con Construct 2 son bastante asequibles y sencillos para los usuarios. Gracias a la multitud de ayudas tanto visuales como textuales y de autocompletado las personas que se inicien con el programa no encontraran la tarea frustrante ni excesivamente compleja.

El principal problema al que se enfrentan los usuarios a la hora de comenzar a usar Construct 2 es su uso de nombres excesivamente técnico en algunas funciones. El programa hace uso de expresiones como variables instancia, vectores o constantes que pueden resultar dificultosas para usuarios que no cuentan con ese tipo de formación o conocimiento.

Respecto a la creación de un pequeño proyecto, con Construct 2 y gracias a los detallados tutoriales se ha conseguido crear dos juegos completos en menos de 9 horas.

La puntuación en el apartado primeros paso sería máxima posible si no fuera por el pequeño inconveniente del uso de nombres técnicos al implementar algunas funcionales si bien no impide que los primeros pasos sean bastante sencillos y gratificantes para el usuario.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Primeros pasos: Creación de un pequeño proyecto.	0-4	3

Tabla 76: Gdevelop subcriterio funcionamiento general

Funciones básicas

A continuación, se va a explicar cómo implementar las funciones básicas descritas en la definición de los criterios de selección y se va a puntuar en función de su dificultad.

Creación de un proyecto.

La creación de un proyecto en Construct 2 es muy sencilla.

Ruta/comandos	File->New
Variables	Tipo de proyecto: Se tiene para elegir una gran cantidad de tipos de proyectos, véase Figura 29 , desde completamente vacíos, a otros de estilo retro, proyectos con resoluciones y para pantallas determinadas, ejemplos de juegos, hasta plantillas para juegos de tipos determinados como plataformas, juegos de físicas, shooters etcétera.

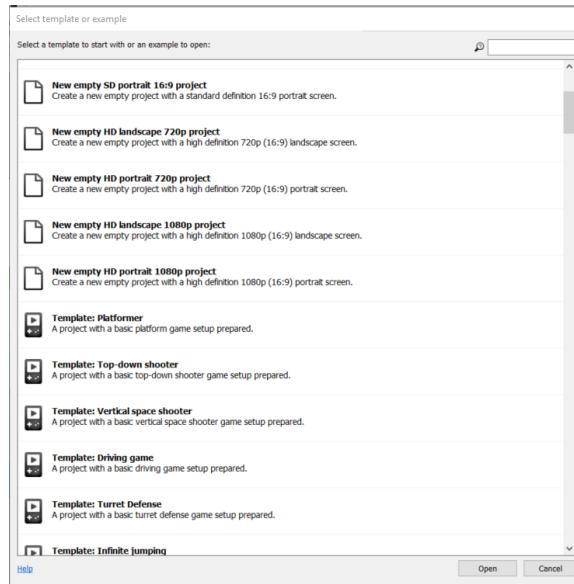


Figura 42: Selector tipo de proyecto

Visto el proceso puntuación de la valoración es la siguiente.

Funcionalidad	Valoración	Puntuación
Creación proyecto.	Muy fácil	10

Insertar fondo en la escena.

1. En primer lugar, habrá que insertar un objeto tipo Background en la escena.

Ruta/comandos	Doble click sobre la escena-> Seleccionar el objeto "Tiled Background"-> Pulsar sobre el Layout-> Ahora se abrirá el editor de imágenes seleccionar "image from file"->seleccionar la imagen
Variables	

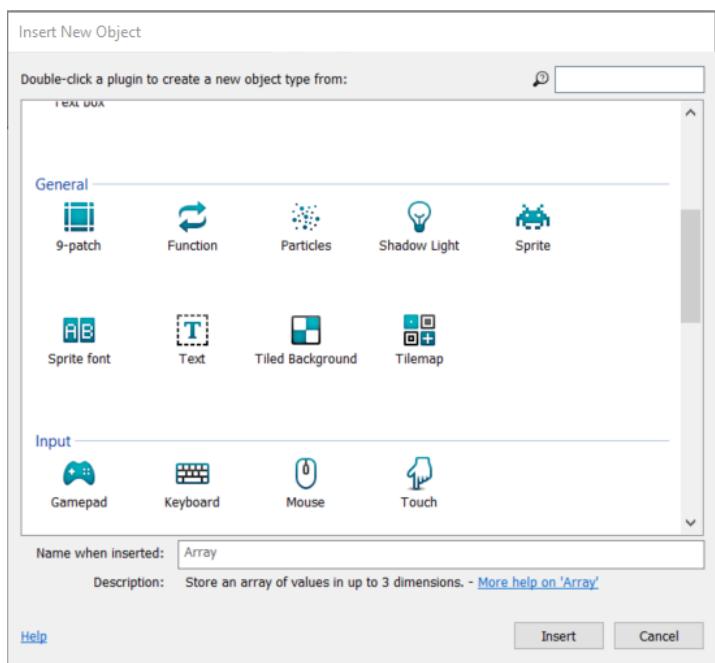


Figura 43: Construct 2: selector de objetos

- Una vez el objeto este en la escena solo quedará escalarlo y posicionarlo mediante las flechas y las guías visuales hasta que se encuentre en el tamaño y lugar deseado.

Tabla resumen funcionalidad.

Funcionalidad	Valoración	Puntuación
Insertar fondo	Muy fácil	10

Movimiento de personaje en dos direcciones.

El movimiento del personaje en dos dimensiones puede implementarse de dos formas.

Existe una forma sencilla y rápida de realizar esta acción y otra bastante más compleja añadiendo elementos que es la recomendada en todos los manuales, y que permite una mejor interacción con el entorno, mejor sistema de colisiones, mejora las animaciones y dota al juego de un acabado profesional.

A continuación, se explicarán las dos opciones aunque para la puntuación se tendrá en cuenta la sencilla.

Forma 1.

- Incluir el Sprite que será el personaje en la escena.

Ruta/comandos	Doble click sobre la escena -> en el menu "Insert Object" seleccionar Sprite -> "Image from file" en el editor de sprites-> Elegir la imagen que se quiera que sea el sprite.
----------------------	---

2. Añadir al Sprite el comportamiento tipo plataforma.

Ruta/comandos	Seleccionar el Sprite haciendo click sobre el-> En el menú izquierdo doble click en "Add Behavior" -> Pulsar sobre el icono de más -> "Apartado Movement" -> Opción Platform.
Variables	<p>Max Speed: Velocidad máxima de movimiento</p> <p>Acceleration: Velocidad a la que el personaje acelera hasta alcanzar su velocidad máxima.</p> <p>Deceleration: : Velocidad con la que el personaje desacelera hasta detenerse.</p> <p>Jump strength: Fuerza con la que el personaje salta, determina la distancia recorrida en el mismo junto con el parámetro gravedad.</p> <p>Gravity: Fuerza con que el personaje es atraído hacia el suelo.</p> <p>Max fall speed: Máxima velocidad que se puede alcanzar durante una caída desde una plataforma o después de salto.</p> <p>Double jump: Permitir o no permitir doble salto.</p> <p>Jump sustain:</p>

	<p>Default controllers: Controles por defecto del personaje o editarlos y realizar una configuración propia.</p> <p>Initial state: Determinar si el estado en el momento que el objeto se encuentra o aparece sobre la escena se está moviendo o no.</p>
--	--

3. Insertar el objeto el Sprite plataforma de la misma forma que el Sprite del jugador.
4. Dar al objeto plataforma el comportamiento tipo sólido que será el objeto por el que el Sprite se desplazará.

Ruta/comandos	Seleccionar el objeto plataforma haciendo click sobre él -> En el menú izquierdo doble click en "Add Behavior" -> Pulsar sobre el icono de más -> Apartado "Atributtes" -> Opción Solid.
Variables	Initial State: Determinar si el objeto está activo como sólido al comiendo de la escena o de su aparición o no.

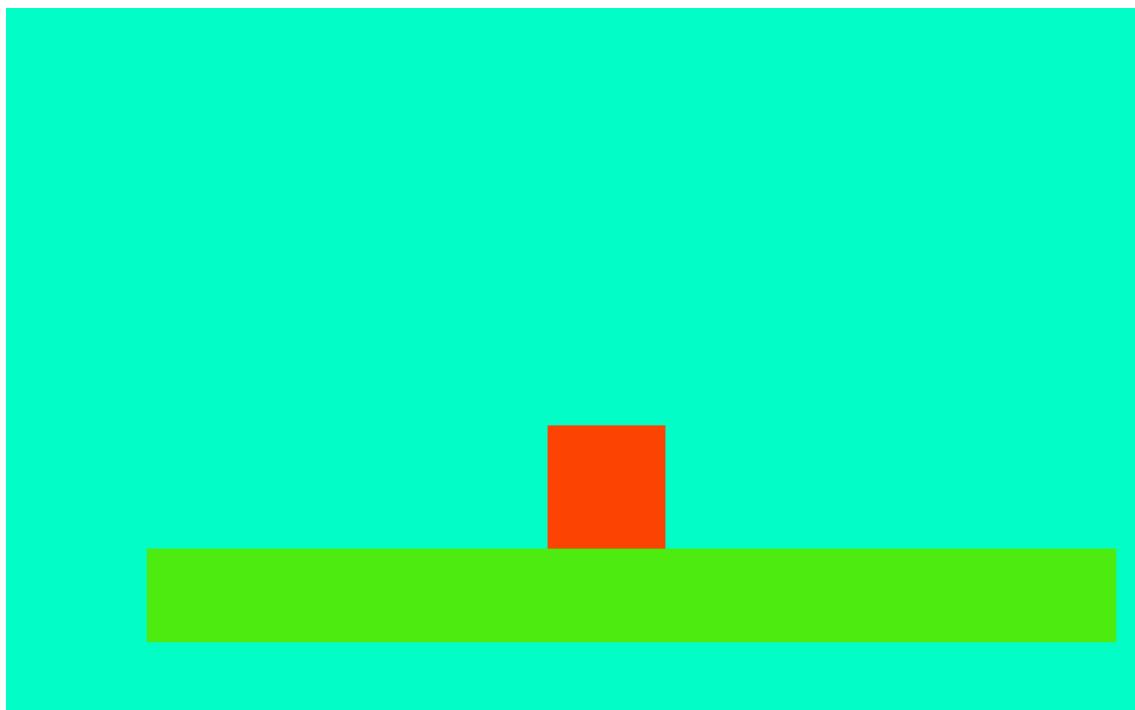


Figura 44: Construct 2: Movimiento sencillos 2 dimensiones

Forma 2: Recomendada para proyectos avanzados.

1. Incluir el Sprite que será el personaje en la escena como en la forma 1.
2. Crear otro objeto cuadrado que hará la función de box de colisiones en el movimiento del personaje

Ruta/comandos	Insertar otro Sprite como la acción anterior pero en el editor de imágenes realizar la siguiente acción: Opción Flit y pintar la imagen de un color.
Variables	

3. Ahora en la escena se encuentran dos objetos, el Sprite de jugador al que se denominara a partir de ahora Player y un cuadrado o Sprite box, véase Figura . En este paso habrá que colocar arrastrando sobre la escena el Player y la Box en el mismo lugar ajustando este último lo máximo posible al personaje.

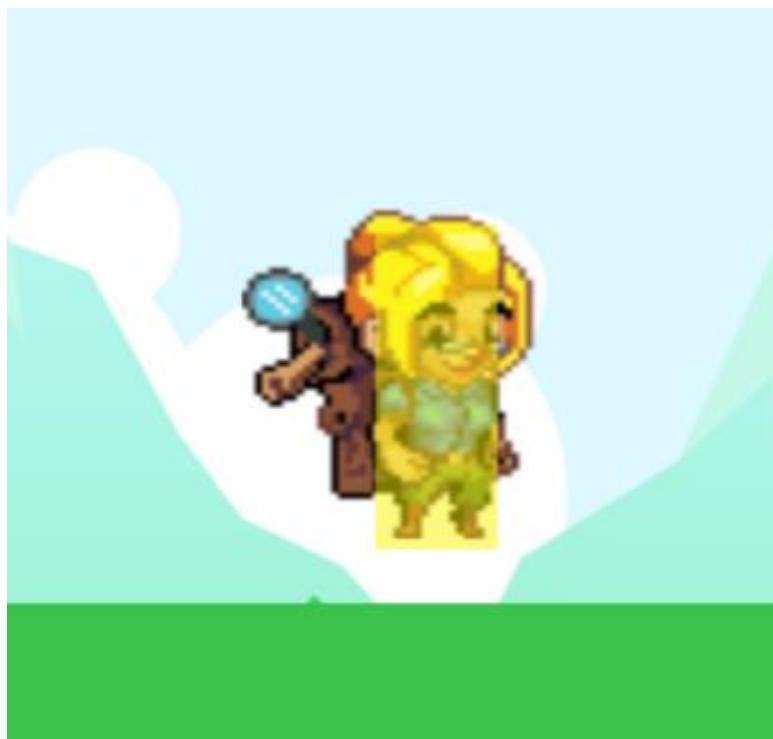


Figura 45: Construct 2 Jugador y box de colisión

5. Añadir a la Box el comportamiento como en la forma 1.

Ruta/comandos	Seleccionar la Box haciendo click sobre ella -> En el menú izquierdo doble click en "Add Behavior" -> Pulsar sobre el ícono de más -> "Apartado Movement" -> Opción Platform.
Variables	<p>Max Speed: Velocidad máxima de movimiento</p> <p>Acceleration: Velocidad a la que el personaje acelera hasta alcanzar su velocidad máxima.</p> <p>Deceleration: : Velocidad con la que el personaje desacelera hasta detenerse.</p> <p>Jump strength: Fuerza con la que el personaje salta, determina la distancia recorrida en el mismo junto con el parámetro gravedad.</p> <p>Gravity: Fuerza con que el personaje es atraído hacia el suelo.</p> <p>Max fall speed: Máxima velocidad que se puede alcanzar durante una caída desde una plataforma o después de salto.</p> <p>Double jump: Permitir o no permitir doble salto.</p> <p>Jump sustain:</p> <p>Default controllers: Controles por defecto del personaje o editarlos y realizar una configuración propia.</p> <p>Initial state: Determinar si el estado en el momento que el objeto se encuentra o aparece sobre la escena se está moviendo o no.</p>

6. Crear el evento para que el Sprite del personaje se mueva a la vez que la Player Box.

Ruta/comandos	Pestaña “Event Sheet” -> Seleccionar “Add Event” de tipo “System” y dentro de estos “Every tick” -> El sistema volverá la pantalla de eventos pulsamos “Add action” sobre el evento creado en el paso anterior-> Doble Clic sobre el Sprite del personaje -> Ahora en el sistema de acción elegir la acción “Set Postion to another object” , véase Figura 33, dentro del apartado “Size and position” -> En el menú que se abrirá a continuación elegir el objeto Player Box.
----------------------	--

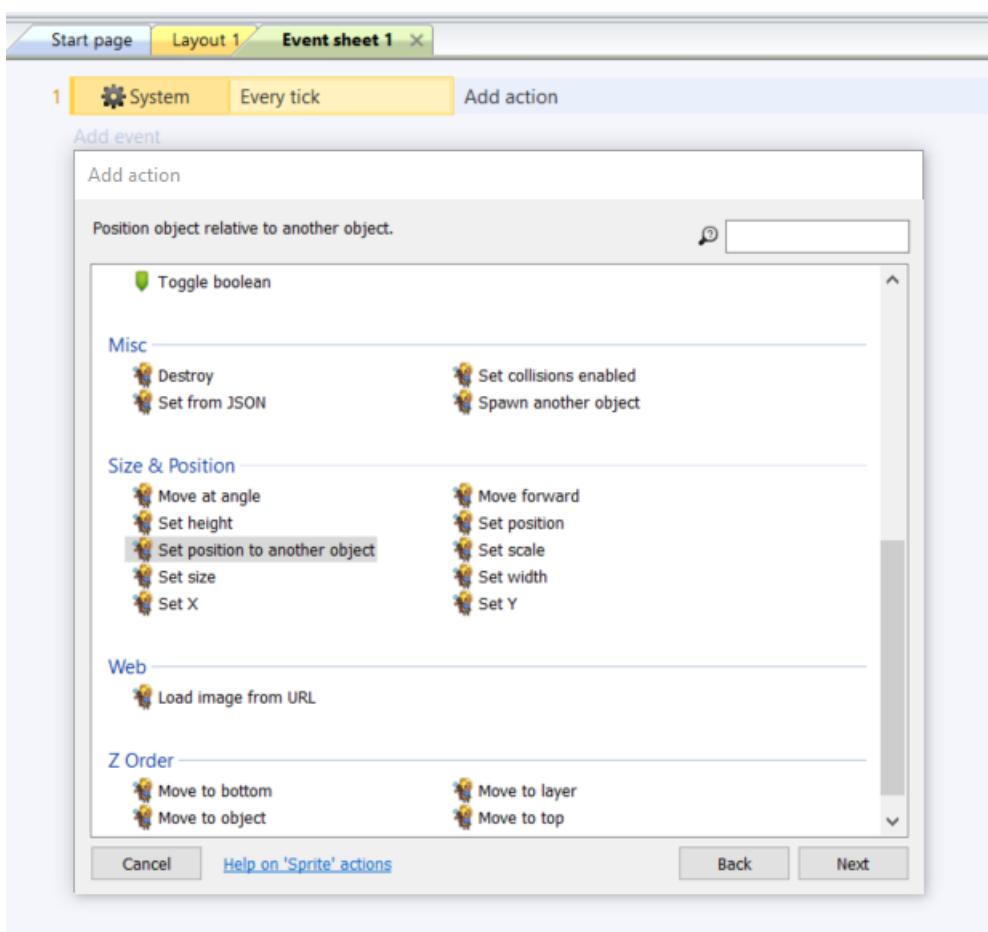


Figura 46: Construct 2 acción "set postion to another object"

7. Como en el modo sencillo de implementar el movimiento en dos dimensiones, insertar y dar el comportamiento tipo sólido a los objetos que se quiera que actúen como plataformas.
8. El resultado del movimiento avanzado se puede observar en la siguiente Figura.

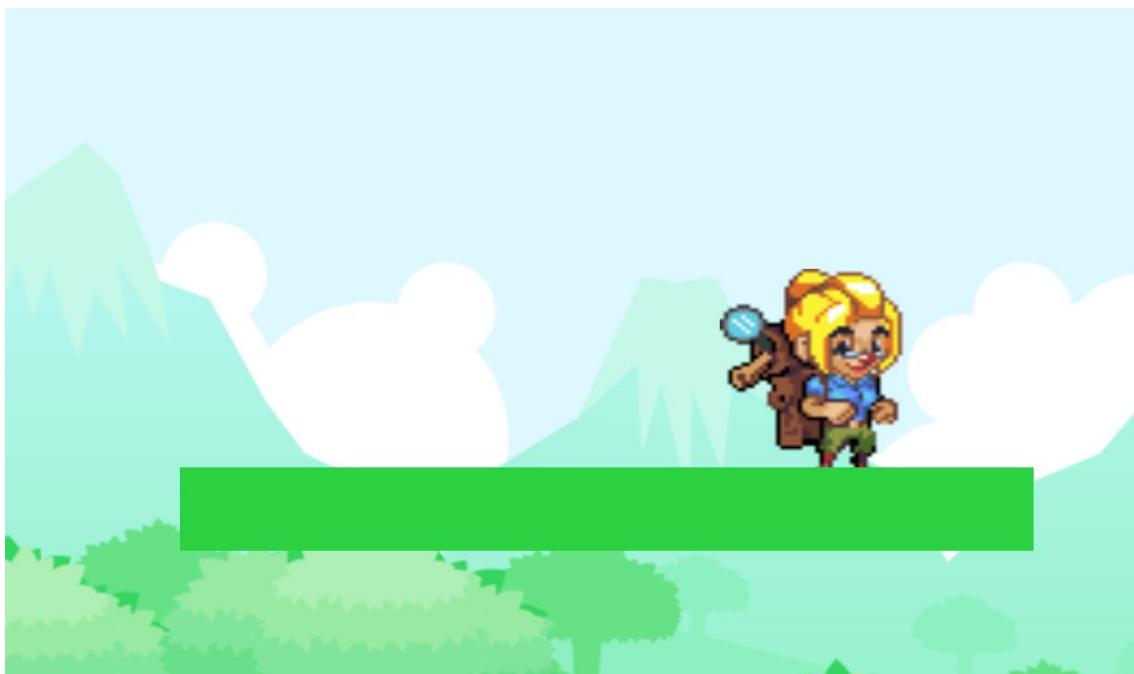


Figura 47: Movimiento 2d modo avanzado.

Tabla resumen funcionalidad.

Funcionalidad	Valoración	Puntuación
Movimiento en dos dimensiones (Forma 1)	Muy fácil	7.5

Salto de personaje que se mueve en dos direcciones.

El salto en dos direcciones se incluye por defecto en el momento que a un objeto se le añade el movimiento tipo plataforma. Con lo que el proceso sería el mismo del paso anterior para su aplicación.

Tabla resumen funcionalidad.

Funcionalidad	Valoración	Puntuación
Salto.	Muy fácil	10

Movimiento personaje en 8 direcciones.

Para que un Sprite se mueva en 8 dimensiones solo tenemos que añadir el dicho comportamiento al mismo.

1. Añadir el Sprite que va ser el personaje que se mueva.

2. Insertar el comportamiento movimiento en 8 direcciones al Sprite añadido en el paso anterior.

Ruta/comandos	Clic sobre el Sprite -> En el menu “Properties” situado a la izquierda en la interfaz seleccionar “Add/Edit Behavior” -> Pulsar el simbolo más de añadir comportamiento -> Clic en el comportamiento “Movement 8 directions” .
Variables	<p>Max speed: Velocidad máxima de movimiento.</p> <p>Acceleration: Velocidad a la que el personaje acelera hasta alcanzar su velocidad máxima.</p> <p>Deceleration: Velocidad con la que el personaje desacelera hasta detenerse.</p> <p>Directions: Las direcciones en que el Sprite puede moverse. Por defecto izquierda, derecha, arriba, abajo y las cuatro diagonales.</p> <p>Set angle: Ángulo de giro de personaje. Valores posibles: No, 90º, 180º, 360º</p> <p>Default controllers: Controles por defecto del personaje o editarlos y realizar una configuración propia.</p> <p>Initial State: Determinar si el estado en el momento que el objeto se encuentra o aparece sobre la escena se está moviendo o no.</p>

Tabla resumen funcionalidad.

Funcionalidad	Valoración	Puntuación
Movimiento 8 direcciones	Muy fácil	10

Disparo de bala.

La implementación del disparo de bala en Construct 2 se implementa de la siguiente forma:

1. Insertar el Sprite de la bala y del personaje que disparará.
2. Insertar un objeto de tipo Mouse (podría ser también Keyboard) para detectar la pulsación de la tecla que va desencadenar el disparo.

Ruta/comandos	Doble clic sobre el layout para abrir el menú de insertar nuevo objeto -> Dentro de los objetos tipo "Input" Clic en Mouse.
---------------	---

3. Insertar en el Sprite bala el comportamiento tipo "Bullet" , véase Figura 33 .

Ruta/comandos	Clic sobre el Sprite -> En el menu "Properties" situado a la izquierda en la interfaz seleccionar "Add/Edit Behavior" -> Pulsar el simbolo más de añadir comportamiento -> Clic en el comportamiento "Bullet".
Variables	<p>Max speed: Velocidad máxima de movimiento.</p> <p>Acceleration: Velocidad a la que el personaje acelera hasta alcanzar su velocidad máxima.</p> <p>Gravity: Fuerza con la que la bala es atraída hacia el suelo y por lo tanto deja de moverse.</p>

	<p>Bounce of solids: Determina si la bala rebota contra los objetos sólidos de la escena o no.</p> <p>Set angle: Si/No. Define si la bala se mueve en línea recta o con ángulo.</p> <p>Initial State: Determinar si el estado en el momento que el objeto se encuentra o aparece sobre la escena se está moviendo o no.</p>
--	--

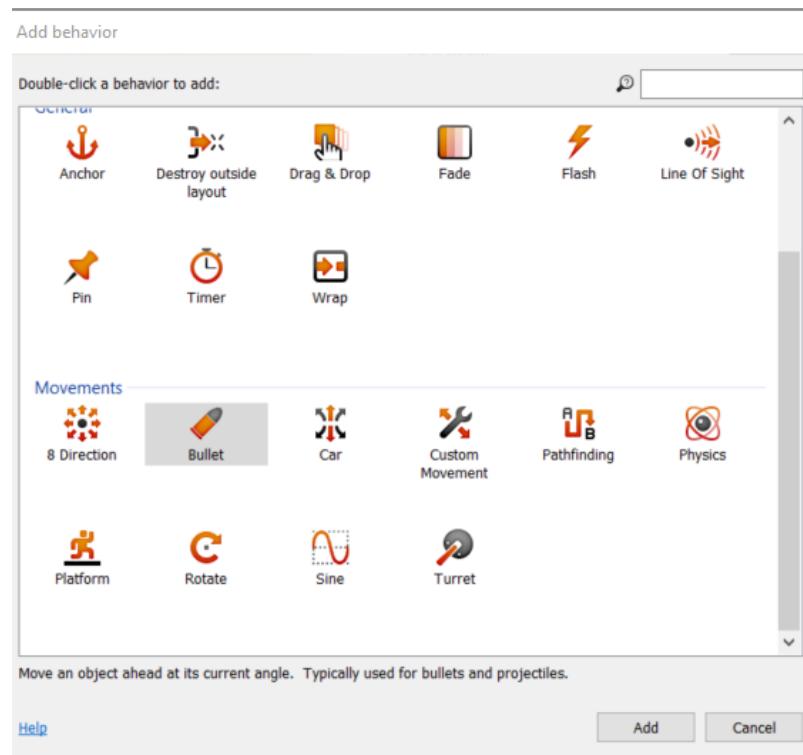


Figura 48: Construct 2: Comportamiento tipo Bullet

4. Ahora hay que crear el evento que va desencadenar el disparo mediante la pulsación de una tecla del mouse, véase Figura 36.

Ruta/comandos	Pestaña "Event Sheet" -> Clic en Add Event -> Elegir el objeto mouse en la pestaña de selección de objeto del evento -> Dentro de los eventos de Mouse elegir el evento "On click" -> Introducir la tecla que se quiera
----------------------	---

que realice la acción -> De nuevo en "Event Sheet" opción Add Action sobre el evento que acabamos de crear -> Dentro de las acciones clic en Sprite Player-> Clic en el evento "Spawn another object" -> Elegir el objeto bala.

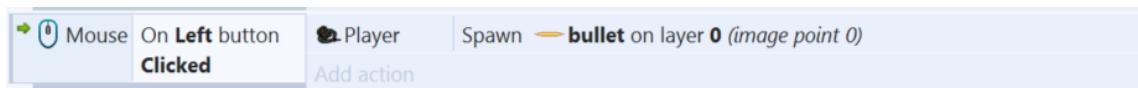


Figura 49: Construct 2 disparo de bala tras pulsación

Hay dos puntos importantes para una mejor visualización y funcionamiento del disparo de una bala.

Por un lado se debe posicionar la bala en el editor de sprites , horizontalmente ,en el caso que la bala no sea circular, ya que de esa forma la bala saldrá del personaje de manera correcta y no se creará en el personaje verticalmente (esto también se puede hacer añadiendo un evento que rote la bala en su creación pero la forma indicada es la más sencilla) , por otra parte también es aconsejable también desde el editor de sprites seleccionar en el Sprite Player editar su punto de origen, véase Figura 37, y posicionarlo en la pistola o desde el lugar que se quiera que se disparen las balas.

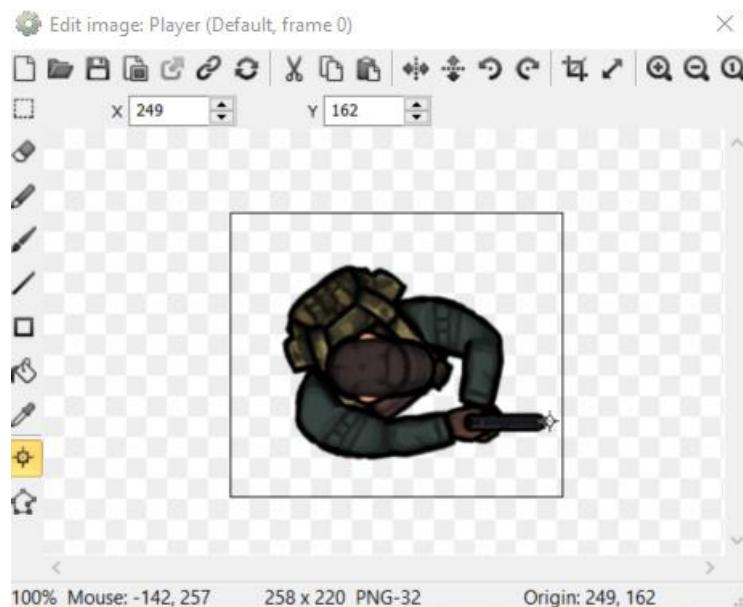


Figura 50: Construct 2 : posicionar origen de objeto para disparo

Tabla resumen funcionalidad.

Funcionalidad	Valoración	Puntuación
Disparo	Fácil	7.5

Colisión con objeto.

Para la colisión entre dos objetos simplemente habrá que añadir el evento de colisión entre dos objetos con los dos objetos presentes en la escena.

Ruta/comandos	Descripción
Colisión	Pestaña "Event Sheet" -> Doble clic sobre "Add Event" -> Seleccionar el objeto 1 de la colisión -> elegir el evento tipo "Colision with another object" -> En el menú siguiente clic sobre le otro objeto de la colisión.

Tabla resumen funcionalidad.

Funcionalidad	Valoración	Puntuación
Colisión	Muy fácil	10

Destrucción de objeto tras colisión.

La implementación de la destrucción del objeto tras colisión es bastante sencilla y rápida en Construct 2. Únicamente habrá que añadir una acción, destrucción de objeto Figura 38, al evento creado en el paso anterior.

Ruta/comandos	Descripción
Colisión	Doble clic en "Add action" -> Elegir el objeto a destruir en el menú siguiente -> Seleccionar la opción "destroy" dentro del apartado "Misc".

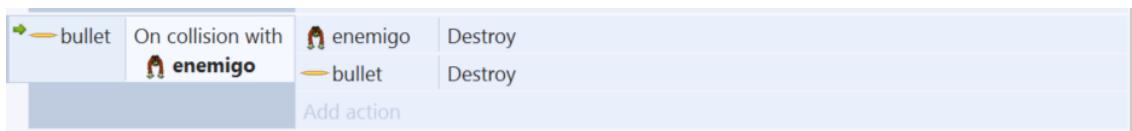


Figura 51:Construct 2: Evento y acción de destrucción de objeto tras colisión

Funcionalidad	Valoración	Puntuación
Destrucción tras colisión	Muy fácil	10

Resumen de funciones básicas y puntuación del subcriterio.

Función	Dificultad	Puntuación
Creación de un proyecto	Muy fácil	10
Insertar fondo a una escena.	Fácil	10
Movimiento del personaje en 2 direcciones	Muy Fácil	7.5
Salto de personaje en 2 direcciones	Muy fácil	10
Movimiento de personaje en 8 direcciones	Muy fácil	10
Disparo de bala.	Fácil	7.5
Colisión entre objetos.	Muy Fácil.	10
Destrucción de Objeto tras colisión.	Muy Fácil.	10
PUNTUACIÓN FINAL		75
FACTOR MULTIPLICADOR DE CORRECIÓN		0.95
PUNTUACIÓN CON FACTOR DE CORRECIÓN		10.68

Tabla 77: Construct 2: puntuación funcionalidades

En la siguiente tabla se resumen la puntuación de Construct 2 en el apartado de funcionalidades.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Funcionalidades	0-12	10.68

Tabla 78: Puntuación subcriterio funcionalidades

Puntuación subcriterio lógica de programación necesaria

En Construct 2 la mayoría de las acciones se nombran como se realiza en la programación. Por ejemplo, los bucles **While**, **Each** o **for**, están dentro de las acciones que el programa permite realizar.

Sucede lo mismo para las comparaciones con los operadores clásicos de programación.

Otro ejemplo es en el acceso a variables de un objeto que se realiza de misma forma que la mayoría de lenguajes de programación , por ejemplo si se quiere reducir la vida de un personaje en 5 puntos se realizaría de la siguiente forma: *Jugador.vida = Jugador.vida-5.*

Como se acaba de comprobar Construct 2 hace el uso en la mayoría de sus partes de la nomenclatura clásica de los lenguajes de programación. Por lo que la puntuación en este apartado debe ser bastante baja.

Pese a ello hay que indicar que la mayoría de las acciones pese a tener dicha nomenclatura están bastante bien explicadas y cuenta con las suficientes ayudas tanto de documentación como durante el manejo del programa para que usuarios sin conocimientos del programa pueden llegar a desenvolverse bien sin tener estos conceptos de programación, aunque habrá partes que no entenderán.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Lógica de programación necesaria	0-4	1

Tabla 79: Construct 2 Subcriterio lógica de programación

Resumen puntuación funcionamiento y dificultad de uso

La tabla siguiente muestra la puntuación de Construct 2 en el criterio funcionamiento y dificultad de uso.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Funcionamiento General	0-4	3.5
Primeros Pasos: Creación de un pequeño proyecto.	0-4	3
Funciones básicas	0-12	10.68
Lógica de programación necesaria	0-4	1
TOTAL		18.18

Tabla 80: Construct 2 criterio funcionamiento y dificultad de uso

Criterio 4: Editor Gráfico/Visual.

A continuación, se exponen los datos obtenidos por Construct 2 en la encuesta de valoración de la interfaz gráfica.

Concepto	Puntuación Máxima	Puntuación minima	Media (Sobre 5)
Rapidez al inciarse	5	4	4,875
El programa se muestra rápido	5	5	4,625
La interfaz es atractiva	5	2	4
Tiene un aspecto cuidado y moderno	5	2	3,375
Puntuación diseño del programa	4	2	2,875
Los iconos representan bien los elementos Y/o funciones	4	3	3,5
A priori parece sencillo de usar	5	2	3,75

Consumo de recursos	5	3	3,625
MEDIA TOTAL			3,85/5
PUNTUACIÓN PONDERADA			6.15/8

Tabla 81: Resultado encuestas sobre interfaz Construct 2

En la siguiente tabla se muestra la puntuación de la encuesta ponderada al tipo de importancia del criterio editor visual y gráfico.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Editor Gráfico/Visual	Clave	6.15

Tabla 82: Criterio: Editor gráfico/visual

Criterio 5: Escalabilidad.

Puntuación subcriterio posibilidad de creación de funciones, scripts y/o lenguaje de programación propio.

Construct 2 cuenta ya de por si con una gran cantidad de funciones que hace que la mayoría de usuarios no tengan que crear funcionales o scripts propios. Pero si se desean crear se pueden realizar mediante la construcción de plug-ings o la modificación de los ya existentes.

Los plug-ings se pueden crear con el lenguaje de programación Javascript, veáse Figura 55, por lo que no es necesario aprender ningún lenguaje propio , aunque si será necesario conocer las funciones propias de Construct 2 para integrarlas con las existentes.

Esta posibilidad de crear funciones se complementa con la posibilidad del uso de JSON para comunicarse con el programa, por ejemplo, para la creación de mapas dinámicos.

```
// Scrolling one parent page frame we need to block certain keys from being processed
var keysToBlockWhenFramed = [32, 33, 34, 35, 36, 37, 38, 39, 40, 44];

instanceProto.onKeyDown = function (info)
{
    var alreadyPreventedDefault = false;

    // Always block certain key presses in frames which can result in page scrolling.
    if (window != window.top && keysToBlockWhenFramed.indexOf(info.which) > -1)
    {
        info.preventDefault();
        alreadyPreventedDefault = true;
        info.stopPropagation();
    }

    // Key already down: ignore, must be a repeat
    if (this.keyMap[info.which])
    {
        if (this.usedKeys[info.which] && !alreadyPreventedDefault)
            info.preventDefault();

        return;
    }
}
```

Tabla 83: Código Javascript Plug-ing Keyboard

Además, se tiene acceso al código de todos los plug-ings que viene por defecto con Construct 2 por lo que se pueden modificar para adaptarlas a las necesidades de nuestro proyecto o crear nuevas a partir de las mismas al disponerlas de ejemplo.

El uso de Javascript para expandir las funcionalidades del programa junto al de JSON para comunicación y automatización de la creación de proyectos o de partes de los mismos, hace que la puntuación en este apartado sea muy alta. La puntuación se reduce en un punto dado que en el lenguaje Javascript es bastante complicado trabajar de forma orientada a objetos y es precisamente la forma que requiere Construct 2. Aunque se puede subsanar mediante el uso a modo de ejemplo de los plug-ings oficiales o con el uso de herramientas como TypeScript¹⁶.

¹⁶ Lenguaje de programación libre y de código abierto desarrollado y mantenido por Microsoft. Es un superconjunto de JavaScript, que esencialmente añade tipado estático y objetos basados en clases.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Posibilidad de crear funciones y scripts propios y/o lenguaje de programación propio.	0-8	7

Tabla 84: Construct 2: Puntuación subcriterio creación de funciones propias.

Puntuación subcriterio Add-ons y plug-ins

En la página oficial de Scirra hay disponible una tienda en la que se pueden comprar diferentes recursos gráficos, plug-ins y complementos que aumentan las funcionalidades de Construct 2 o aumentan las posibilidades de las ya existentes. Los tipos de recursos que se pueden comprar se clasifican de la siguiente forma:

- **Addons:** Dentro de los addons se encuentran dos tipos de elementos. Los Plugin-ins que aumentan las funcionalidades del programa y los comportamientos que se pueden aplicar los objetos en el juego. Entre los primeros destacan los dedicados a la integración con otros programas o librerías externas como Facebook, Steam o Sweet Alert y los que se centran en el uso de funciones propias de los dispositivos móviles como el acceso a las notificaciones y ubicación del terminal.
- **Gamedev Tools:** En este apartado se clasifican aquellas herramientas que no son propias del programa, sino que son accesorias al mismo y que ayuda al desarrollo de juegos. Por ejemplo, un servidor dedicado para los juegos multiplayer creados con Construct un programa para la edición de Sprites o un manejador de archivos JSON.
- **Royalty Free Assets:** Paquetes de sprites, animaciones , fondos, iconos, efectos y otro elementos gráficos para usar en el desarrollo de los juegos. Estos paquetes de Sprite una vez realizada la compra se pueden usar los proyectos sin limitación ninguna ni de modificación ni de uso comercial. También en este apartado se encuentran las plantillas, proyectos de Construct 2 completamente accesibles de juegos finalizados o de funcionalidades concretas, a partir de las que construir el juego del genero de la plantilla.

- **Ebooks:** Libros sobre el desarrollo de videojuegos, el proceso creativo y las creaciones de proyectos en Construct 2.
- **Games:** Juegos completos desarrollados con Construct 2. Dentro de esta sección se encuentran juegos con el código y el proyecto incluido. Para la contabilización de los Plug-ins se contarán únicamente los juegos con código y proyecto ya que los sin estos elementos no se pueden considerar añadidos ni plug-ins.

En la tabla siguiente se muestran los tipos de plugins, la cantidad de cada uno de ellos disponible en la tienda y su precio máximo y mínimo.

Tipo de extensión	Cantidad total	Precio mínimo	Precio máximo
Addons	34	1.19€	69.99€
Gamedev Tools	12	1.19€	53.99€
Royalty Free Assets	1981	1.19€	599.99€
Ebooks	7	13.99€	14,99€
Games.	354	0.59€	77.99€

Tabla 85: Construct 2 addons store

Además de estos plug-ins oficiales en el foro de Construct 2 existe un apartado en el que los usuarios comparten los creados por ellos.

La puntuación en este apartado para Construct 2 es la mitad de la máxima puntuación. Como se ha comprobado cuenta con una gran cantidad de Plug-ins oficiales relativos a los Sprites y elementos gráficos. Pero en lo relativo a complementos que aumentan las funcionalidades del programa se queda bastante corto. A esto hay que añadir la falta de plug-ins gratuitos con los que si que cuentan el resto de programas analizados en el presente trabajo.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Add-ons y plug-ins	0-6	3

Tabla 86: Puntuación subcriterio addons y plugins

Resumen puntuación escalabilidad.

La tabla siguiente muestra el resumen de la puntuación del criterio escalabilidad.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Posibilidad de crear funciones y scripts propios y/o lenguaje de programación propio.	0-8	7
Add-ons y plug-ins	0-6	3
TOTAL		10

Tabla 87: Construct 2 puntuación escalabilidad

Criterio 6: Resultados profesionales.

Al contrario que sucedido con otras herramientas Construct 2 la lista que nos proporciona su web es bastante escasa en cuanto a juegos desarrollados con su herramienta y los juegos que se encuentran en ella son juegos de navegador, minijuegos o juegos que no han tenido relevancia en el mercado.

De entre los títulos desarrollados con Construct solo se ha encontrado un título disponible en Steam, otro en WIIU y XboxOne y otra en GooglePlay el resto solo están disponibles desde la web de los desarrolladores. Aunque estos juegos no hayan tenido el respaldo de ventas, crítica y público, merecen ser tenidos en consideración.

Airscape: The Fall of gravity.

Airscape es un juego de acción, puzzle y plataformas que cuenta como principal atractivo su uso de las físicas, principalmente de la gravedad, y el diseño de niveles como sus puntos fuertes.

El juego sorprende por su colorido mundo, véase Figura 49, sus efectos sonoros y por el hecho de que fue creado por un grupo de jóvenes estudiantes de diferentes partes del mundo trabajando colaborativamente. Por ejemplo, todo el diseño de arte esta creado en Alemania por una estudiante de 13 años y la programación por un estudiante de Australia.

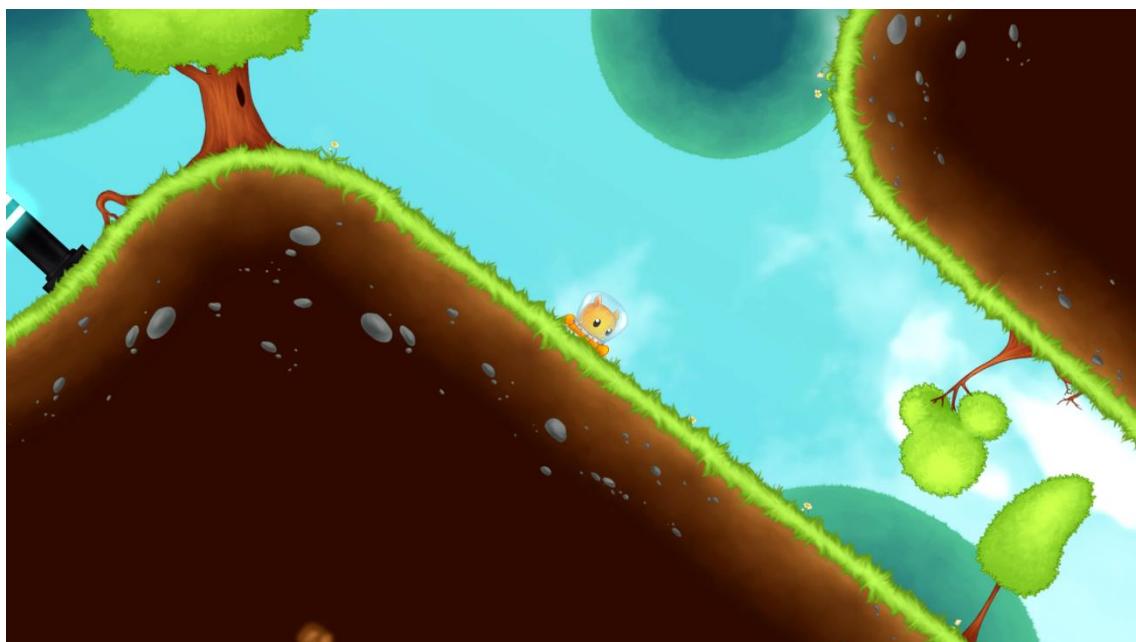


Figura 52: Airscape

Genero	Acción, Plataformas Puzzle.

Lanzamiento	Agosto 2015
Nota Metacritic	75/100
Plataformas disponibles	Pc, Mac.

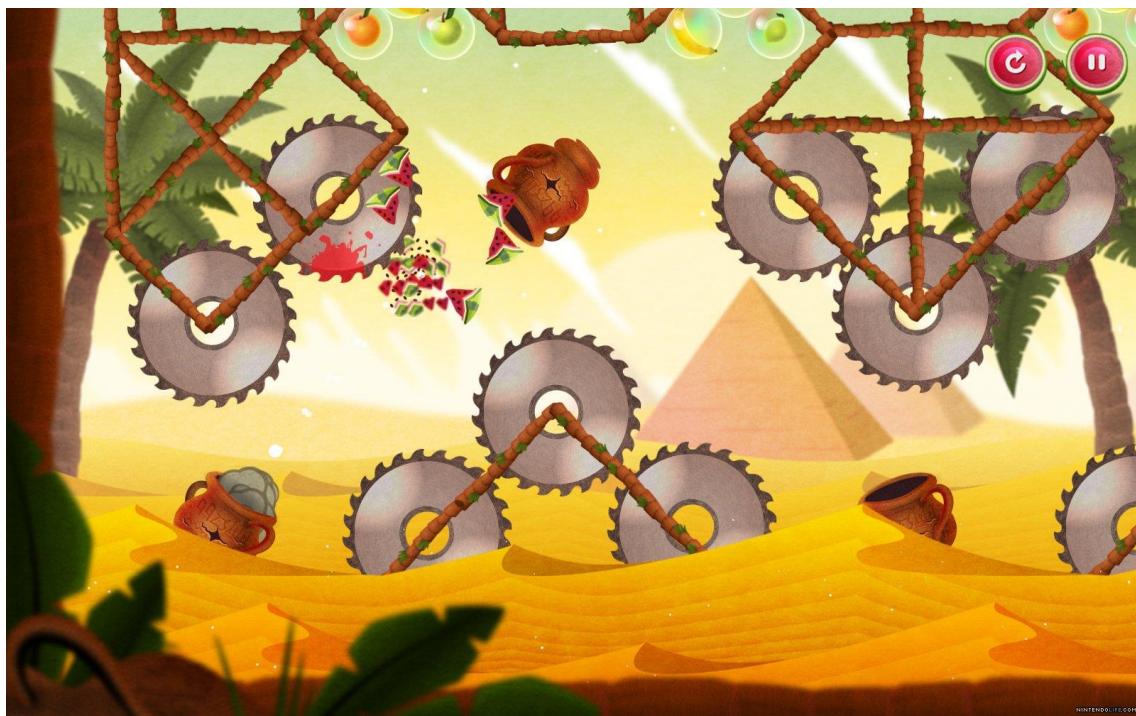
Tabla 88: Airscape datos

Mortal Melon.

Al igual que sucedía con Airscape Mortal Melon también cuenta como piedra fundamental de su jugabilidad las físicas, pero se trata de un juego totalmente diferente.

Mortal Melon es juego de puzzles en que debemos llevar un melón desde un cañón a otro esquivando sierras mecánicas, hielo, agujeros negros y otros elementos que intentan romperlos, véase Figura 40.

El juego apareció en WiiU en formato descargable y de igual forma en XboxOne, pero no tuvo apenas repercusión, esto se constata en que en el principal indicador de notas a nivel de prensa internacional, Metacritic, no se encuentra puntuado ya que son pocos medios los que han analizado el título.

*Figura 53: Mortal Melon.*

Genero	Puzzle.
Lanzamiento	Agosto 2015
Nota Metacritic	Sin nota
Plataformas disponibles	WiiU, Xbox One

Figura 54: Mortal Melon datos

Puntuación subcriterio principales juegos desarrollados

Por lo que se ha comprobado son pocos juegos comerciales los que hacen uso de Construct 2 para su concepción contando únicamente con pequeñas excepciones que se han comercializado pero ninguno de ellos ha logrado ganar popularidad entre la prensa especializada o los jugadores.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Principales juegos desarrollados	0-8	1

Puntuación subcriterio diferenciación

Dado la falta de juegos comerciales, es complicado analizar la diferenciación entre los juegos desarrollados con Construct 2. Si observamos el apartado en la web de Scirra que contiene juegos desarrollados por usuarios de Construct, los juegos son principalmente plataformas en dos dimensiones con un apartado jugable muy parecido entre ellos, aunque existen juegos de acción de vista cenital e incluso algún ARPG, véase Figura 61, son la excepción.



Figura 55: Eurgava

Por lo anteriormente comentado la puntuación, reflejada en la siguiente tabla, este apartado es mínima. La diferenciación entre los juegos desarrollados con Construct 2 es bastante escasa y la mayoría se limitan a un género y aspecto determinado, plataformas 2D.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Diferenciación	0-4	1

Tabla 89: Construct 2 subcriterio diferenciación

Resumen puntuación resultados profesionales.

En la siguiente tabla se muestra el resumen de la puntuación obtenida por Construct 2 en el criterio que analiza los resultados profesionales.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Principales juegos desarrollados.	0-8	1
Diferenciación	0-4	1
TOTAL		2

Tabla 90: Puntuación criterio resultados profesionales Construct 2

Criterio 7: Estado actual.

Construct 2 lleva desde el 2011 en el mercado, es una nueva versión de Construct, que fue lanzado en 2006 el que cuál actualmente ésta retirado.

El estado de Construct 2 es actualmente activo, pese a estar anunciada la nueva versión de la que de momento se conocen pocos detalles.

Las actualizaciones que incluyen nuevas funcionalidades se producen cada 4-5 meses y la corrección de errores y bug es constante por lo que la puntuación en este apartado es bastante alta, aunque se reduce un poco, en el apartado de estado actual, por el anuncio de Construct 3 ya que en el momento que este sea lanzado, las actualizaciones de la versión 2 irán poco a poco decreciendo hasta el abandono del programa como sucedió con Construct.

Resumen puntuación criterio Estado actual

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Estado actual	0-2	1.5
Diferenciación	0-1	1
Fecha desde la última actualización.	0-1	1
TOTAL		3.5

Criterio 8: Sistemas operativos soportados y requerimientos.

Todas incluidas las posibilidades de exportación están disponibles en la versión Business del programa, la versión más avanzada a la venta que se detallará en el siguiente punto del análisis dedicado a las licencias.

Resumen puntuación sistemas operativos soportados y posibilidades de exportación.

Las posibilidades de exportación de los juegos de Construct 2 son muy altas cubriendo casi todos los dispositivos y sistemas en el mercado. Aunque salvo WiiU no es compatible con consolas, véase tabla

Sistema/Plataforma	Exportación.
Windows	Si
Linux	Si
iOs	Si
Android	Si
HTML 5	Si
Flash	Si
Windows Store (Mobile y Store escritorio)	Si
Sistemas PlayStation (PS3, PS-VITA, PS4)	No
Sistemas Xbox (XBOX360, XBOX ONE)	No
Sistemas Nintendo (3DS, Wii, WiiU)	Si
RESUMEN Y PUNTUACIÓN PONDERADA	
TOTAL SISTEMAS	8 DE 10
PUNTUACIÓN PONDERADA	6.4

Tabla 91: Posibilidades de exportación Construct 2

Hay que indicar que las exportaciones no son mediante la creación de aplicaciones nativas de los diferentes sistemas, sino a través de la introducción de una capa

intermedia entre la versión HTML 5 del juego y el dispositivo usando un Middleware¹⁷. Por este motivo la puntuación será reducida en dos puntos, ya que con el uso de este Middleware para la exportación los resultados no son todo los buenos que serían con otro método de exportación nativo o sin la introducción de una capa tan invasiva.

A parte de los sistemas puntuados también cuenta con la posibilidad de exportación a Facebook Games y el propio sistema de juegos de Scirra.

En la siguiente tabla se encuentra el resumen de la puntuación de Construct 2 en este criterio.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Posibilidades de exportación: Sistemas y plataformas	0-8	4.4
TOTAL		4.4

Tabla 92: Construct 2 puntuación sistemas exportación

¹⁷ Software que asiste a una aplicación para interactuar o comunicarse con otras aplicaciones, o paquetes de programas, redes, hardware y/o sistemas operativos.

Criterio 9: Licencia.

Construct 2 tiene tres tipos de licencia que intenta abarcar a todos los tipos de usuarios posibles del sistema, desde usuarios novatos a desarrolladores que usen el programa para la creación de sus juegos. A continuación, se analizan las tres versiones disponibles del programa.

Construct 2 Free edition.

La versión gratuita de Construct 2, al contrario de la mayoría de software de creación de videojuegos, permite un uso más o menos avanzado del programa y un entorno en el que salvo proyectos de dimensiones muy elevadas o con especiales características como podría ser un juego multijugador online.

Hay que indicar que si se desea comercializar el juego habrá que adquirir alguna de las versiones superiores ya que la licencia gratuita no permite usar los juegos creados con fines lucrativos o corporativos.

Las principales funcionalidades limitadas en la versión gratuita son las siguientes:

- Modo debug en paralelo a la ejecución del juego en tiempo real.
- Exportación a sistemas Ios, Android, Windows ejecutable, Linu y, WiiU. Por el contrario, si permite la exportación a Windows Store, HTML y publicar el juego en Scirra.
- No permite la inclusión de compras in game.
- Capas de eventos limitadas a 4 y 100 eventos totales.
- Creación de subcarpetas en los proyectos.
- Creación de familias de objetos.

Las limitaciones respecto a la exportación del proyecto para sistemas no incorporados en la versión gratuita se pueden suplir con el uso de un Middleware como CrossWalk, véase Figura 44. Además, Scirra entre sus manuales oficiales cuenta con algunos dedicados a aprender a realizar dicha tarea. Existen también una gran cantidad de tutoriales y guías en el foro oficial.

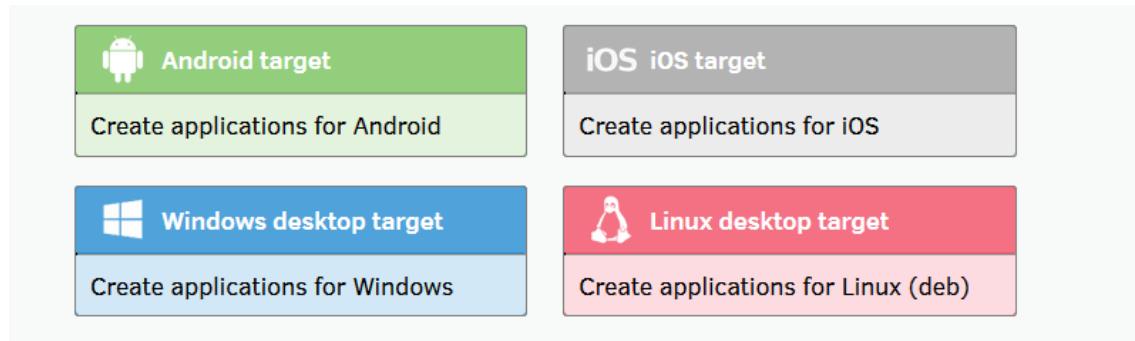


Figura 56: Posibilidades de exportación CrossWalk

Construct 2: Personal licence.

Con la versión Personal Licence de Construct 2 se eliminan todas las restricciones de la versión gratuita. Se tienen disponibles todas las funcionalidades del programa.

La única limitación de este tipo de licencia al viene referido al uso comercial de Construct 2 ya que solo se permiten los siguientes casos:

- -Solo es aplicable a personas físicas, las personas jurídicas o empresas no pueden usar esta versión (como tampoco la gratuita) para fines lucrativos o para la promoción de su marca.
- -Cuando una persona jurídica supere los 5000€ de beneficios deberá hacer uso de la versión Bussines.

El precio de la personal licence es de 75,99€.

Construct 2: Bussines.

La versión Bussines cuenta con las mismas características de funcionamiento que la versión Personal, pero elimina las restricciones relativas a los beneficios obtenidos con el uso del programa y a la posibilidad de que una empresa use Construct 2 para uso lucrativo o corporativo.

Esta licencia no puede adquirirse por separado se tiene que contar con una licencia personal y realizar la actualización a la versión Bussines o adquirir un pack con las dos licencias. El precio de ambas licencias en conjunto es de 405.99€

Puntuación de subcriterio precio de licencias

El precio de la licencia más avanzada de Construct 2 es de 405,99€ por lo tanto la diferencia con el precio medio de las licencias (179.25€) de los programas analizados es mayor de 75€, exactamente 226€, por lo que la puntuación conforme al baremo definido es cero.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Precio de licencias	0-4	0

Tabla 93: Construct 2 subcriterio precio de licencias

Puntuación de subcriterio modo gratuito.

El modo free, visto en la primera parte de este apartado, permite la mayoría de funciones que se necesitan para trabajar con el programa y para realizar juegos sobre todo en etapas iniciales del conocimiento de la herramienta.

El número de eventos al que se está limitado es bastante alto por lo que salvo que se trate de un proyecto de grandes dimensiones o un multiplayer que no está soportado

El número de eventos, objetos y funcionalidades

Las limitaciones de la versión gratuita en cuanto al número de eventos, objetos y funcionalidades que se pueden usar es lo suficientemente alto para que la versión gratuita sea totalmente funcional. Salvo proyecto con unas grandes dimensiones la versión gratuita esta limitación rara vez se alcanzará.

Por lo anteriormente explicado la puntuación en este apartado es alta, ya que aunque nos encontramos con limitaciones, sobre todo en lo relativo a la exportación y uso comercial del programa, el modo gratuito es completamente funcional y adecuado para todo tipo de usuarios.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Modo de prueba o modo free	0-2	1.5

Tabla 94: Construct 2: subcriterio modo gratuito

Puntuación de subcriterio precio de modulos de exportación.

Construct 2 no tiene la venta de módulos para la exportación como un modelo de negocio, por lo que todas las características de exportaciones están incluidas en las licencias Personal y Business. Por lo que la puntuación, reflejada en la siguiente tabla, este apartado es la máxima posible.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Precio de módulos de exportación.	0-2	2

Tabla 95: Construct subcriterio precio módulos de exportación

Resumen puntuación licencias.

La siguiente tabla refleja la puntuación del criterio licencia.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Precio de Licencias	0-4	0
Modo de prueba o free	0-2	1.5
Precio módulos de exportación	0-2	2
TOTAL		3.5

Tabla 96: Construct 2: puntuación criterio licencias

Valoración final y resumen de puntuación

En la tabla siguiente se resumen la puntuación en los 9 criterios evaluables y la puntuación obtenida en total por Construct 2 que ha sido de 66.8 sobre 100.

En el apartado de análisis de resultado se analizarán los resultados y se dará una valoración sobre la misma además de compararla con el resto de programas analizados.

Criterio	Puntuación máxima	Puntuación obtenida	Puntuación porcentual
Requisitos e instalación	3,16	52,67%	3,16
Documentación, tutoriales y comunidad.	14,00	87,50%	14,00
Funcionamiento y dificultad de uso	18,75	78,13%	18,75
Editor Gráfico/Visual	6,15	76,88%	6,15
Escalabilidad	10	71,43%	10
Resultados profesionales	3	25,00%	3
Estado actual	3,5	87,50%	3,5
Sistemas operativos soportados y requerimientos.	4,4	55,00%	4,4
Licencias	3,5	43,75%	3,5
TOTAL	100	66,46	

5.2.4. Análisis de Click Team Fusion (Lo tengo corregido en mail)

Criterio 1: Requisitos e instalación.

Los requisitos para poder desarrollar en Gdevelop son bastante asequibles y la mayoría de equipos pueden trabajar con el programa ya que consume pocos recursos del sistema, como se puede observar en la siguiente tabla.

Por el contrario, CTF solo está disponible para sistemas operativos Windows.

	REQUISITOS CTF
Sistema Operativo	Mínimo Windows XP SP3, soportados Windows Vista, 7, 8 y 10
Procesador	200Mhz o mayor
Memoria	256 MB de RAM
Tarjeta Gráfica	Mínimo Tarjeta Gráfica con soporte para Direct3D 9
Otros	Conexión internet necesaria para la instalación, actualización y descarga del software.

Tabla 97: Gdevelop Requisitos de instalación

Puntuación subcriterio requisitos

Como se ha visto en la tabla anterior los requisitos son bastante bajos pero la puntuación se ve reduce en gran medida, como muestra la tabla siguiente, por el hecho solo estar disponible en sistemas operativos Windows.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Requisitos de instalación.	0-4	2.5

Tabla 98: CTF subcriterio requisitos

Para la instalación de CTF el proceso es el siguiente:

1. Descargar el instalador desde la Web oficial del programa, en el apartado de versión Free .

2. Se ejecuta el instalador y se siguen los pasos para completar la instalación. Durante el proceso se pregunta si se desea instalar junto con el programa proyectos ejemplo y librerías gráficas.

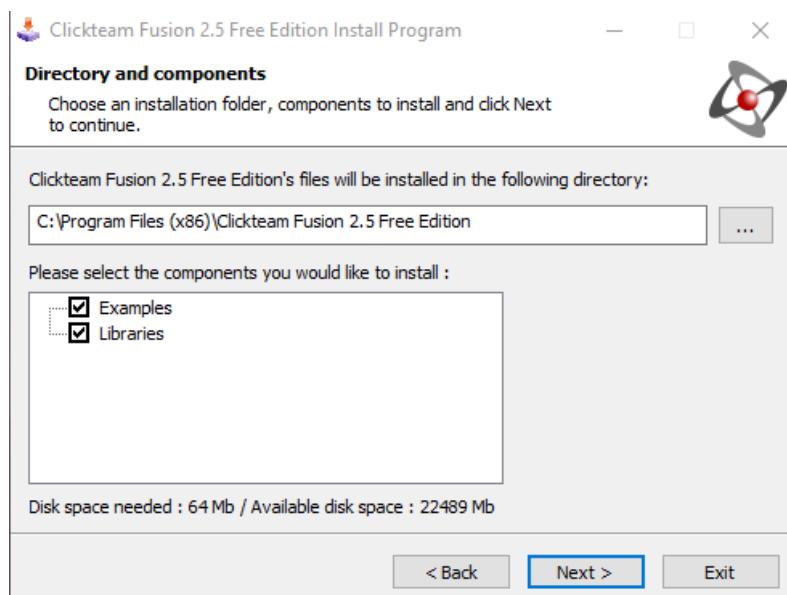


Figura 57: CTF instalación de librerías y ejemplos

3. El programa de instalación ha necesitado un tiempo de 2:25 minutos, más 1:17 minutos de actualización que se lanza la primera vez que se ejecuta el programa.

Resumen proceso de instalación

La tabla siguiente muestra el resultado de la instalación y el resumen de los datos obtenidos.

CTF Instalación en el equipo de pruebas	
Tamaño de descarga	39.8 Mb
Requiere programas adicionales	Si (Para el proceso de compilación y crear los ejecutables del programa es necesaria la instalación de Java Runtime ¹⁸ .)
Instalación de módulos o recursos adicionales durante la instalación	Si
Tiempo total	8 minutos 26 segundos.
Tamaño posterior a la instalación	572 Mb

¹⁸ Conjunto de utilidades que permite la ejecución de programas Java.

Tabla 99: CTF Datos de instalación en el equipo de pruebas

Datos de instalación encuesta.

Los datos recopilados de la encuesta sobre la instalación de CTF son los mostrados en la siguiente tabla.

CTF datos estudio instalación	
Tiempo medio	4 minutos 36 segundos
Valoración media sencillez del proceso de instalación (0 a 10)	9.125
Valoración media de la funcionalidad del proceso de instalación (0 a 10)	9.375
Puntuación media funcionalidad y sencillez	9.25
Fallos en instalación.	No

Tabla 100: Gdevelop datos encuesta instalación

Puntuación subcriterio instalación.

Conforme a la ponderación definida en el marco teórico a CTF le corresponde 1 punto por tener una instalación menos de 5 minutos y 1 punto por tener una valoración media en la funcionalidad y la sencillez del proceso de instalación mayor de 9. Con lo que obtiene la máxima puntuación en este subcriterio.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Instalación.	0-2	2

Tabla 101: CTF Subcriterio instalación

Puntuación criterio requisitos e instalación

En la siguiente tabla se muestra la puntuación del criterio requisitos e instalación de Click Team Fusion.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Requisitos de instalación.	0-4	2.5
Instalación	0-2	2
TOTAL		4.5

Tabla 102: CTF puntuación requisitos e instalación

Criterio 2: Documentación, tutoriales y comunidad.

El punto central de la comunidad que desarrolla o desea iniciarse en CTF es la web **Community Click Team Fusion**¹⁹. Se trata de una web oficial, enlazada con la principal del programa. En ella se da soporte oficial a los desarrolladores del programa con un foro, un blog de los desarrolladores de CTF donde van incluyendo las novedades del programa y un apartado para que los usuarios puedan compartir sus creaciones. Es un portal muy completo y que aporta un gran valor añadido al programa.

El foro tiene un apartado de Tutoriales y Guías muy completo donde los propios usuarios enseñan determinadas funcionalidades mediante ejemplos.

Los tutoriales oficiales disponibles en la página del programa se dividen en:

1. **Game Usage:** Enseñan los fundamentos básicos del programa y como moverse por el mismo. Entre ellos se encuentra tutoriales relativos por ejemplo al uso de la librería gráfica, del editor de imágenes o de la interfaz en general. En este apartado encontramos 5 tutoriales.
2. **Begginner:** Primero juegos que un usuario debería realizar para empezar a dominar el programa, estos juegos no requieren de un conocimiento previo de Click Team fusión y están pensados para usuarios sin ningún conocimiento del mismo. Además de tutoriales de juegos completos , se encuentran otros relativos a funcionalidades básicas como por ejemplo el scrolling, realizar un menú o la explicación de diversos tipos de objetos. 14 tutoriales se encuentran dentro de esta sección.
3. **Intermediate:** Tutoriales en los que ya se empiezan a crear proyectos de dificultad media y para los que se necesita un conocimiento más o menos general del programa y de las funcionalidades básicas. El apartado dedicado a tutoriales de dificultad intermedia contienen 19 tutoriales.
4. **Advanced:** En la parte avanzada de las guías del programa, se explica mediante ejemplos características avanzadas que se puede dar a los juegos que se crean con Click Team Fusión. Entre estas funciones complejas que se explican en este apartado se incluye la gestión de archivos de guardado, la creación de un ranking online o una guía de programación de scripts y nuevos patrones de comportamiento. Se explican catorce funciones avanzadas en este apartado.
5. **Getting the most of:** Aquí se incluyen aquellos tutoriales que no tienen que ver con la creación de juegos en si misma. Son tutoriales que explican características accesorias para los juegos, pero no imprescindibles para su desarrollo. Por ejemplo, en este apartado se incluyen explicaciones como importar gráficos o como adaptar los juegos a resoluciones de gran tamaño. Esta sección la componen 12 tutoriales.
- 6.

¹⁹ <http://community.clickteam.com>

7. **Videos:** Enlaces a páginas con videos de ayuda o de aprendizaje del programa. En estos momento solo existen dos pero ambos enlaces son poco útiles , ya que el primero es un curso de Udemy con un coste de 200€, y el segundo el enlace no funciona, redirige a una web que no está disponible.
8. **Open files:** Colección de juegos, sprites y recursos gráficos de carácter abierto que se pueden utilizar en los proyectos o que se pueden usar como ejemplo por los usuarios. Se encuentran 12 recursos de este tipo en esta sección.

Pese a que el número de tutoriales es bueno, se ha podido comprobar que muchos enlaces a otras páginas o recursos no están disponibles.

Se ha detectado también un error bastante frecuente en los archivos fuente de los proyectos de ejemplo y de los tutoriales y es que pertenecen a la versión anterior del programa, Multimedia Fusion, y no es compatible con Click Team Fusion.

Puntuación subcriterio documentación, foros y tutoriales oficiales.

El número de recursos oficiales que dispone Click Team Fusion es bastante inferior otros frameworks analizados y además tienen los errores anteriormente citados, por lo que en este apartado la puntuación es bastante baja.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Documentación, foros y tutoriales oficiales	0-6	1.5

Tabla 103: CTF subcriterio documentación, foros y tutoriales oficiales.

Tutoriales oficiales realizados.

Catch The Fruit

Este tutorial se encuentra dentro de la categoría de Begginers Tutorials de la web oficial de CTF. En él se enseña a realizar un juego en el que un personaje se mueve de un lado al otro del escenario recogiendo la fruta que cae del cielo en posiciones aleatorias.



Figura 58: CTF tutorial Catch the fruit

El tutorial se proporciona a través de un archivo comprimido, donde se encuentra el tutorial en sí y un archivo de proyecto. El archivo de proyecto no funciona ya que pertenece a la versión antigua del programa Multimedia Fusión. El tutorial no viene con Sprites para realizarlo y se ha tenido que emplear bastante tiempo la búsqueda de los mismo.

El tutorial es bastante incorrecto, con toda una serie de problemas que impiden que el usuario aprenda y se interese por el programa, sino sino más bien todo lo contrario, pues hace que sienta frustración y abandone el aprendizaje de la herramienta. Entre los aspectos que hacen de este tutorial una experiencia negativa destacan los siguientes:

1. Toda la parte de animación está mal explicada y se ha tenido que optar por conseguir el mismo resultado de otras formas posibles investigando la documentación oficial y de otros tutoriales no oficiales.
2. El apartado que explica el movimiento de los objetos que caen esta desactualizado. Por ejemplo, aparece un tipo de movimiento llamado *pinball* que ya no ésta disponible en la versión actual del programa.

3. No se ha conseguido la creación de objetos de forma aleatoria fuera del objetivo de la cámara tal como explica el tutorial ya que no funciona.

Los aspectos positivos del tutorial son los siguientes:

1. El formato PDF permite tener siempre disponible el tutorial para consultas.
2. Se apoya mucho en imágenes que ilustran las acciones a realizar.
3. La idea del juego, además de sencilla, es original y bastante diferente lo que este tipo de tutoriales iniciales suelen ser.

Para que el tutorial fuese funcional y se consiguiera completar un juego o prototipo al finalizarlo, véase Figura 47, se han realizado las siguientes modificaciones:

1. La animación comienza cuando el personaje comienza a moverse.
2. Para conseguir un escenario más grande, se ha centrado la cámara en el personaje para seguirle.
3. Además de la fruta que cae, se han añadido bombas que hacen que el usuario muera al contacto con éstas.
4. Se ha añadido un menú de juego y la puntuación actual en la pantalla.

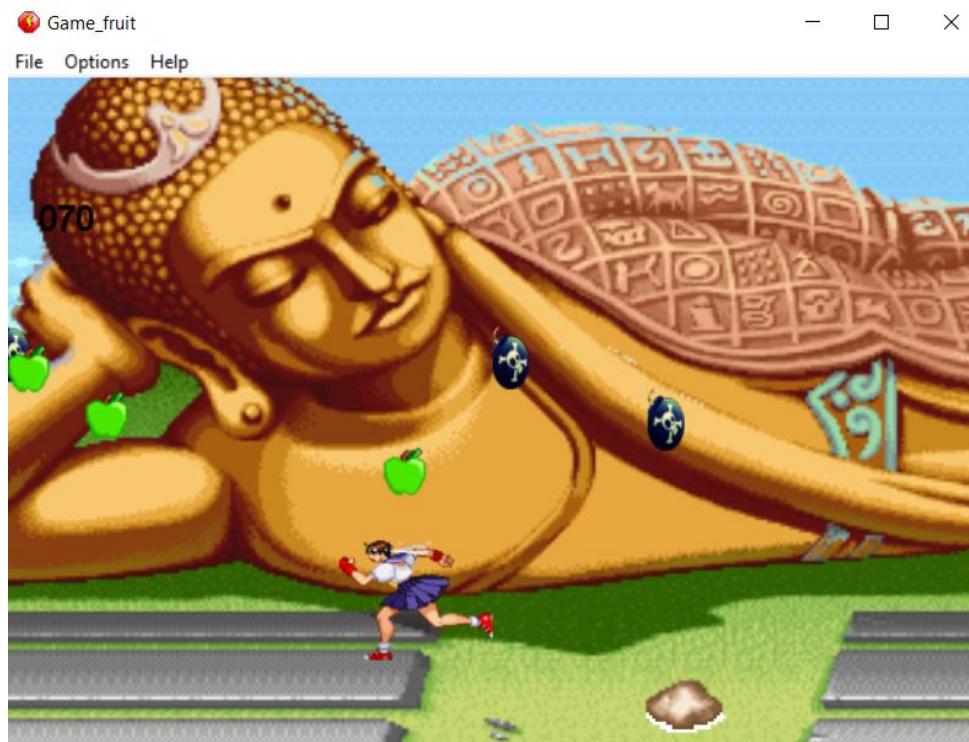


Figura 59: Catch the fruit modificado

Otros tutoriales.

Se ha intentado realizar otros tutoriales oficiales como por ejemplo, *Create an Artillery Game From Scratch* que se encuentra en los tutoriales oficiales la categoría

Begginers. El tutorial es demasiado complejo para que un usuario medio pueda aprender a manejar el programa.

Además muchos de sus apartados no están explicados correctamente, ya que el tutorial se limita a decirle al usuario que copie trozos de código sin explicaciones adjuntas, véase Figura 48.

Then add the following code in the Event Editor (get to this via View > Event Editor):

```
+ Always
- [P1Body] Set DistanceToMouse to:
  Distance("Advanced Direction Object", X("P1Body"), Y("P1Body"), XMouse, YMouse)
- [P1Body] Set AngleToMouse to:
  Direction("Advanced Direction Object", X("P1Body"), Y("P1Body"), XMouse, YMouse)
- [P1Turret] Set position at (0,0) from P1Body
- [P1Turret] Set Angle To AngleToMouse("P1Body")
- [Crosshair] Set X Coordinate to XMouse
- [Crosshair] Set Y Coordinate to YMouse
```

Figura 60: Tutorial Create an Artillery Game From Scratch

Muchas de las funciones que se aplican en el tutorial son demasiado avanzadas para un tutorial de iniciación al programa, junto con que el tutorial no las explica han hecho que el tutorial fuese abandonado durante su realización, ya que no sirve para el propósito del proyecto.

Otro ejemplo de tutorial que se ha intentando realizar es el de **Glob Wars** de la categoría Intermedio, pero tiene los dos problemas de los anteriores. Por un lado al igual que **Catch de fruit** viene con archivos fuente de ejemplo que no funcionan y se limita a decir al usuario lo que tiene que hacer sin explicar lo que realiza, y otro que el documento pdf adjunto en el que viene no cuenta con imágenes es una sucesión de instrucciones de texto.

Como se ha podido comprobar los tutoriales oficiales están desactualizados, son demasiado complejos para usuarios que quieran iniciarse y además vienen con archivos fuente erróneos, por lo que se tiene que recurrir a tutoriales externos no oficiales para aprender el manejo básico del programa , véase Figura 50.

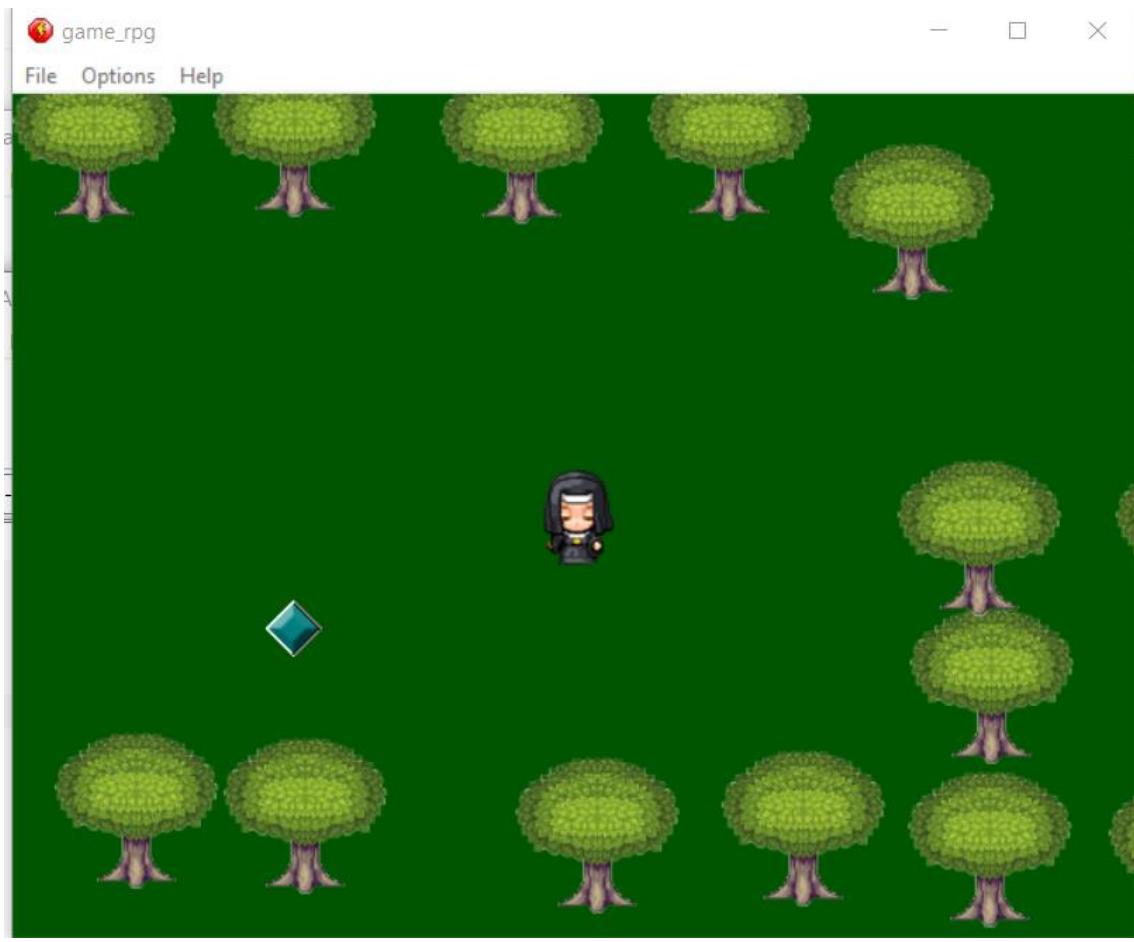


Figura 61: CTF Tutorial externo realizado

Resumen de tutoriales realizados.

La tabla siguiente contiene el resumen de los tutoriales realizados, el tiempo invertido y las funciones aprendidas durante su realización.

Tiempo dedicado.	9 horas 27 minutos
Tutoriales realizados	3
Tutoriales completados	1
Juegos completos	1
Errores detectados en los tutoriales	Considerables
Modificaciones realizadas	Considerables
Funcionalidades aprendidas en los tutoriales	<ul style="list-style-type: none"> • Mover personaje en dos direcciones. • Colisiones. • Insertar fondo a la escena. • Destruir objetos cuando están fuera de la pantalla. • Creación aleatoria en posición de objetos. • Movimiento de caída por la gravedad de objetos. • Mostrar puntuación en pantalla y sumar.

Tabla 104: CTF Resumen tutoriales oficiales realizados

Puntuación subcriterio análisis de tutoriales oficiales.

Como muestra la tabla anterior, los tutoriales oficiales tienen bastantes errores y no son una experiencia de aprendizaje recomendable y resultan bastante frustrantes para un usuario primerizo en el programa, por lo que la puntuación en este apartado es muy baja.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Análisis de tutoriales oficiales	0-8	1

Tabla 105: Subcriterio Gdevelop Análisis de tutoriales oficiales

A continuación se detalla la presencia de CTF en Reedit y Taringa.

Reddit

Reddit

En Reedit el tema de CTF es bastante activo y cuenta con un número bastante grande de hilos. Las interacciones y respuestas suelen ser bastante habituales y los hilos se actualizan con frecuencia. Cuenta con más 500 subscriptores en el momento de realizar este informe.

The screenshot shows a Reddit thread on the 'clickteam' subreddit. The first post is a resource titled 'Welcome New Comers to CT Fusion 2.5! Please make sure to check out the tutorials to get started ASAP!' with 19 upvotes and 15 comments. Below it is a post from 'RockyCoop' about 'Firefly is 3d for Fusion' with 13 upvotes and 13 comments. Other posts include 'Transitions having issues with Direct3D 9' by 'JoshCreatesGames' and 'Help with Timers' by 'dduong16'. The sidebar shows a search bar, user login fields, and a sign-in button. At the bottom right are buttons for 'Enviar un nuevo enlace' and 'Enviar un nuevo post de texto'.

Tabla 106: CTF hilo Reedit

Taringa.

En Taringa, CTF no cuenta con una comunidad de desarrolladores y usuarios.

Puntuación subcriterio comunidades y foros de desarrolladores.

CTF solo tiene una comunidad activa en Reedit por lo que en este apartado le corresponde 1 punto.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Foros y comunidades de desarrolladores	0-2	1

Tabla 107: CTF subcriterio *foros y comunidad de desarrolladores*

Resumen de puntuación Documentación, tutoriales y comunidad.

La tabla siguiente resume la puntuación del este criterio.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Documentación, foros y tutoriales oficiales	0-6	1.5
Análisis de tutoriales oficiales	0-8	1
Foros y comunidades de desarrolladores	0-2	1
TOTAL		3.5

Tabla 108: CTF criterio *documentación, tutoriales y comunidad*

Criterio 3: Funcionamiento y dificultad de uso.

CTF se muestra rápido y no se han detectado problemas de estabilidad o errores durante la ejecución.

Pero tiene una muy mala gestión de Sprites, reescalado, posicionamiento y/o animaciones. Aunque un aspecto positivo respecto a los Sprites es que detecta los números de las animaciones. Si se introduce el mismo nombre y un número al final, al importar las imágenes el programa detecta la serie de imágenes como animación. Si se introduce el mismo nombre y un número al final, al importar las imágenes el programa detecta la serie de imágenes como animación y las ordena.

Otro aspecto poco funcional es que los objetos deben estar en la escena, aunque vayan a aparecer posteriormente en la misma, pero si durante la escena se van a crear o se van a usar dichos objetos deben estar en ella posicionados, aunque sea fuera de la ventana de visualización.

Puntuación subcriterio funcionamiento.

Como se acaba de comprobar CTF tiene algunos defectos de funcionamiento, por lo que la puntuación en este apartado será reducida un 50%.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Funcionamiento General	0-4	2

Tabla 109: CTF subcriterio funcionamiento general

Puntuación primeros pasos: creación de un pequeño proyecto.

Los primeros pasos de Click Team Fusion son algo complejos. No porque muchas de sus funcionalidades sean difíciles de aplicar, sino más bien por el diseño de la interfaz y la dificultad de encontrar los elementos.

Por ejemplo, la división entre eventos y escenas es bastante compleja de manejar en un primer momento, y además no cuenta con un sistema de iconos que representen bien a los elementos, por lo que hasta que no se realizan un número determinado de tutoriales el usuario se encuentra bastante perdido en el programa.

Pero hay un elemento que ayuda a que los primeros pasos el usuario sienta que progresá y que es capaz de empezar a crear funcionalidades jugables: el editor de eventos. La gestión de los eventos es muy sencilla y permite al usuario gestionarlos de una manera rápida y muy visual.

Aún con dificultades citadas se ha conseguido completar durante el tiempo de prueba del programa un prototipo de juego. Por lo que la puntuación en este apartado será del 50%.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Primeros pasos: Creación de un pequeño proyecto.	0-4	2

Tabla 110: Gdevelop subcriterio funcionamiento general

Funciones básicas

A continuación, se va a explicar cómo implementar las funciones básicas descritas en la definición de los criterios de selección y se va a puntuar en función de su dificultad.

Creación de un proyecto.

La creación de un proyecto en Click Team Fusion se realiza de la siguiente forma.

Ruta/comandos	File->New
Variables	Ninguna.

Una vez realizada esta operación el programa automáticamente muestra la vista principal de un proyecto, véase Figura 51, desde donde se puede dar nombre a la aplicación y a la escena, así como determinar la resolución de la misma y otros parámetros.

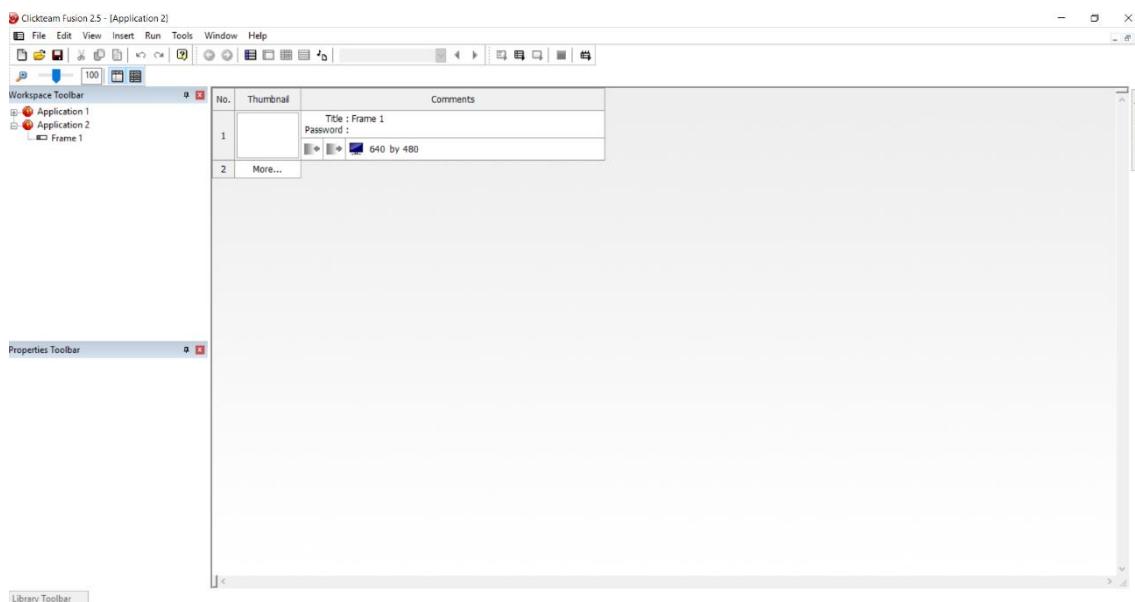


Figura 62: CTF vista de proyecto

La puntuación de este criterio es la siguiente:

Funcionalidad	Valoración	Puntuación
Creación proyecto.	Muy fácil	10

Insertar fondo en la escena.

El proceso para insertar fondo en Click Team Fusion es el siguiente:

1. Habrá que pulsar sobre el Frame de la aplicación donde se desea situar el fondo.

Ruta/comandos	Insert->New Object->Background->Seleccionar "Backdrop"
Variables	Image : Al pulsar sobre "Edit" se abrirá un menú donde se permitirá seleccionar la imagen que se desea como fondo.

2. Con este proceso se incluirá la imagen en la escena. A continuación, véase Figura 51, se puede colocar en el lugar donde se deseé posicionar arrastrando y escalándola.

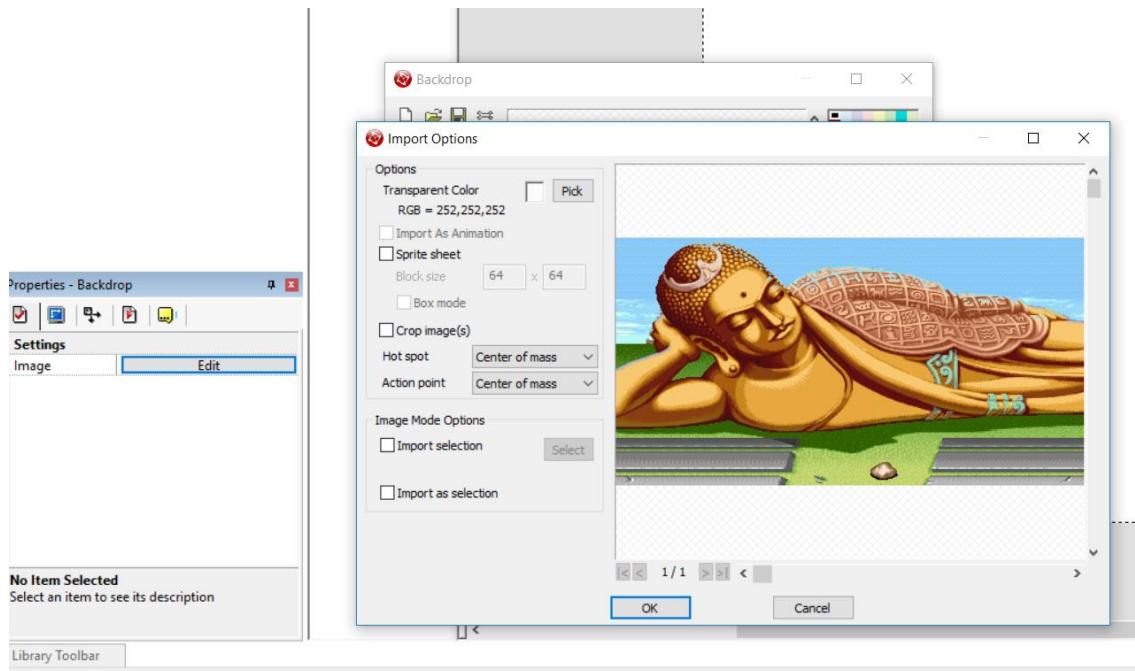


Figura 63: CTF posicionar background

Tabla resumen funcionalidad.

Funcionalidad	Valoración	Puntuación
Insertar fondo	Muy fácil	10

Movimiento de personaje en dos direcciones.

1. Primero se debe incluir el Sprite que va a moverse.

Ruta/comandos	Seleccionar el Frame haciendo doble clic -> Insert de la barra superior -> New Object -> All Object -> Active
----------------------	---

2. Dotar de movimiento al Sprite incluido, véase Figura 54.

Ruta/comandos	Clic sobre el objeto en la escena -> Menú lateral izquierdo inferior Properties -Active" pestaña clic en pestaña "Movement" -> Seleccionar lista desplegable type
----------------------	---

	"Eight Directions" -> Clic en direcciones -> Dejar seleccionado solo izquierda y derecha
Variables	<p>Directions: En este caso será izquierda y derecha.</p> <p>Initial Directions: Dirección a la que se desea que el Sprite esté mirando en el momento de iniciar el Frame.</p> <p>Speed: Velocidad de movimiento.</p> <p>Deceleration: Velocidad con la que el personaje desacelera hasta detenerse.</p> <p>Acceleration: Velocidad a la que el personaje acelera hasta alcanzar su velocidad máxima.</p> <p>Moving at Star: Determina si el personaje se está moviendo al inicio de la escena.</p> <p>Stick to obstacle: Si se selecciona esta opción el personaje no se moverá al colisionar con obstáculos, mientras que si se deja deseleccionada el personaje retrocederá.</p>

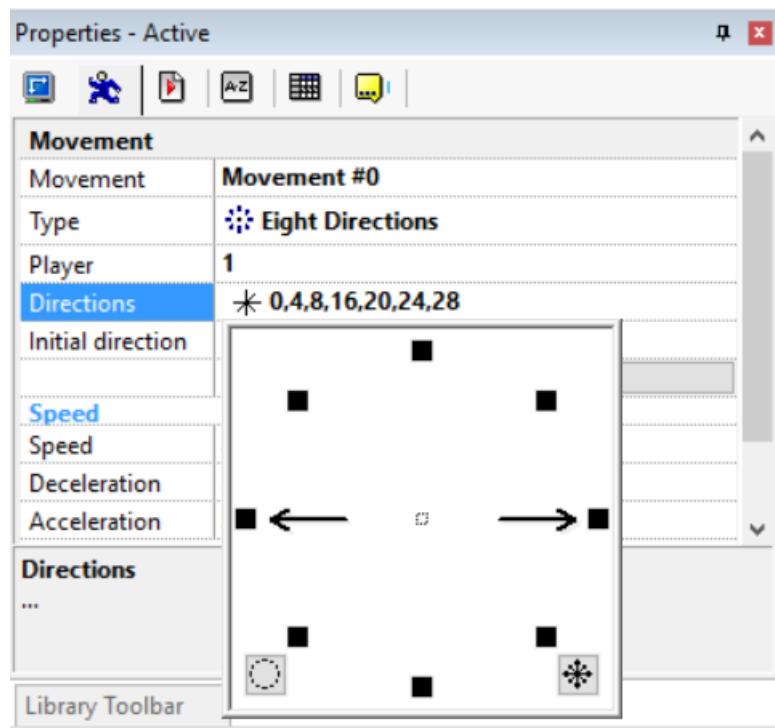


Figura 64: CTF Movimiento 2D

Tabla resumen funcionalidad.

Funcionalidad	Valoración	Puntuación
Movimiento en dos dimensiones	Fácil	7.5

Salto de personaje que se mueve en dos direcciones.

Para poder hacer que el personaje salte y no sólo se mueva en dos direcciones, el proceso es algo más complejo ya que se necesita añadir el motor de física del programa y varios objetos a la escena. El procedimiento es el siguiente.

1. Añadir dos objetos activos a la escena.

Ruta/comandos	Seleccionar el Frame haciendo doble clic -> Insert de la barra superior -> New Object -> All Object -> Active
----------------------	---

2. Uno de ellos actuará de plataforma, por lo que es conveniente ponerle una imagen sólida, por ejemplo, un rectángulo a lo largo de la escena.
3. Ahora hay que aplicar al segundo objeto: el movimiento tipo Plataformas físicas.

Ruta/comandos	Click sobre el objeto -> Pestaña propiedades -> Movement -> Type -> Seleccionar " Physics ->"Platform movement"
Variables	<p>Initial Directions: Dirección a la que se desea que el Sprite esté mirando en el momento de iniciar el Frame.</p> <p>Speed: Velocidad de movimiento.</p> <p>Deceleration: Velocidad con la que el personaje desacelera hasta detenerse.</p> <p>Aceleration: Velocidad a la que el personaje acelera hasta alcanzar su velocidad máxima.</p> <p>Use acceleration when changing direction: Decidir si al cambiar de dirección se usa la aceleración o no.</p> <p>Climbing speed: Velocidad de subida por inclinaciones.</p> <p>Allow crouch: Permite que el personaje se agache o no.</p> <p>Speed when crouched: Velocidad con la que el personaje se agacha.</p> <p>Jump:</p> <ul style="list-style-type: none"> • Strength: Altura del salto. • Multiple jumps: Veces que el personaje puede saltar seguidas. • Next Strength: Altura del segundo salto. • Jump while crouch: Permitir saltar cuando se está agachado. • Apply deceleration to jumping when input stops: Aplicar o no frenado al pulsar el movimiento del personaje. • Horizontal control when jumping: Permite mover al personaje horizontalmente cuando se está en el aire o saltando. • Control: Tecla que activa el salto. <p>Object:</p> <ul style="list-style-type: none"> • Densidad • Elasticity

- **Gravity Scale**

4. Cuando se aplica este movimiento a un objeto el programa nos avisa que se necesita incluir el motor de físicas en la escena para que funcione. Para ello:

Ruta/comandos	Seleccionar el Frame haciendo doble clic -> Insert de la barra superior -> New Object -> Physics -> Seleccionar Physics Engine.
----------------------	---

5. Ahora el objeto 2, se comporta como un objeto que se mueve sobre plataformas con un sistema de físicas. El siguiente paso es realizar un evento para que el objeto 2 deje de caer al contactar con el 1 y actúe como una plataforma, véase Figura 53. Para ello primero se crea la condición de la colisión:

Ruta/comandos	Event Editor-> New Condition->Seleccionar Objeto 2->Collisions ->Another Object->Seleccionar Objeto 1
----------------------	---

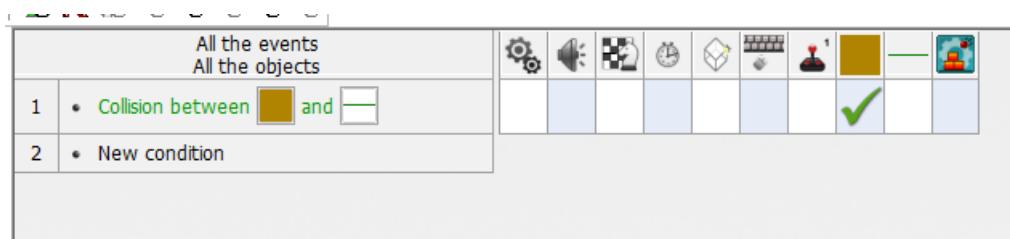


Figura 65: CTF Colision

6. Después, en el menú visual, se selecciona el objeto 2 dentro de la condición.

Ruta/comandos	Doble click en el cuadro del objeto 2->New action->Seleccionar Objeto 2->Movement->Stop.
----------------------	--

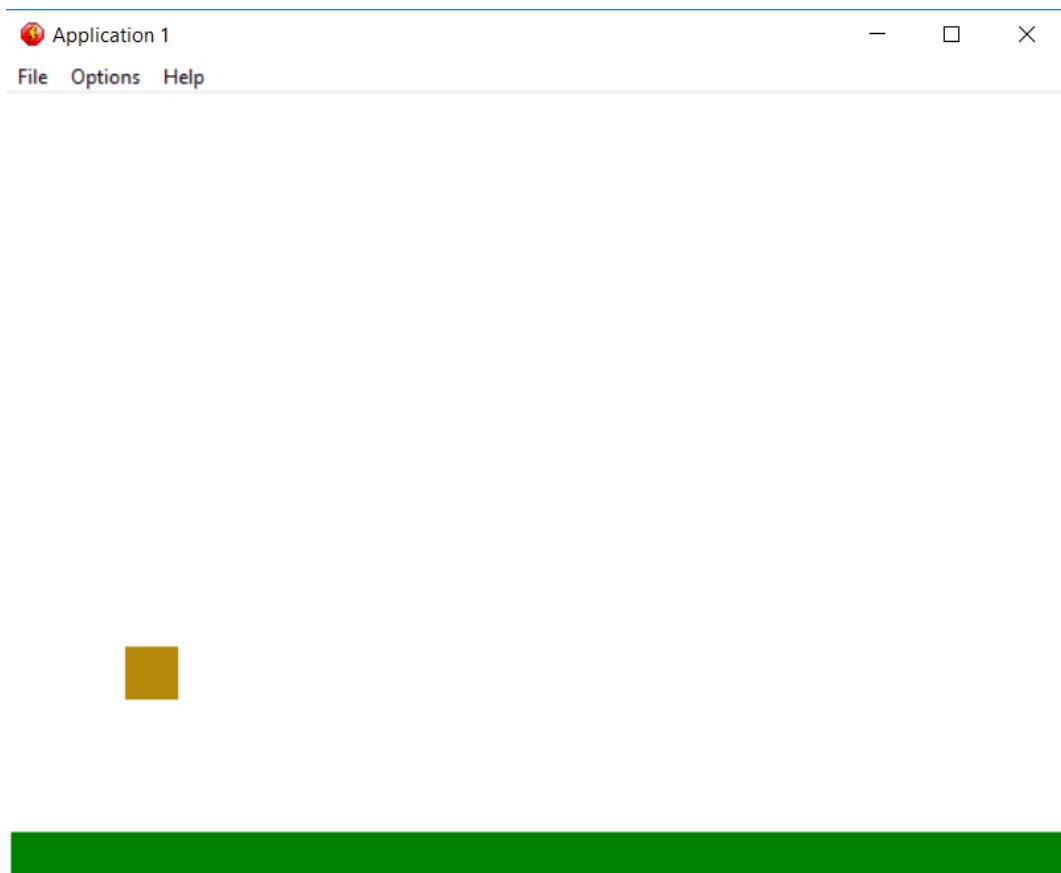


Figura 66: CTF Salto 2D

Tabla resumen funcionalidad.

Funcionalidad	Valoración	Puntuación
Salto.	Difícil	2.5

Movimiento personaje en 8 direcciones.

El proceso para la creación de un personaje que se mueva en 8 direcciones es el siguiente, véase Figura 55.

Ruta/comandos	Insert->New Object->Object tipo "Active"-> Propiedades -> Type seleccionar "Eight Directions"
Variables	<p>Player: Jugador que se va a mover.</p> <p>Directions: Direcciones posibles, aquí se puede restringir diagonales o alguna dirección en que el personaje no se quiera que se mueva.</p> <p>Initial Directions: Dirección a la que se desea que el Sprite esté mirando en el momento de iniciar el Frame.</p> <p>Speed: Velocidad de movimiento.</p> <p>Deceleration: Velocidad con la que el personaje desacelera hasta detenerse.</p> <p>Acceleration: Velocidad a la que el personaje acelera hasta alcanzar su velocidad máxima.</p> <p>Moving at Start: Determina si el personaje se está moviendo al inicio de la escena.</p> <p>Stick to obstacle: Si se marca esta opción el personaje no se moverá al colisionar con obstáculos, mientras que si se deja deselegionada el personaje retrocederá.</p>

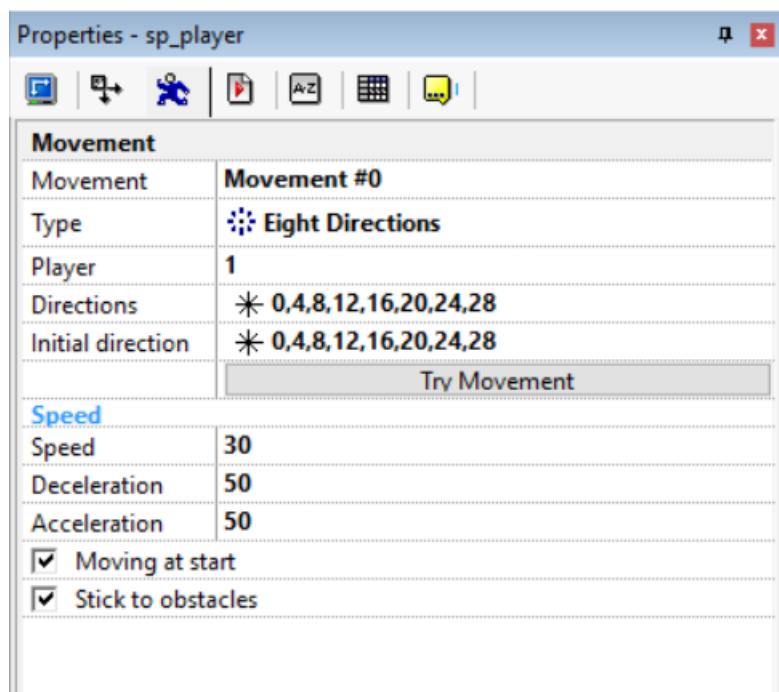


Figura 67: CTF Variables del movimiento en 8 direcciones

Tabla resumen funcionalidad.

Funcionalidad	Valoración	Puntuación
Movimiento 8 direcciones	Muy fácil	10

Disparo de bala.

Para que un objeto o personaje dispare balas los pasos a realizar son los siguientes.

1. Añadir un objeto de tipo activo que va ser el que va disparar.

Ruta/comandos	Seleccionar el Frame haciendo doble clic -> Insert de la barra superior -> New Object -> All Object -> Active

2. Pulsar dos veces sobre el objeto y se mostrará el menú de edición de sprites. En este menú, además de seleccionar desde un archivo la imagen o gráfico que se desea como Sprite hay que cambiarel punto de acción, que funcionará como centro de cara a los eventos, que es donde se va crear la bala inicialmente. Para posicionarlo en el lugar que se desea, véase Figura 56, habrá que pulsar sobre la opción "View hot spot" y aparecerá en el lugar en que ésta colocado

actualmente, y de la posibilidad de cambiarlo al lugar deseado. Una vez posicionado confirmamos.

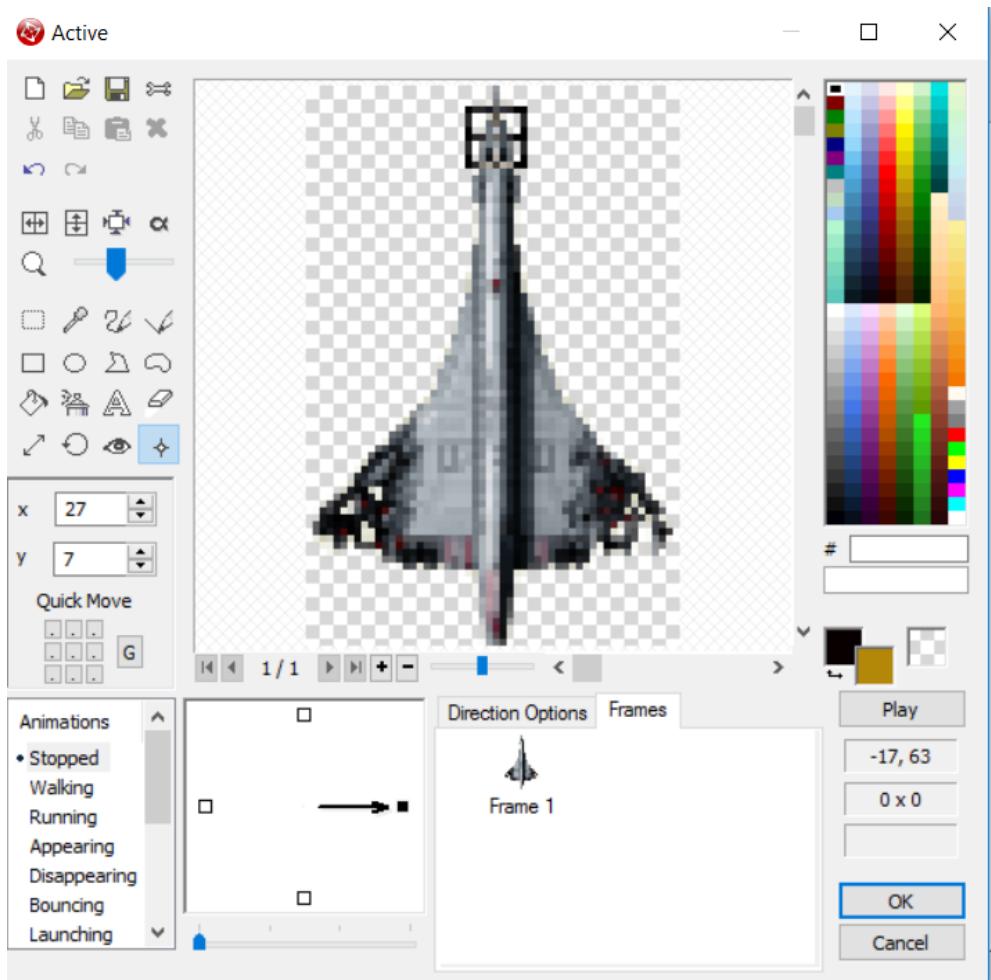


Figura 68: CTF Cambio de en Sprite

3. Se añade a la escena el otro objeto que va a actuar como bala de la misma forma que en el paso 1.
4. Se crea el evento de la pulsación de la tecla, mediante, la condición siguiente:

Ruta/comandos	
	Event Editor-> New Condition -> Opción Mouse and Keyboard ->The mouse -> User clicks -> Left click

5. Una vez implementada la condición para que suceda el evento de creación de la bala, ahora hay que realizar dicha acción.

Ruta/comandos	Botón secundario del ratón sobre la opción de la condición de creación (Cubo con un más) -> Create Object -> Seleccionar el sprite que hace de bala -> Opción Relative to -> Click sobre el sprite 1

6. Ahora se debe dar dirección a la bala para que no se quede estática tras ser creada en la opción anterior.

Ruta/comandos	Botón secundario del ratón sobre la opción de la condición del objeto bala -> Launch an Object -> Seleccionar bala.
Variables	<p>Speed of object: Velocidad del objeto.</p> <p>Direction : Elegir entre las siguientes tres opciones.</p> <ul style="list-style-type: none"> • Use the dirección of “object name”: El objeto se moverá en la dirección que se ha indicado en su posicionamiento y creación. • Launch in select directions...: Elegir la dirección en que la bala va a ser lanzada. • Launch in direction of... : Seleccionar un objeto hacia el cual la bala va a ser lanzada.

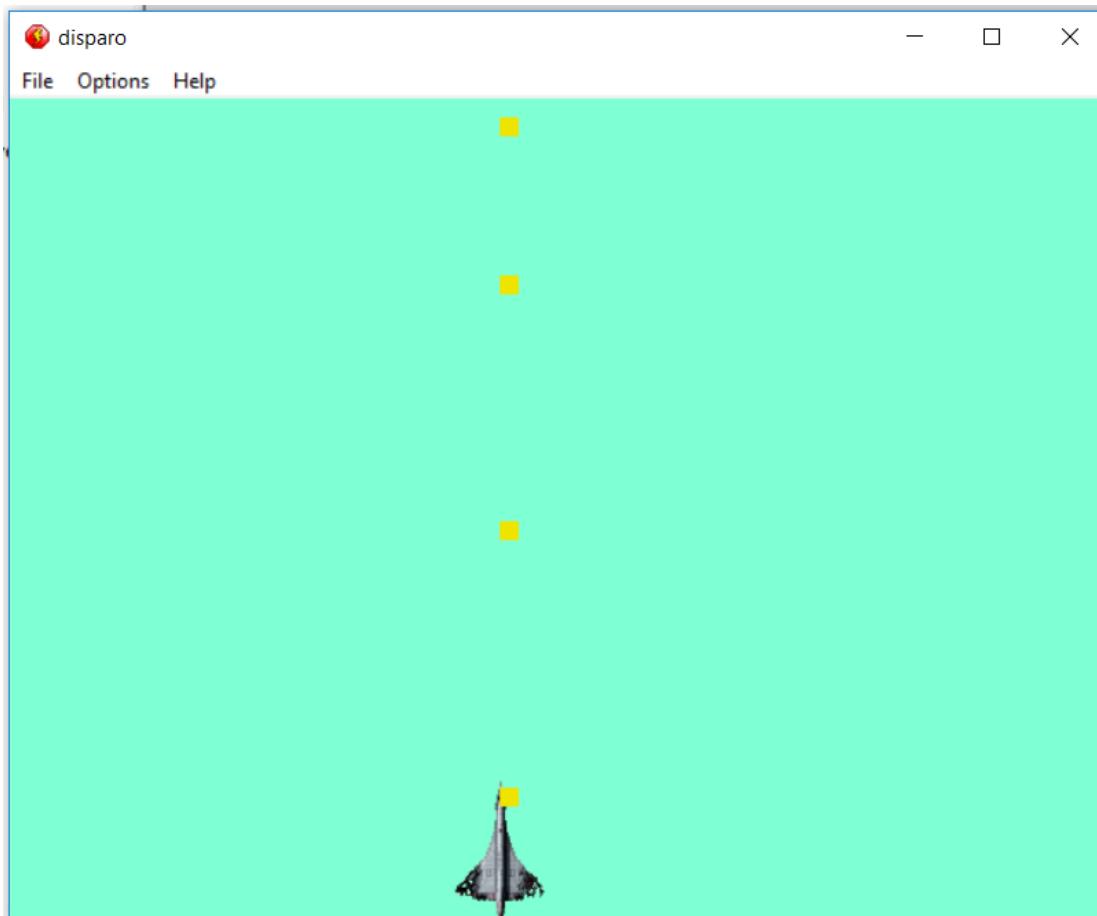


Figura 69: CTF Disparo de bala

Tabla resumen funcionalidad.

Funcionalidad	Valoración	Puntuación
Disparo	Media	5

Colisión con objeto.

La colisión entre objetos es una tarea bastante sencilla de realizar en Click Team Fusion.

Ruta/comandos	View->Event Editor->New condition ->Seleccionar Objeto y doble clic -> Colisions-> Another Object-> Seleccionar visualmente el segundo objeto implicado en la colisión.

Tabla resumen funcionalidad.

Funcionalidad	Valoración	Puntuación
Colisión	Muy fácil	10

Destrucción de objeto tras colisión.

1. Crear el evento con la condición “colisión” como se ha explicado en el punto anterior. Los objetos de la colisión serán por ejemplo Objeto 1, y Objeto 2.
2. Añadir la acción destruir uno de los objetos implicados en la colisión en el evento creado en el punto 1.

Ruta/comandos	Acción->Todos los objetos->Objetos->Eliminar el objeto
Variables	Objeto: El objeto que se quiere eliminar.

Tabla resumen funcionalidad.

Funcionalidad	Valoración	Puntuación
Destrucción tras colisión	Muy fácil	10

Resumen de funciones básicas y puntuación del subcriterio.

En la siguiente tabla se detalla la dificultad y la puntuación de cada una de las funcionalidades analizadas para CTF.

Función	Dificultad	Puntuación
Creación de un proyecto	Muy fácil	10
Insertar fondo a una escena.	Muy fácil	10
Movimiento del personaje en 2 direcciones	Fácil	7.5
Salto de personaje en 2 direcciones	Difícil	2.5
Movimiento de personaje en 8 direcciones	Muy fácil	10
Disparo de bala.	Media	5
Colisión entre objetos.	Muy Fácil.	10
Destrucción de Objeto tras colisión.	Muy Fácil.	10
PUNTUACIÓN FINAL		65
FACTOR MULTIPLICADOR DE CORRECIÓN		0.90
PUNTUACIÓN CON FACTOR DE CORRECIÓN		58.5

Tabla 111: CTF resumen funcionalidades

La tabla siguiente muestra la puntuación del criterio funcionalidades básicas de CTF escalada a la puntuación correspondiente a este criterio.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Funcionalidades básicas	0-12	8.77

Tabla 112: CTF subcriterio funcionalidades

Puntuación subcriterio lógica de programación necesaria

En Click Team Fusión la única referencia que se ha encontrado a la programación es el uso de la función Random, véase figura 58 .Todas las expresiones de programación o similares son implementadas por CTF mediante una interfaz visual que permite elegir visualmente sin tener que introducir las expresiones de programación.

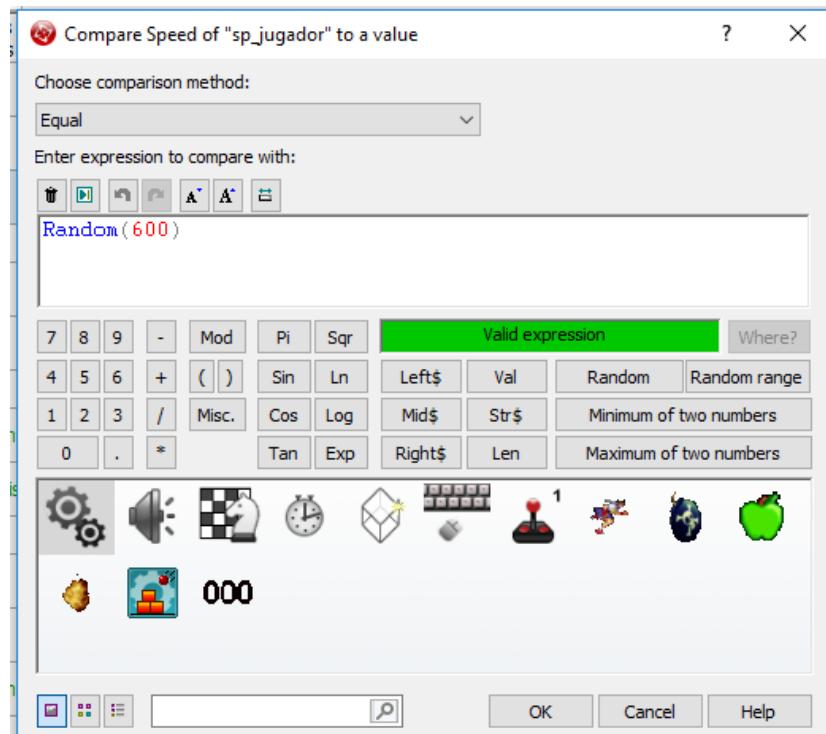


Figura 70: CTF uso de la función Random

La lógica de programación necesaria para usar CTF es mínima como se ha podido observar. Salvo pequeñas excepciones es posible desenvolverse completamente en el programa sin este tipo de conocimiento. Por lo que la puntuación en este apartado es casi la máxima.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Lógica de programación necesaria	0-4	3.5

Tabla 113: CTF subcriterio lógica de programación necesaria

Resumen puntuación funcionamiento y dificultad de uso

En la siguiente tabla se resume la puntuación de los cuatro apartados que componen este criterio.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Funcionamiento General	0-4	2
Primeros Pasos: Creación de un pequeño proyecto.	0-4	2
Funciones básicas	0-12	8.77
Lógica de programación necesaria	0-4	3.5
TOTAL		16.27

Tabla 114: Puntuación criterio funcionamiento y dificultad de uso

Criterio 4: Editor Gráfico/Visual.

Los datos obtenidos por CTF en la encuesta de valoración que evaluaba su interfaz son los mostrados en la siguiente tabla.

Concepto	Puntuación Máxima	Puntuación minima	Media (Sobre 5)
Rapidez al iniciarse	4	2	2,875
El programa se muestra rápido	4	3	3,625
La interfaz es atractiva	4	1	2,375
Tiene un aspecto cuidado y moderno	3	1	2
Puntuación diseño del programa	3	1	2,25
Los iconos representan bien los elementos Y/o funciones	4	2	2,75
A priori parece sencillo de usar	4	2	3,125
Consumo de recursos	4	3	3,625
MEDIA TOTAL			2,82/5
PUNTUACIÓN PONDERADA			4.5/8

Tabla 115: CTF Resultado encuesta interfaz

En la siguiente tabla se muestra la puntuación ponderada de CTF en el criterio editor gráfico/visual.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Editor Gráfico/Visual	Clave	4.5

Tabla 116: Criterio: Editor gráfico/visual

Criterio 5: Escalabilidad.

Puntuación subcriterio posibilidad de creación de funciones, scripts y/o lenguaje de programación propio.

CTF ésta pensado para no tener que recurrir a la creación de scripts y funciones en ningún momento y realizar todo mediante la interfaz propia del programa, ya que en ella se pueden introducir comportamientos bastante complejos mediante editores visuales o textuales.

Pero si se desea se pueden crear scripts propios con el lenguaje de scripting Lua²⁰.

Pero esta tarea resulta excesivamente compleja ya que apenas existe documentación ni en la página oficial (no se hace ninguna referencia al respecto) ni en los foros de desarrollo.

Por lo comentando, es posible crear funcionalidades propias es CTF, pero resultan una tarea muy compleja dada la poca documentación que existe al respecto, por lo que la puntuación en este apartado, como muestra la tabla siguiente es bastante baja.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Posibilidad de crear funciones y scripts propios y/o lenguaje de programación propio.	0-8	1.5

Tabla 117: CTF Subcriterio funciones propias

Click Team Fusión cuenta con una serie de extensiones o complementos oficiales, que mejoran el programa o partes del mismo o añaden funcionalidades nuevas. Los add-ons disponibles para el programa se pueden clasificar en los siguientes tipos:

- **Recursos:** Estos recursos se encuentran integrando el apartado de soporte dentro de la web del programa, constituyendo toda una serie de elementos como Sprites , sonidos, background , etc. que van a ayudar a realizar una primera aproximación al funcionamiento de CTF, ya que es muy importante que durante la fase de aprendizaje no se pierda excesivo tiempo en buscar este tipo de elementos.
- **Librerías 3D:** Conjunto de añadidos que permiten trabajar en 3D, para lo que el programa de base no está preparado. Por ejemplo, una extensión nos

²⁰ Lenguaje de programación imperativo, estructurado y bastante ligero que fue diseñado como un lenguaje interpretado con una semántica extendible.

permite realizar Raycast entre objetos de la escena Y otra que transforma la interfaz de Click Team Fusión para poder trabajar en 3D.

- **Shaders:** Añaden efectos gráficos, cambian la iluminación y el sombreado.
- **Librerías Graficas:** Packs de assets, personajes, escenarios u otros elementos que permiten construir las escenas y los juegos. Conjunto de elementos consistentes entre sí con los que se pueden crear juegos sin la necesidad de diseñar los elementos gráficos por parte del usuario.
- **Efectos:** Permiten añadir a los juegos efectos de diferentes tipos, de una manera fácil, rápida y visual. Ejemplos de este tipo de complemento podría ser efectos climáticos, explosiones o rayos.
- **Skins:** Cambian el aspecto visual de la interfaz y no aportan ninguna funcionalidad extra, es decir, simplemente modifican el diseño del entorno de trabajo.
- **Audio:** Conjunto de librerías con distintos tipos de sonidos, efectos y música. Son el equivalente de las librerías gráficas en el campo del sonido, y permiten al usuario no tener que crear por sí mismo o con ayuda de un tercero todo ese apartado del juego.
- **Aplicaciones y Herramientas:** Este tipo de complementos no aumentan las funcionalidades del programa, ni son extensiones del mismo. Son programas independientes, pero que complementan a CTF en la producción de un juego ya sea en la fase de creación de Sprites o elementos gráficos o en la etapa posterior cuando el juego está terminado. Como ejemplos de complementos de este tipo se encuentran un codificador de los textos en el juego para que no sean accesible desde los archivos fuente o de instalación, un creador de mapas de Tiles exportables directamente a Click Team o un creador de animaciones.
- **Otros:** Elementos que no pueden ser clasificado en ninguna de las categorías anteriores.

En la siguiente tabla se muestra la cantidad de cada tipo de extensión, su precio máximo y mínimo.

Tipo de extensión	Cantidad total	Precio mínimo	Precio máximo
Recursos	16	Gratis	Gratis
Librería 3D	6	0.99\$	11.99\$
Shaders	3	Gratis	11.99\$
Librerías Gráficas	90	Gratis	14.99\$
Efectos	34	0.99\$	24.99\$
Skins	15	Gratis	2.99\$
Audio	40	Gratis	40\$
Aplicaciones	48	Gratis	69.99\$
Otros	25	Gratis	39.99\$

Tabla 118: CTF Recursos

Como se ha comprobado el número de complemento disponibles para CTF es bastante alto y muchos de estos son gratuitos, por lo tanto, la puntuación en este apartado es bastante alta.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Add-ons y plug-ins	0-6	5

Tabla 119: CTF subcriterio addons y plugins

Resumen puntuación Escalabilidad.

La tabla siguiente muestra el resumen de la puntuación del criterio escalabilidad.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Posibilidad de crear funciones y scripts propios y/o lenguaje de programación propio.	0-8	1.5
Add-ons y plug-ins	0-6	5



Criterio 6: Resultados profesionales.

A continuación, se exponen los juegos más representativos desarrollados con CTF.

Saga Five nights at Freddy's.

Es quizás una de las sagas más conocidas creadas con CTF y además el que más se aleja del tipo de juego que se suelen crear con estos frameworks. Se trata de una aventura de terror al estilo clásico de aventuras gráficas en primera persona de *point and click*.

El videojuego trata sobre una pizzería en la que nuestro personaje deberá defenderse de los ataques de animales robots desde su sala de vigilancia. La jugabilidad del juego es bastante característica ya que el jugador no se puede mover del lugar inicial en el que aparece (sentado en su puesto de vigilante de la pizzería), y todo el desarrollo se realiza interactuando con los objetos disponibles en ese lugar.

El juego se hizo muy popular sobre todo en YouTube y plataformas de streaming, convirtiéndose en todo un fenómeno viral. Cuenta con 5 secuelas realizadas también CTF.

Datos Five nights at Freddy's.

Genero	Aventura gráfica, Terror
Lanzamiento	Agosto 2014
Nota Metacritic	78/100
Plataformas disponibles	Pc, Pc-Steam, Android, Ios , Windows Phone.

Tabla 120: Datos Five nights at Freddy's



Figura 71: Juego Five nights at Freddy's

The Escapists

The Escapists es un juego de aventura y acción con ciertos toques de estrategia y rol que se desarrolla dentro de una prisión y en el que el objetivo del jugador es escaparse pero sin dejar de lado la vida diaria de la prisión, véase Figura 60, su relación con otros presos y bandas, y cubrir las necesidad básicas del personaje como comer, dormir y beber.



Figura 72: The Escapists

El juego fue muy valorado tanto por la crítica como por los usuarios, por su planteamiento tan original, sus mecánicas y su estilo humorístico.

Datos The Escapists.

Genero	Acción-Aventura.
Lanzamiento	Febrero 2015
Nota Metacritic	71/100
Plataformas disponibles	Pc, Pc- Steam, Linux, Mac OS X, Xbox ONE*, PS4 ²¹

Tabla 121: Datos the Escapists

Freedom Planet

En la escena independiente actual existe un gran movimiento de retomar géneros que las grandes compañías han ido abandonando poco a poco. En esta idea se encaja Freedom Planet, todo un homenaje a los juegos plataformas de las consolas de 8 y 16 bits.

El juego se inspira en la saga Sonic de Megadrive, ya que el diseño de niveles, el aspecto visual de los mismos, los personajes y la jugabilidad son una evolución natural de aquella mítica saga, véase Figura 61, añadiendo algunos elementos como la presencia de ataques cuerpo a cuerpo, habilidades distintas por personajes o distintos modos de juego.

²¹ Las versiones de consolas han sido desarrolladas con CTF y pero para la exportación se ha usado Unity.



Figura 73: Freedom Planet

Datos Freedom Planet.

Genero	Plataformas
Lanzamiento	Julio 2014
Nota Metacritic	81/100
Plataformas disponibles	Pc, Pc- Steam, Linux, Mac OS X, Playstation 4, WiiU.

Tabla 122: Datos Freedom Planet

Not a hero.

Juego de acción en el que se van completando fases eliminando a todos los enemigos en pantalla. Tiene como principal novedad el uso de una perspectiva 2.5D gracias a la creación por parte de Devolver Digital ²² de complemento para CTF que permite esta vista.

²² Editora de videojuegos fundada en 2009.



Figura 74: No a hero

Datos Not a Hero.

Genero	Shooter 2D
Lanzamiento	Mayo 2015
Nota Metacritic	75/100
Plataformas disponibles	Pc, Pc- Steam, Linux, Mac OS X, Playstation 4, Xbox One, Android

Figura 75: Datos Not a hero

Resumen puntuación resultados profesionales.

Como se ha comprobado los juegos desarrollados con CTF tienen unas cotas de calidad bastante altas. Además, existen los géneros son muy variados no limitándose a un solo género u aspecto visual. Por lo que la puntuación en los dos apartados que componen este criterio es bastante alta.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Principales juegos desarrollados.	0-8	7
Diferenciación	0-4	3.5
TOTAL		10.5

*Tabla 123: CTF puntuación resultados profesionales***Criterio 7: Estado actual.**

Pese a que Click Team Fusión se mantiene activo y con actualizaciones constantes, ya está disponible en fase beta la versión 3, que será un producto distinto, y que se encuentra actualmente en la parte final de su desarrollo. Por esta razón es posible que la versión actual 2.5 deje de tener soporte en un periodo relativamente corto de tiempo.

Lo mismo es aplicable a la frecuencia de las actualizaciones ya que durante el ciclo de vida del programa éstas han sido frecuentes y con un tiempo relativamente corto entre ellas, tanto las relativas a añadir nuevas funcionalidades, como las de mejora de las ya disponibles y corrección de errores. Pero desde que se anunció la nueva versión, la frecuencia de las mismas se ha reducido llegando hasta el estado actual donde solo se utilizan para corregir pequeños errores o bugs.

La última actualización importante, la 2.5, que aportaba muchas novedades para los desarrolladores, fue lanzada en Febrero de 2016.

ESTADO	Activo
Última versión	2.5 Febrero 2016.
Frecuencia de actualizaciones	Desde el lanzamiento en Beta de la versión 3 poco frecuentes.

*Tabla 124: CTF Estado***Resumen puntuación criterio Estado actual**

Por lo visto anteriormente, el programa se encuentra activo, pero con un futuro incierto con el lanzamiento de la versión 3, y las actualizaciones cada vez son menos frecuentes por lo que la puntuación es la mostrada en la tabla siguiente.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Estado actual	0-2	1.5
Diferenciación	0-1	0.5
Frecuencia de actualizaciones	0-1	0.5
Total		2.5

Tabla 125: CTF Criterio estado actual

Criterio 8: Sistemas operativos soportados y requerimientos.

Los juegos creados con CTF se pueden exportar a los sistemas que indica la tabla siguiente.

Sistema/Plataforma	Exportación.
Windows	Si
Linux	No
iOs	Si
Android	Si
HTML 5	Si
Flash	Si
Windows Store (Mobile y Store escritorio)	Si
Sistemas PlayStation (PS3, PS-VITA, PS4)	No
Sistemas Xbox (XBOX360, XBOX ONE)	Si
Sistemas Nintendo (3DS, Wii, WiiU)	Si
RESUMEN Y PUNTUACIÓN PONDERADA	
TOTAL SISTEMAS	8 DE 10
PUNTUACIÓN PONDERADA	6.4

Tabla 126: CTF Plataformas de exportación

Resumen puntuación sistemas operativos soportados y posibilidades de exportación.

Después de realizar la ponderación la puntuación en este criterio es la indicada en la siguiente tabla.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Posibilidades de exportación: Sistemas y plataformas	0-8	6.4

Tabla 127: CTF puntuación sistema de exportación

Criterio 9: Licencia.

Click Team Fusión presenta varios modelos de negocio a los que los usuarios pueden acogerse. Los tipos de licencia que se pueden a la hora de utilizar el programa son los siguientes:

FUSION 2.5- FREE EDITION

Con esta versión gratuita podemos realizar las primeras pruebas con el programa y nos permite hacernos una idea de todo lo que se puede realizar. Puede ser un buen punto de partida para conocer el funcionamiento general y la interfaz del programa, pero las restricciones son bastante importantes e impiden que sea una opción para trabajar con él.

Las limitaciones más importantes de la versión free son las siguientes:

- Las posibilidades de exportación son prácticamente nulas.
- Editor de eventos y gestor de archivos de proyecto no disponibles.
- Imposibilidad de incluir videos externos en los juegos.
- El número de objetos en la escena ésta limitado.

Puntuación subcriterio versión gratuito.

Como se ha comprobado las limitaciones de la versión gratuita de CTF son tan abundantes que impiden considerarlo como una opción incluso para un usuario nuevo en el programa. Esta versión únicamente sirve para tener una idea del funcionamiento del programa. Por todo ello la puntuación en este apartado es mínima.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Modo gratuito	0-2	0.5

Tabla 128: CTF subcriterio modo gratuito

FUSIÓN 2.5

Elimina las limitaciones que tiene la versión free. añade toda una serie de características adicionales como son las siguientes:

- Posibilidad de utilizar filtros gráficos y de sonido.
- Exportar e importar la barra de herramientas y configuraciones del programa.
- Creación de nuevas barras de herramientas.
- Disponibles extensiones de movimiento de los objetos.

- Creación de SDK's.

La limitación principal de esta versión es que tienes que indicar en los juegos que han sido realizados con Click Team Fusion, así como añadir su logo al principio de la ejecución del mismo.

El precio de esta versión es de 75,99€.

FUSIÓN 2.5 Developer.

La versión Developer es la versión superior de todas las disponibles.

Elimina la necesidad de indicar que el juego se ha realizado con Click Team Fusión y añade las siguientes características:

- Permite cambiar la versión e información del archivo .EXE de la aplicación.
- Posibilidad de crear aplicaciones MID.
- Creación de sub-aplicaciones dentro de una misma aplicación.

El precio de esta versión es de 219,99€.

Puntuación de subcriterio precio de licencias

CTF tiene un coste máximo de licencia de 219.99€. La diferencia con el precio medio de las licencias de los programas analizados (179.25€) es de 40.74€. Con el baremo definido antes de comenzar el análisis a CTF le corresponde un 25% de total de la puntuación.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Precio de licencias	0-4	1

Tabla 129: CTF subcriterio precio de licencias

Puntuación de subcriterio precio módulo de exportación,

Como se ha visto anteriormente la exportación en los diferentes formatos y sistema operativos que no sean Windows requieren no solo una versión standard o Developer, sino que también el módulo correspondiente al formato que se quiere exportar. Los módulos disponibles se muestran en la siguiente tabla.

Sistema/Plataforma	Precio
iOs	99.83€

Android	69.58€
HTML5	59.49€
Windows Store y Xbox	180,50€
Flash	59.99€

Tabla 130: CTF precio módulos de exportación

En la siguiente tabla se muestra el precio de los módulos con influencia en la puntuación y su precio medio.

Sistema/Plataforma	Precio
iOs	99.83€
Android	69.58€
HTML5	59.49€
PRECIO MEDIO TOTAL	76.3€

Tabla 131: CTF Precio módulos de exportación y precio medio

Por lo visto y según el criterio de asignación de puntuación definido para el precio de los módulos a CTF le corresponde un 25% de la puntuación total de este subcriterio, ya que tienen un precio medio de módulos no superior a 50€ sobre el precio medio de todos los programas (39.9€).

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Precio de módulos de exportación	0-2	0.5

Tabla 132: CTF Subcriterio Precio módulos de exportación

Resumen puntuación Licencias.

A continuación, se exponen los resultados de CTF en los tres apartados que componen el criterio licencias.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN

Precio de Licencias	0-4	1
Modo de prueba o free	0-2	0.5
Precio módulos de exportación	0-2	0.5
TOTAL		2

Tabla 133: CTF Criterio licencias

Valoración final y resumen de puntuación

La puntuación total de Click Team Fusion ha sido de 56,67 sobre 100. En el apartado de análisis de resultados se dará una valoración a estos resultados.

En la siguiente tabla se puede ver la puntuación obtenida en los distintos criterios evaluables.

Criterio	Puntuación máxima	Puntuación obtenida	Puntuación porcentual
Requisitos e instalación	6	4,5	75,00%
Documentación, tutoriales y comunidad.	16	3,5	21,88%
Funcionamiento y dificultad de uso	24	16,27	67,79%
Editor Gráfico/Visual	8	4,5	56,25%
Escalabilidad	14	6,5	46,43%
Resultados profesionales	12	10,5	87,50%
Estado actual	4	2,5	62,50%
Sistemas operativos soportados y requerimientos.	8	6,4	80,00%
Licencias	8	2	25,00%
TOTAL	100	56,67	

Tabla 134: CTF Puntuación Total.

5.2.5. Análisis de Game Maker

Criterio 1: Requisitos e instalación.

En la siguiente tabla se detallan los requisitos de instalación de Game Maker.

REQUISITOS GDEVELOP	
Sistema Operativo	Mínimo Windows XP SP3, soportados Windows Vista, 7, 8 y 10, Mac.
Procesador	200Mhz o mayor
Memoria	256 MB de RAM
Tarjeta Gráfica	Mínimo Tarjeta Gráfica con soporte para Direct3D 9
Otros	Conexión internet necesaria para la instalación, actualización y descarga del software.

Tabla 135: Game Maker Requisitos de instalación

Puntuación subcriterio requisitos

Los requisitos de Game Maker son bastante bajos por lo que la mayoría de usuarios no tendrán problemas en trabajar con él. La puntuación se ve reducida en medio punto ya que no ésta soportado en sistemas Linux, aunque si en dispositivos Mac.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Requisitos de instalación.	0-4	3.5

Tabla 136: Game Maker puntuación subriterio requisitos

El proceso para la instalación de GM tiene los siguientes pasos:

1. En primer lugar, hay que registrarse, véase Figura 64, en la web de Yoyo Games ²³ ya que para acceder a la descarga de GM se necesita tener una cuenta de usuario.

²³ <https://account.yoyogames.com>

The image shows a registration form titled 'REGISTER'. It includes fields for 'Email' (username@domain.com) and 'Confirm email' (Re-enter your email address). A reCAPTCHA box contains the text 'No soy un robot' and a checkbox labeled 'reCAPTCHA'. Below the form are two checkboxes: one for accepting the User Agreement and Privacy Policy, and another for opting out of the mailing list. A green 'Register' button is at the bottom.

Figura 76: Proceso de instalación Game Maker

2. Con la cuenta de usuario se accede a la descarga del instalador de GM.
3. Se instala con el paquete descargado anteriormente el programa y reproductor de juegos.
4. Se ejecuta el programa por primera vez y se selecciona el modo, véase Figura 65, se quiera mantener el programa actualizado. Se puede seleccionar entre versión estable, donde se las actualizaciones constan de paquetes de actualizaciones probados por completo; o Beta, en este modo las actualizaciones se presentan cada poco tiempo pero pueden tener fallos o pequeños errores que es recomendable que se reporte a Yoyo Games.



Figura 77: GM modo de actualización

Resumen proceso de instalación

La tabla siguiente muestra el resultado de la instalación y el resumen de los datos obtenidos.

GAME MAKER Instalación en el equipo de pruebas	
Tamaño de descarga	115 Mb
Requiere programas adicionales	No
Instalación de módulos o recursos adicionales durante la instalación	No
Tiempo total	5 minutos 38 segundos.
Tamaño posterior a la instalación	152 Mb

Figura 78: GM instalación equipo de pruebas

Datos de instalación encuesta.

Los datos de la encuesta de valoración del proceso de instalación de Game Maker son los presentados en la siguiente tabla. Los datos completos del estudio se encuentran en el Anexo I: Resultados estudio de instalación e interfaz.

GAME MAKER datos estudio instalación	
Tiempo medio	4 minutos 46 segundos
Valoración media sencillez del proceso de instalación (0 a 10)	7.87
Valoración media de la funcionalidad del proceso de instalación (0 a 10)	9
Puntuación media funcionalidad y sencillez	8.43
Fallos en instalación.	No

Tabla 137: GM datos encuesta instalación

Puntuación subcriterio instalación.

El tiempo de instalación de GM es menor de 5 minutos por lo que le corresponde un punto en este apartado. Respecto a la valoración media de funcionalidad y sencillez a obtenido un 8.43 por lo que le corresponde 0.66.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Instalación.	0-2	1.66

Tabla 138: GM subcriterio instalación

Puntuación criterio requisitos e instalación

En la siguiente tabla se muestra la puntuación del criterio requisitos e instalación de GM.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Requisitos de instalación.	0-4	3.5
Instalación	0-2	1.66
TOTAL		5.16

Tabla 139: GM puntuación requisitos e instalación

Criterio 2: Documentación, tutoriales y comunidad.

GM tiene un completo manual disponible, véase Figura 67, en su web oficial en el que se explican todos los apartados del programa.

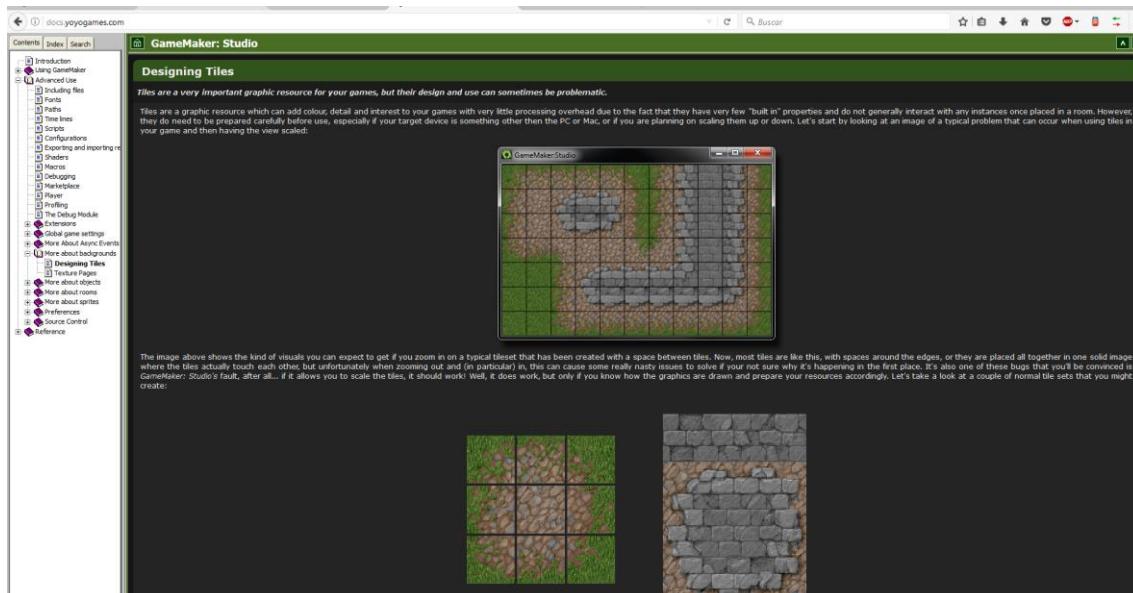


Figura 79: Manual online GM

El manual se presenta en formato documento online y sus apartados se clasifican en tres grandes áreas:

- 1. Using GameMaker:** En este apartado se incluye una explicación completa del uso básico del programa, así como una explicación de sus elementos fundamentales y su interfaz. Tiene cuatro sus subsecciones:

- Actions.
- Drop Down Menus.
- Events.
- Rooms.

- 2. Advanced Use:** Apartado para usuarios expertos en Game Maker o quieran hacer uso de las funciones avanzadas del programa. Dentro de este apartado se incluyen acciones como la creación de Scripts bajo el lenguaje propio de GM, **Shaders***, o debug sobre los proyectos. Cuenta con los siguientes subapartados:

- Extensions.
- Global game settings.
- More about Async events.
- More about backgrounds.
- More about objects.
- More about roms.
- More about sprites.
- Preferencias.
- Source Control.

3. References: Referencias de los apartados anteriores, aclaraciones, y explicaciones de conceptos.

Como se ha comprobado el manual es extenso y completo y es una buena fuente para consultar si surgen dudas durante el desarrollo o simplemente para aprender conceptos nuevos.

Aparte del manual en la web de Yoyo Games existe un apartado llamado "Base de conocimiento", véase Figura 68, en el que se van recopilando las dudas que los usuarios han ido teniendo sobre los distintos apartados del juego y han sido resueltas por los desarrolladores. En este apartado también se incluyen explicaciones de implementación de funciones que por su naturaleza quedan fuera el manual , se trata de funcionalidades que la comunidad de desarrolladores ha preguntado recurrentemente como realizarlas o por propia iniciativa de los miembros de Yoyo Games. Estos elementos de la base de conocimiento se suelen presentar en forma de manual con una explicación y una guía para su implementación incluyendo también códigos a modo de ejemplo. Incluyen un apartado de comentarios para que los usuarios puedan preguntar dudas respecto a la guía.

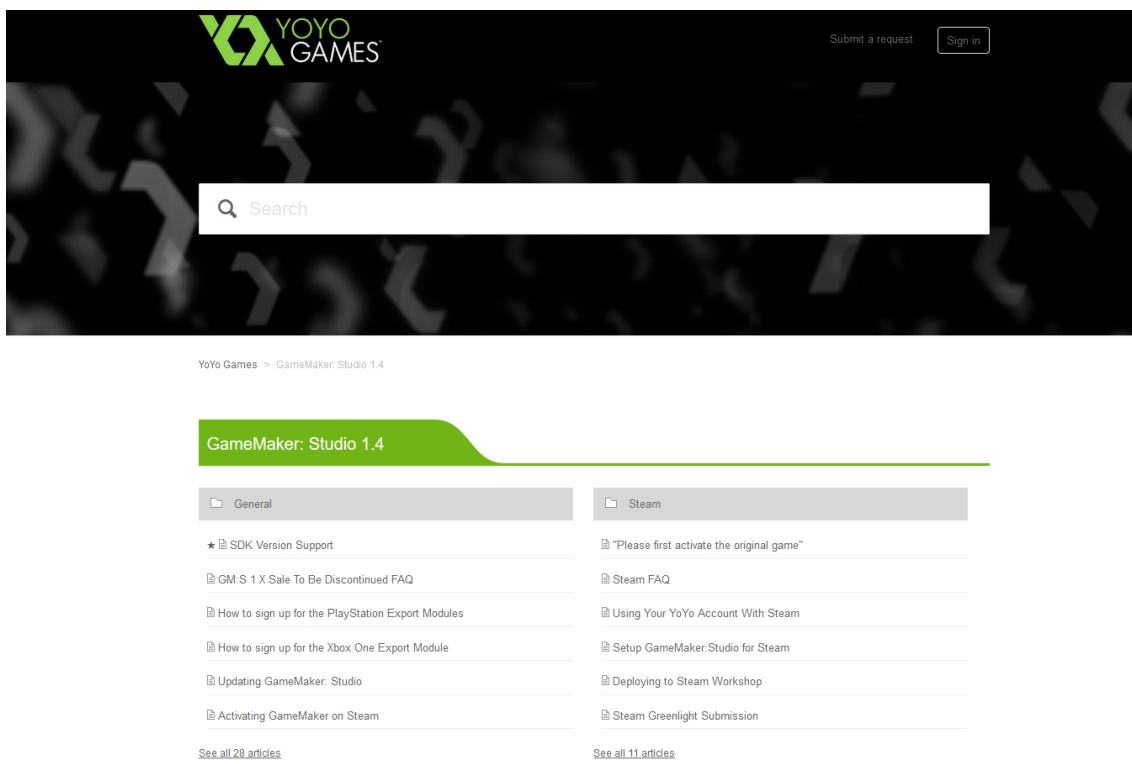


Figura 80: GM base de conocimiento

Respecto a los tutoriales los prestan de tres formas distintas: en la propia web de Yoyo Games, en el sitio oficial Game Maker Tutorials²⁴ y en el propio programa.

En la propia web de GM se encuentra una sección llamada Learn, en ella podemos encontrar varios tipo de videotutoriales; de nivel básico sobre los fundamentos de GM ; sobre la creación de proyectos de distinto tipo y de funcionalidades concretas . Los tutoriales en esta sección van cambiando cada cierto tiempo.

Los videotutoriales anteriormente comentando que van apareciendo rotativamente en la web de Yoyo Games son extraídos de la web Game Maker tutorials. Esta web creada por Shaun Spalding, community manager de Yoyo Games constituye el núcleo central de recursos para aprender a usar GM a todos los niveles. Los tutoriales incluidos en el sitio web se encuentran clasificados de la siguiente forma:

1. **Artículos:** No son videos al igual que el resto de los recursos de la web. Son artículos escritos que explican determinadas funcionalidades del programa. Todos los artículos van acompañados de códigos, imágenes y gif para completar las explicaciones. Esta sección la componen 5 tutoriales.

²⁴ <http://gammakertutorials.com/>

2. **Beginner:** Videos que se recomiendan realizar a los usuarios novatos en el uso de GM. Dentro de la categoría beginner se encuentran 12 tutoriales.
3. **General:** Son funcionalidades que pueden ser necesitadas y requeridas en cualquier tipo de proyecto y no son exclusivas de proyectos avanzados, sino que puede ser necesitadas a cualquier nivel y por cualquier tipo de usuario. Por ejemplo, el guardado y carga de partida, pausar el juego o las animaciones. 24 funcionalidades generales son explicadas en este apartado.
4. **Plataformas:** Videos de características propias de los juegos de tipo plataformas. Son 12 los tutoriales dedicados a explicar las mecánicas de este tipo de juegos.
5. **Legacy:** Videos con errores o desactualizados, pero con partes aún válidas. 7 tutoriales se encuentran dentro de esta sección.

Un aspecto importante de la web, que ayuda a los usuarios a seleccionar el tutorial que mejor se adapta a sus características, ésta en todos los tutoriales se encuentran categorizados por dificultad, véase Figura 69. También es importante destacar que una gran parte de ellos contienen el archivo fuente del proyecto y videos explicativos.

Además de los tutoriales presente en la página, la misma enlaza con el canal de Shaun Spalding donde existe otra gran cantidad de tutoriales además de los clasificados en la web.



General

These tutorials explain general GM concepts that can be applied to many different kinds of game. Some of them were designed with a specific genre in mind but can easily be applied elsewhere.

Tutorial	Description	YouTube	GMX	Difficulty
Sprite Origins	A lot of beginner problems are routed in not understanding sprite origins	YouTube	n/a	★
Particle Effects	Create a simple snowfield effect using GM's particle system	YouTube	n/a	★★
Animations	How to set up animations and swap between them	YouTube	n/a	★★
Android Setup	Setting up the Android module of GM:S	YouTube	n/a	★★
Text & Signposts	Drawing some basic text to the screen	YouTube	n/a	★★
Text boxes 1	Drawing text animated one letter at a time	YouTube	.GMX	★★★

Figura 81: GM categorización tutoriales

A parte de todos los tutoriales comentados en los apartados anteriores, GM cuenta también con los tutoriales integrados en el programa. Es una característica diferenciadora ya que es el único de los programas analizados que cuenta con dicha característica.

Una vez ejecutado el programa se encuentra disponible el apartado “tutoriales”, véase Figura 70, donde se existen 47 tutoriales completos.

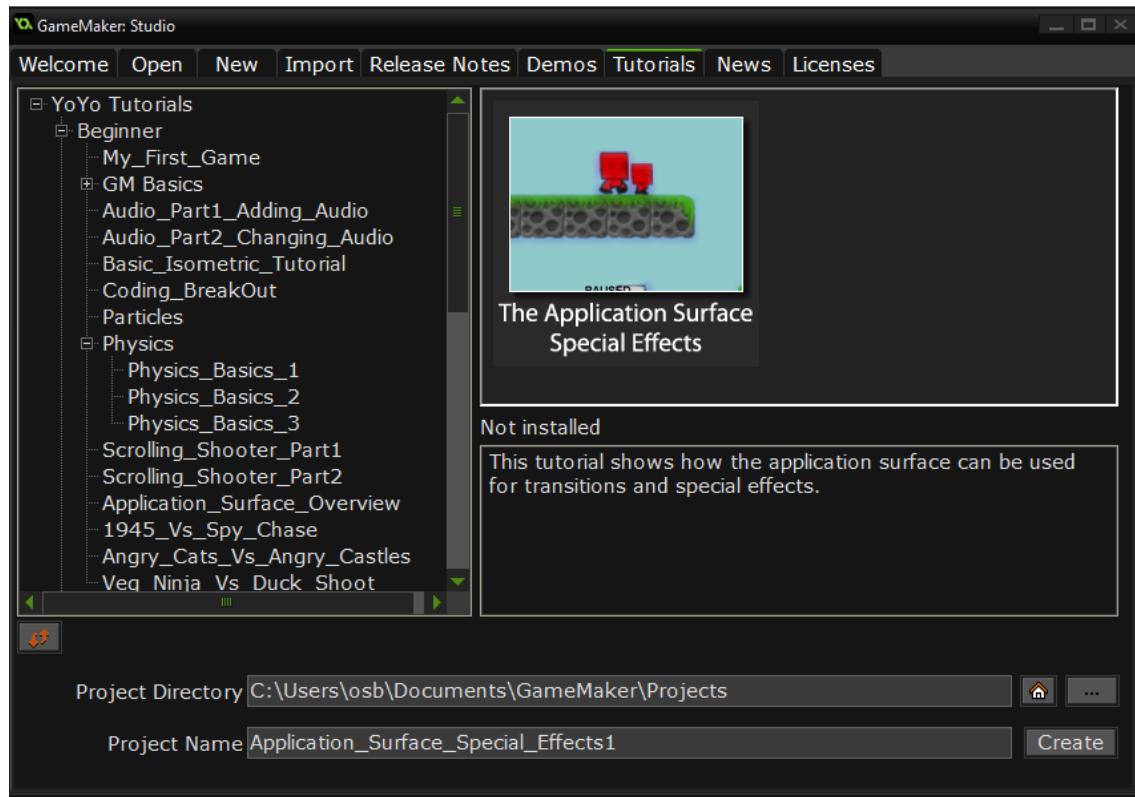


Figura 82: GM:.Índice de tutoriales en el programa

Lo más destacable e importante de estos tutoriales que los hace ser un valor añadido a GM es la forma de presentarse, véase Figura 71, una vez creado el tutorial mediante la opción correspondiente se abre un programa de proyecto con las siguientes características:

- Todos los Sprites y recursos gráficos que se necesitan para completar el tutorial.
- Los scripts y las funciones que el tutorial explica ya creadas.
- Las *roms* y objetos del juego.
- En la parte derecha del programa el tutorial en formato texto e imágenes.

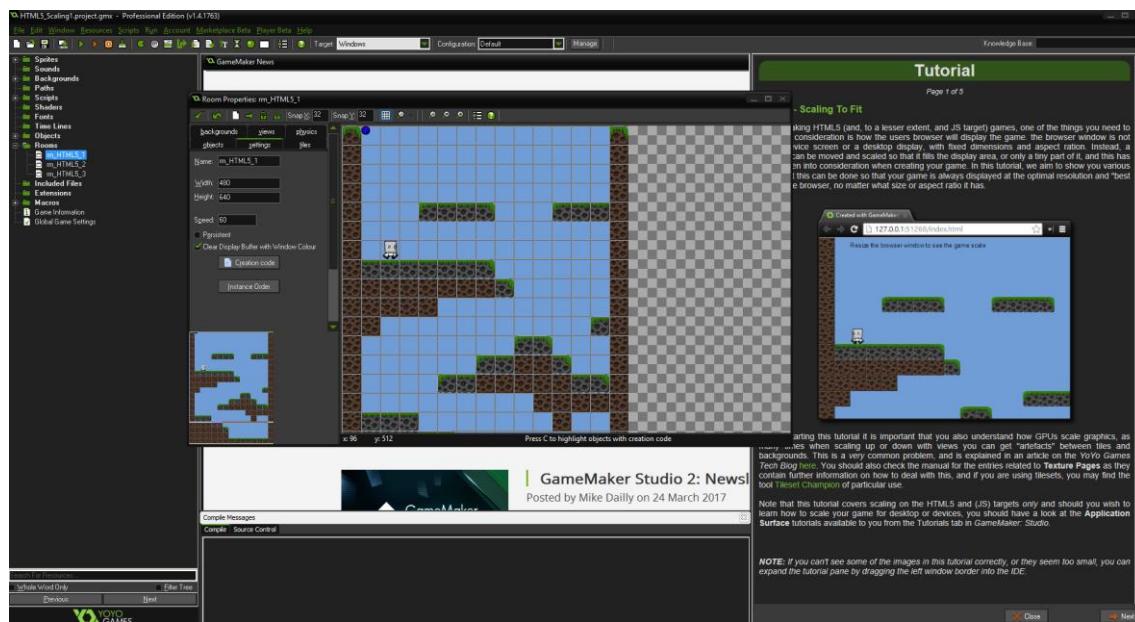


Figura 83: GM: Tutorial en el programa

Estas características hacen que este tipo de tutoriales sean una forma perfecta de aprender el programa, ya que en la misma interfaz de programa se tienen todos los elementos necesarios para implementar las funcionalidades y no se necesita tener que acudir a otros recursos y permite centrarse en el desarrollo del tutorial.

A parte de estos tutoriales también en el propio programa existe un apartado de Demos que son proyectos completos, pero sin el apartado de explicación, pero que también son una fuente importante de conocimiento.

Al igual que el resto de programas analizados GM también cuenta con un foro oficial donde se encuentran desarrolladores y usuarios. En foro destaca por la gran actividad y por el gran número de usuarios. Existen dos secciones muy interesantes para los usuarios que están acercándose por primera vez al programa como son los tutoriales , que son implementaciones realizadas por otros usuarios de funcionalidades completas y el apartado dedicado a dudas técnicas donde los usuarios pueden preguntar sus dudas respecto alguna implementación y el resto de usuarios intentan dar solución a dicho problema.

Puntuación subcriterio documentación, foros y tutoriales oficiales.

Por todo lo comentando anteriormente, GM en este aparto merece la máxima puntuación.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Documentación, foros y tutoriales oficiales	0-6	6

Figura 84: GM subcriterio documentación, foros y tutoriales oficiales

Tutoriales oficiales realizados.

Tutorial Scroll Shooter.

El tutorial Scroll Shooter, véase Figura 73, es el primero recomendado por Game Maker para aprender las funcionalidades básicas del programa.



Figura 85: GM: Tutorial Scroll shooter

Viene con todos los recursos gráficos necesarios para completarlo y con un proyecto para poder comparar las implementaciones que se realicen con el caso ideal.

El tutorial destaca sobre todo en los siguientes apartados:

1. Enseña a realizar las implementaciones de una forma y a que el usuario piense por si mismo otras maneras más optimas de implementar las cosas. Por ejemplo, en el tema de la eliminación de objetos tras destrucción, primero enseña la forma habitual, haciendo que los objetos se eliminen, pero posteriormente explica que es más conveniente si son objetos recurrentes no destruirlos sino moverlos a otra posición fuera de pantalla.
2. Al final el tutorial da unas pautas o próximos pasos se pueden dar para mejorar o hacer más atractivo el juego realizado, dichas funcionalidades extra se pueden implementar con los conocimientos adquiridos, lo que ayuda a practicar y afianzar los conocimientos adquiridos.

No se han realizado modificaciones.

Breakout.

El segundo tutorial que se ha realizado de Game Maker consiste en la creación de un Arkanoid ²⁵.

Una vez completado el tutorial los aspectos positivos que se han encontrado son los siguientes:

1. Contiene todos los recursos necesarios para poder realizar el tutorial, sprites, sonidos y fondos.
2. Se trata de un tutorial oficial y se realiza dentro del mismo programa, véase Figura 74, con lo que se visualiza a la vez que se realiza la implementación.

²⁵ Videojuego de arcade desarrollado por Taito en 1986. Está basado en los Breakout de Atari de los años 70.

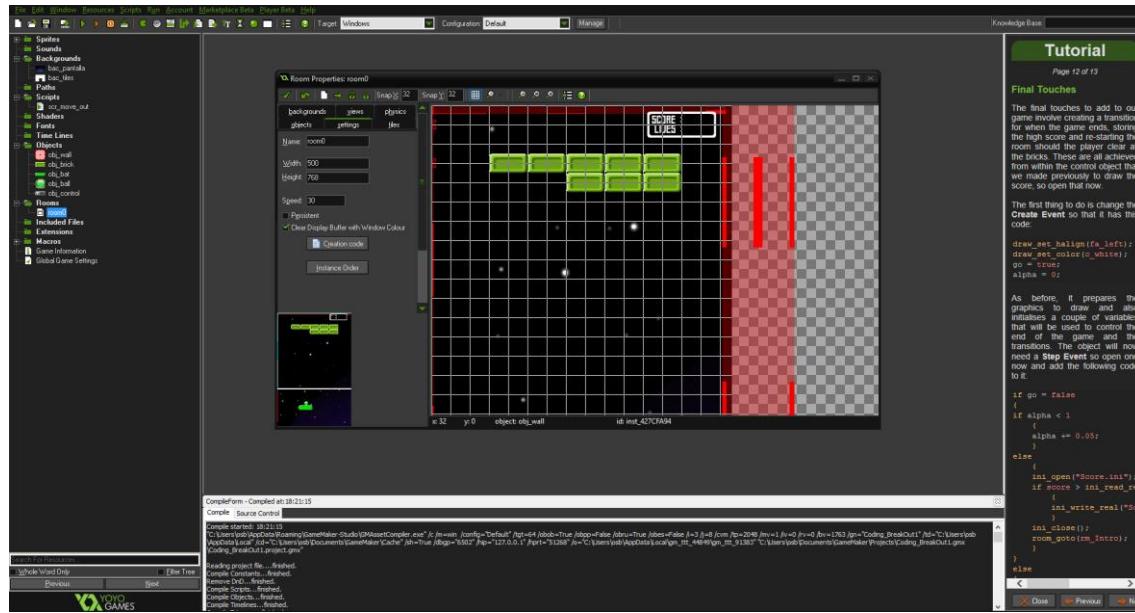


Figura 86: GM tutorial breakout

3. Incorpora links en aquellos apartados que requieren una explicación más avanzada o se desea ampliar conocimiento en la funcionalidad explicada en ese momento.
4. El tutorial una vez finalizado tiene un aspecto visual muy cuidado, con lo que la sensación de logro que obtienen los usuarios al realizarlo es mayor.

Aunque en líneas generales el tutorial es bueno se han encontrado los siguientes apartados mejorables:

1. En algunas partes el texto es demasiado denso y tiene pocas imágenes explicativas.
2. La explicación de los muros invisibles para que la pelota no sobrepase los límites de pantalla es bastante confusa y difícil de entender.
3. No se ha podido implementar el fondo animado, ya que el desarrollo de crear esa funcionalidad no estaba completo.

No se han realizado modificaciones, el resultado final del tutorial es el mostrado en siguiente Figura.

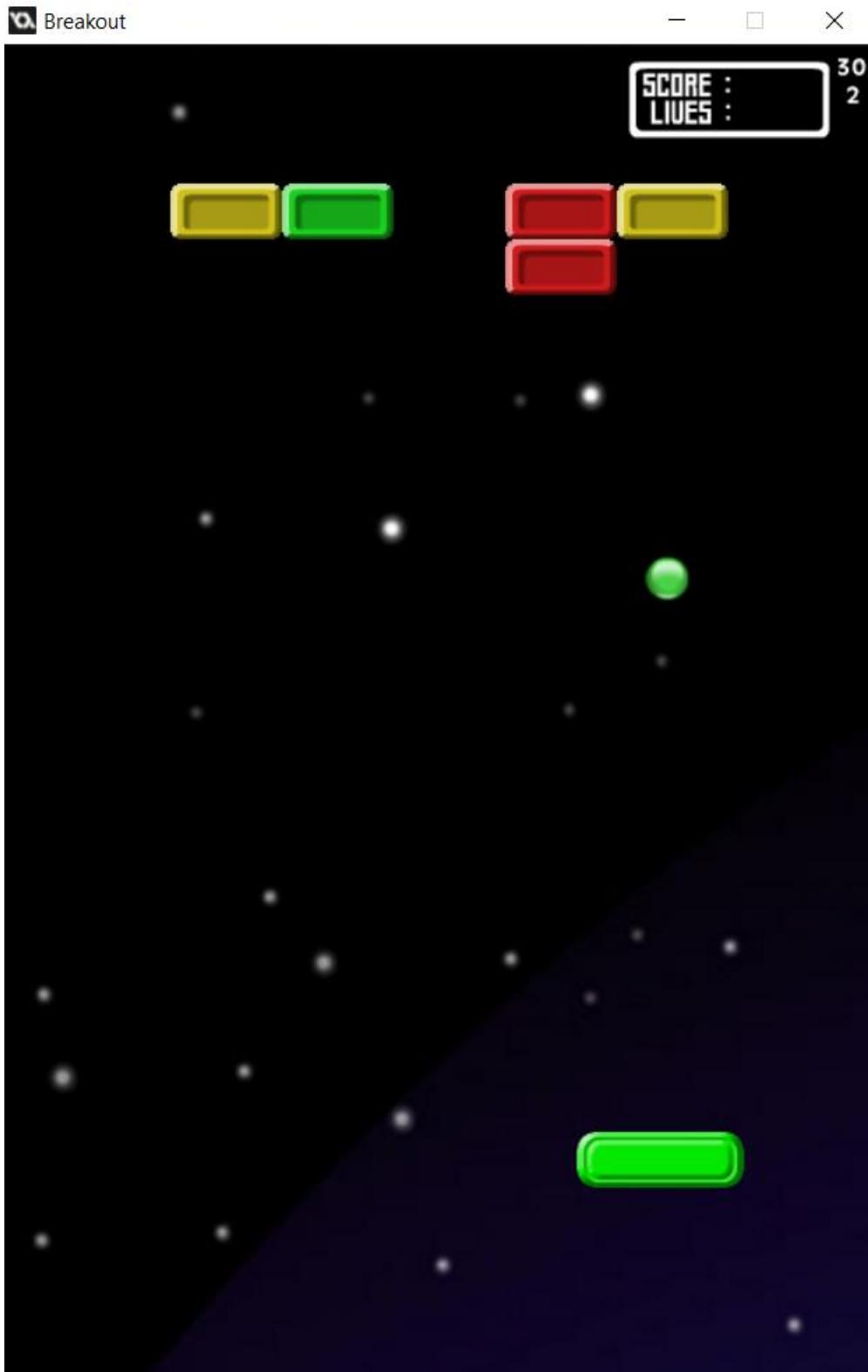


Figura 87: Resultado tutorial Breakout

Resumen de tutoriales realizados.

La siguiente tabla muestra el resumen de los tutoriales de GM realizados.

Tiempo dedicado.	8 horas 10 minutos
Tutoriales realizados	2
Tutoriales completados	2
Juegos completos	2
Errores detectados en los tutoriales	Muy pocos
Modificaciones realizadas	Muy pocas
Funcionalidades aprendidas en los tutoriales	<ul style="list-style-type: none"> • Animar personajes. • Insertar fondo. • Scroll automático de fondo. • Jerarquía entre objetos, herencia de comportamiento entre objetos padres e hijos. • Disparo de bala. • Control de balas con reloj. • Crear enemigos. • Controlador de enemigos para crearlos en lugares aleatorios, cada cierto tiempo. • Colisiones. • Destrucción de objeto tras colisión. • Gestión de sonidos y reproducción en disparo y colisiones. • Finalización de partida. • Mensajes por pantalla. • Puntuación global. • Gestión de vida de personaje • -Implementación de capas ocultas para gestionar colisiones. • Movimiento rebote de objetos, tipo pelota. • Movimientos aleatorios.

Tabla 140: GM Resumen toturiales oficiales realizados

Puntuación subcriterio análisis de tutoriales oficiales.

Como se puede comprobar en la tabla 107 los tutoriales realizados han sido completados, los errores encontrados y las modificaciones realizadas son prácticamente inapreciable y además el número de funcionalidades aprendidas es bastante alto, por todo ello GM en este apartado se merece la máxima puntuación.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Análisis de tutoriales oficiales	0-8	8

Tabla 141:GM: Subcriterio Análisis de tutoriales oficiales

La presencia de Game Maker en Reedit y Taringa se resume en los siguientes puntos.

Reddit

Casi 20.000 personas están suscritos al hilo de Reedit de Game Maker, véase Figura 74. Es un hilo constantemente activo y en el que se encuentran usuarios del programa de todos los niveles, desde principiantes hasta desarrolladores de juegos incluyendo a los creadores del propio programa.

Análisis de Herramientas para la creación de videojuegos | Óscar Sánchez

2

The screenshot shows the GameMaker subreddit page on Reddit. It displays several posts from different users, each with its title, author, and a snippet of code or text. The posts include topics like SegWit arguments, screenshots, GML beginner's guides, help requests for minimum jump height and particle depth, and a resolved issue about a mistake. The interface includes standard Reddit navigation elements like 'Subreddit Guidelines', 'Community Spotlight', 'Monthly Challenge', 'Community', 'Resources', and 'Other'.

Tabla 142: GM Hilo Reedit

Taringa.

Aunque Game Maker cuenta en Taringa con una comunidad con 260 miembros, se encuentra prácticamente inactiva ya que desde hace más de 1 año no existen temas ni contestaciones nuevas.

Puntuación subcriterio comunidades y foros de desarrolladores.

Por lo que se ha anteriormente GM solo tiene presencia significativa en Reedit, por lo que en este apartado le corresponde el 50% de la puntuación.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Foros y comunidades de desarrolladores	0-2	1

Tabla 143: GM: subcriterio foros y comunidades de desarrolladores

Resumen de puntuación Documentación, tutoriales y comunidad.

La tabla siguiente resume la puntuación del este criterio.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN

Documentación, foros y tutoriales oficiales	0-6	6
Análisis de tutoriales oficiales	0-8	8
Foros y comunidades de desarrolladores	0-2	1
TOTAL		15

Tabla 144: GM: puntuación criterio documentación ,tutoriales y comunidad.

Criterio 3: Funcionamiento y dificultad de uso.

Puntuación subcriterio funcionamiento.

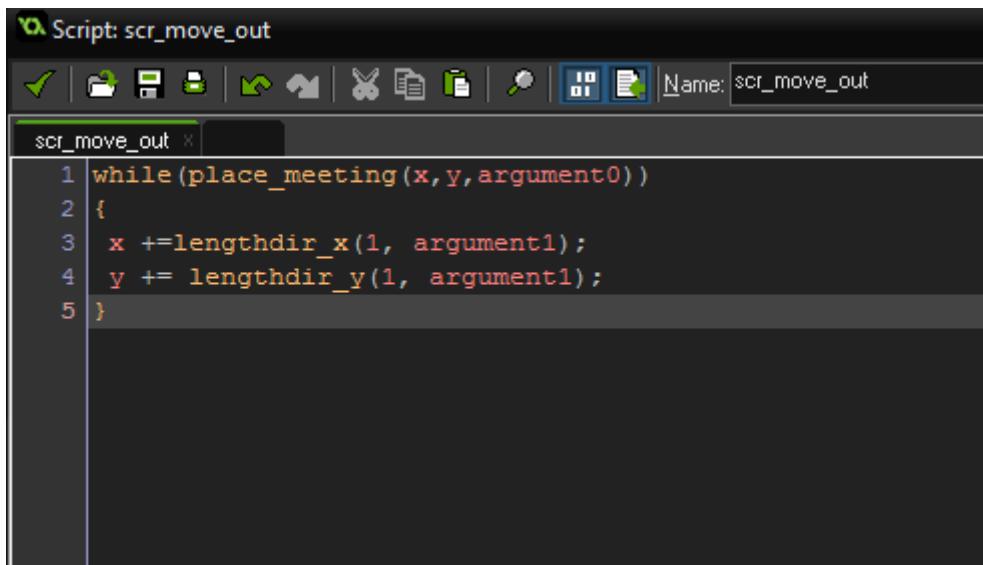
En el funcionamiento de Game maker destaca la opción de trabajar de dos formas dos formas diferentes:

1. **Game Maker modo Drag and Drop**

En este modo todas las funcionalidades se crean a partir de la interfaz visual de programa: seleccionado opciones, arrastrando elementos o eligiendo las funcionalidades mediante desplegable.

2. **Game Maker modo programación.**

Gracias al lenguaje de programación y de creación de scripts propio de GM, denominado GML, véase la Figura 76, se puede crear las funcionalidades sin hacer uso de la interfaz visual.



The screenshot shows the Game Maker Studio interface with the script editor open. The title bar says "Script: scr_move_out". The toolbar has various icons for file operations like save, open, and delete. Below the toolbar is a menu bar with options like File, Edit, View, Tools, and Help. The main window displays the script code:

```
scr_move_out
1 while(place_meeting(x,y,argument0))
2 {
3     x +=lengthdir_x(1, argument1);
4     y += lengthdir_y(1, argument1);
5 }
```

Figura 88: Game Maker lenguaje GML

Ambos modos no son excluyentes y pueden convivir dentro de un mismo proyecto, por lo que permite implementar partes algunas partes con programación visual y otras con GML. Por ello el programa se adapta a la percepción a toda clase de usuarios y niveles.

Los aspectos más destacables del funcionamiento y forma de trabajar de GM son los siguientes:

1. **Permite crear jerarquía y herencia entre los objetos.**

Se puede un objeto padre, como se realizaría en la programación clásica, y objetos hijos que heredan todo el comportamiento de su objeto padre y a los que se les puede añadir funcionalidades propias. Con esta herencia y jerarquía Game Maker permite ahorrar y reutilizar implementaciones con lo que el tiempo

de trabajo se maximiza.

2. Su sistema de organización de objetos, funciones y elementos.

Game Maker dispone de un sistema organizacional de elementos y funciones que estructura todo el juego de manera ordenada y clara, véase figura 77. Con esta opción se mejora la visualización de elementos y su búsqueda.

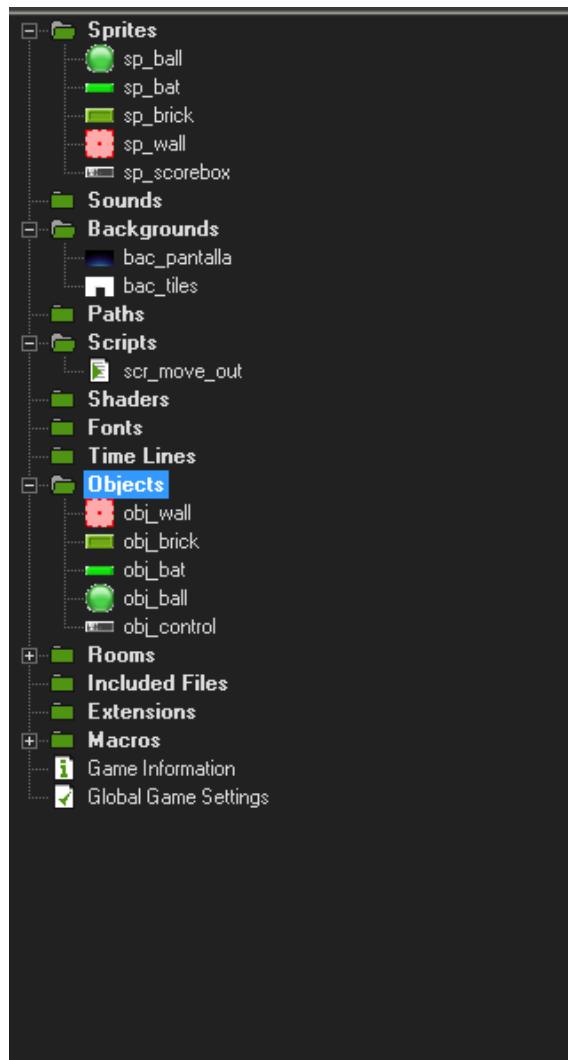


Figura 89: GM: organización por carpetas

3. Permite introducir comentarios en sus eventos y acciones.

Gracias a la introducción de comentarios, véase figura 78, se puede documentar las acciones y eventos que se implementan. Con ello se puede explicar las implementaciones de cara a compartir las con otras personas o de cara a futuras implementaciones propias.



Figura 90: GM : Sistema de comentarios

4. Informes de error detallados.

Uno de los grandes problemas de este tipo de frameworks es que cuando realizamos una implementación, se produce un error y el programa falla no se tiene información de que ésta sucediendo con lo que se pierde tiempo encontrar el error y más en proyectos con cierto tamaño. Game Maker para evitar esta situación cuando la ejecución falla muestra un error que detalla exactamente el objeto, funciones o variables que ésta provocando el error.

Puntuación subcriterio funcionamiento general

Por todos los aspectos que se han comentado, GM merece la máxima puntuación en este apartado.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Funcionamiento General	0-4	4

Tabla 145: GM subcriterio funcionamiento general

Puntuación subcriterio primeros pasos: creación de un pequeño proyecto.

Game Maker en un primer momento puede abrumar al usuario por la gran cantidad de opciones disponibles. Pero gracias a su gran organización, el usuario puede ir dominando pequeñas partes del programa sin necesidad de aprender todas las funcionalidades que pone a su alcance.

Superada esta pequeña barrera inicial, derivada de la gran potencia de GM , el usuario avanza rápidamente y con una constante sensación de logro, hecho que se demuestra

en que en apenas 8 horas se han podido completar dos juegos completos, véase Figura 74.

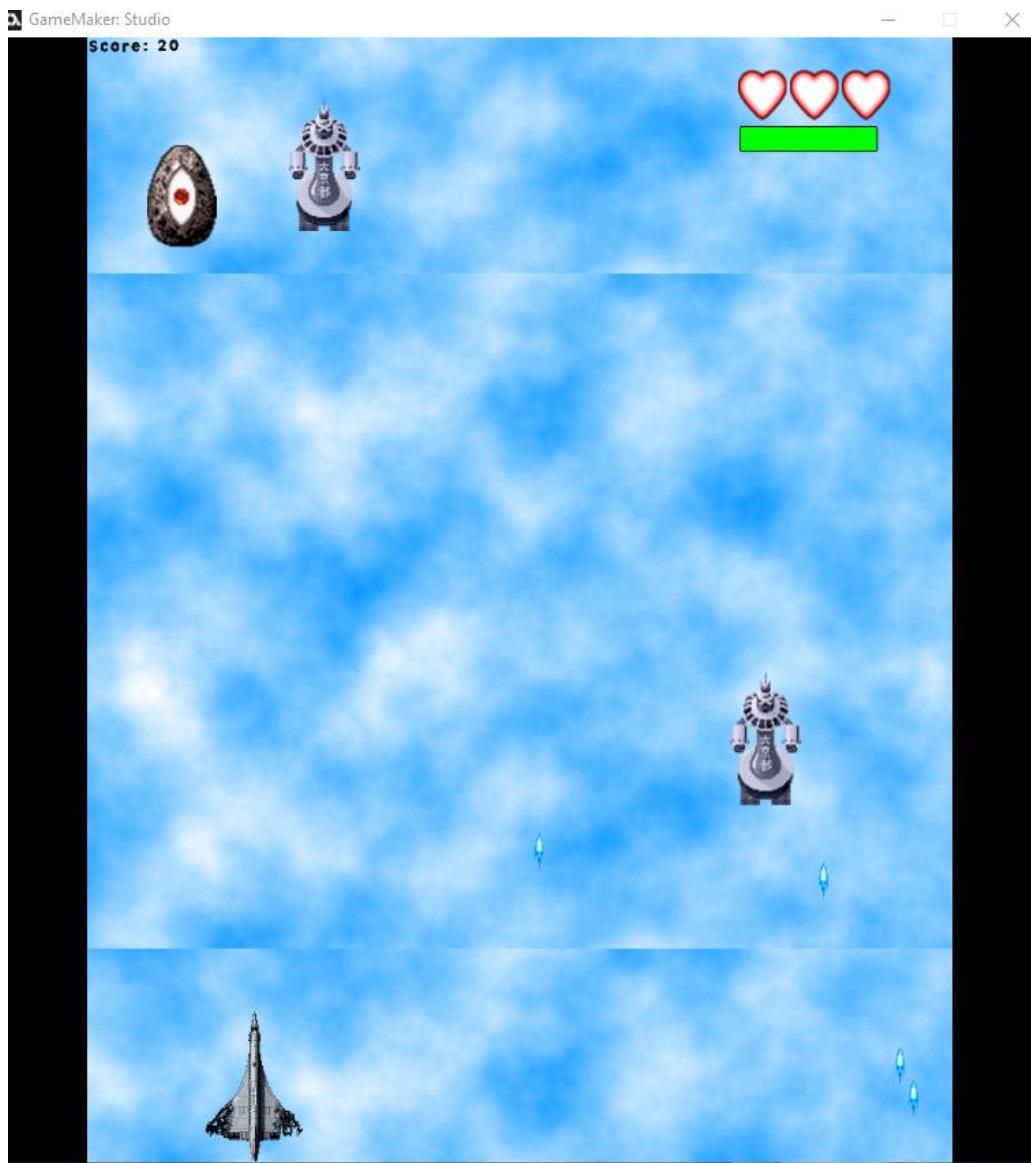


Figura 91: GM: juego desarrollado naves

GM tiene una curva de aprendizaje bastante asumible y adecuada para usuarios novatos, aunque hay que indicar que todo el aprendizaje va desembocar en la aplicación de funcionalidades a través de código, ya que es de esta forma en que el programa muestra todas sus capacidades. En este punto, los usuarios sin dominio de programación podrán tener dificultades, aunque como se ha comentado el proceso es muy progresivo y la mayoría podrá superar esta barrera.

Por todo ello la puntuación de GM es prácticamente la máxima en este apartado.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Primeros pasos: Creación de un pequeño proyecto.	0-4	3

Tabla 146: GM subcriterio funcionamiento general

Funciones básicas

A continuación se va a explicar cómo implementar las funciones básicas descritas en la definición de los criterios de selección y se va a puntuar en función de su dificultad.

Creación de un proyecto.

Crear un proyecto en Game Maker es una tarea muy sencilla, solo hay que seguir los pasos indicados en la siguiente tabla.

Ruta/comandos	New-Create Project
Variables	<p>Project Directory: Ruta donde se guardarán los archivos del proyecto.</p> <p>Project Name: Nombre del proyecto.</p>

Visto el proceso puntuación de la valoración es la siguiente.

Funcionalidad	Valoración	Puntuación
Creación proyecto.	Muy fácil	10

1 Insertar fondo en la escena.

Los pasos para insertar una imagen, véase Figura 80, como fondo de la escena tiene los siguientes pasos.

1. Crear elemento tipo Background

Ruta/comandos	Panel izquierdo -> Click secundario en apartado Background. -> Create Background -> Load from file para cargar una imagen desde un archivo.
Variables	Tile Horizontal: Check que confirma si el tile seleccionado como Sprite se va aplicar en horizontal.

	<p>Tile Vertical: Lo mismo que el anterior pero en vertical.</p> <p>Use for 3D: Se marca esta opción si se está trabajando en juegos 3D.</p> <p>Textura: Si el Tile lo permite aparecerán diversas texturas posibles a aplicar al tile que cambian su aspecto.</p>
--	---

2. Con el background creado habrá que insertarlo en la room o escena.

Ruta/comandos	doble clic sobre la room sobre la que queremos aplicar el fondo -> Panel izquierdo pestaña Background -> Opción Backgroung y elegir el creado en el punto anterior.
Variables	<p>Tile hor X : Los tiles de la escena que va abarcar el fondo horizontalmente.</p> <p>Tile Ver Y: Los tiles que va ocupar verticalmente el fondo.</p> <p>Hor Speed : Velocidad que fondo se va mover horizontalmente (con scroll)</p> <p>Ver Speed: Velocidad del Scroll vertical .</p>

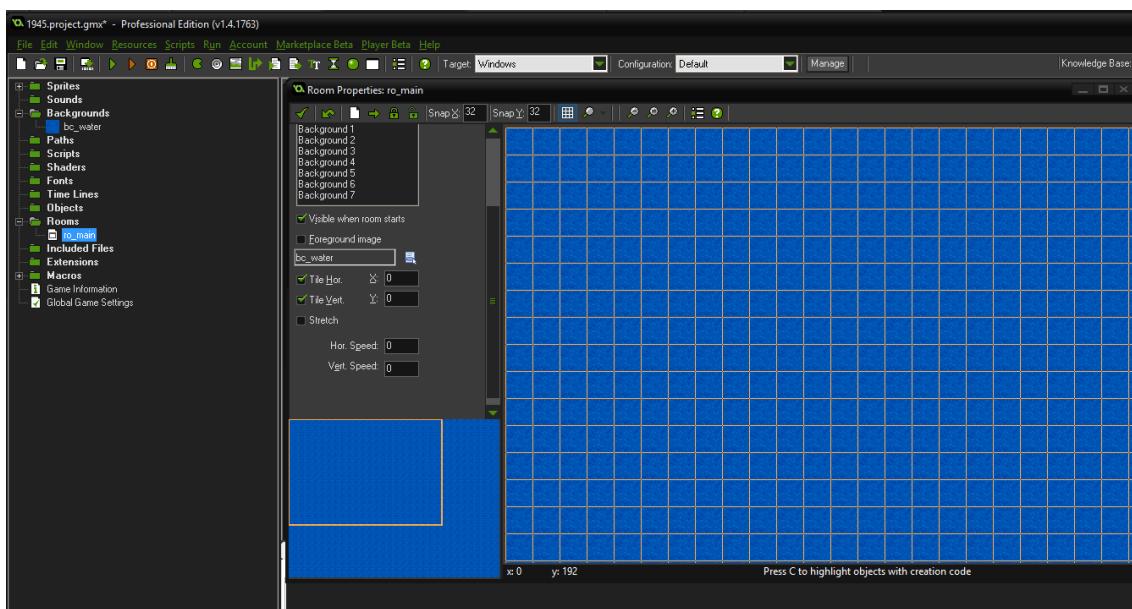


Figura 92: GM insertar fondo

Tabla resumen funcionalidad.

Funcionalidad	Valoración	Puntuación
Insertar fondo	Muy fácil	10

Movimiento de personaje en dos direcciones.

Los pasos a seguir para la crear la funcionalidad del movimiento en 2D de un personaje son los siguientes:

1. Crear los dos sprite, el de la plataforma o suelo y el de personaje.

Ruta/comandos	Boton secundario sobre Object en el panel izquierdo -> Create Objects -> En la ventana de edición de objeto seleccionar el Sprite que corresponda al suelo.
----------------------	---

2. Dar el comportamiento de plataforma a al suelo.

Ruta/comandos	Boton secundario sobre Object en el panel izquierdo -> Create Objects -> En la ventana de edición de objeto seleccionar el Sprite que corresponda al suelo -> Acciones ventana de objeto -> Marcar opción Solid
----------------------	---

3. Añadir al objeto creado el comportamiento de movimiento en 2 direcciones:

Ruta/comandos	Doble clic en objeto -> Add Evento y añadirmos evento keyboard -> Seleccionar cada una de las teclas y le añadido el evento movimiento en horizontal. Por ejemplo para la derecha: Seleccionar Right ->En acciones acción Elegir Jump to X 4 (X horizontal , positivo derecha y negativo hacia la izquierda Y vertical, positivo hacia abajo positivo hacia arriba)y marcar la opción relativo esto significa que se movera hacia la derecha desde donde este cuando mayor sea el número más rápido se movera.
----------------------	--

Tabla resumen funcionalidad.

Funcionalidad	Valoración	Puntuación
Movimiento en dos dimensiones	Fácil	7.5

Salto de personaje que se mueve en dos direcciones.

El proceso de salto en Game Maker se implementa de la siguiente forma :

1. Evento pulsación de tecla en jugador.

Ruta/comandos	Doble clic en jugador-> Add Event -> Tipo de Event Keyboard -> Seleccionar tecla deseada para saltar.
----------------------	---

2. Añadir condiciones dentro del evento de pulsación para que solo suceda cuando el objeto este en el suelo.

Ruta/comandos	Dentro del evento anterior-> Acción if here is collision at position
Variables	<p>X: Marcar 0 si no se quiere que se tenga en cuenta la dirección horizontal para que se cumpla la condición, 1 si se quiere que se cumpla.</p> <p>Y: Marcar 0 si no se quiere que se tenga en cuenta la</p>

	<p>dirección vertical para que se cumpla la condición, 1 si se quiere que se cumpla.</p> <p>Objets: Seleccionar objeto o marcar Only Solid para que se aplique a todos los objetos sólidos.</p> <p>Relative: Marcar se se quiere q se tenga en cuenta la condición sobre el objeto de manera relativa.</p> <p>Not: Marcar para indicar acción contraria.</p> <p>Para el salto los datos son los siguientes, x valor cero, y valor 1, objetos todos los sólidos y relativo si.</p>
--	--

3. Dentro de la acción anterior hay que añadir la acción para que el personaje salte.

Ruta/comandos	Insertar acción set vertical speed debajo de la del paso anterior -> Inserta velocidad , es decir lo que se desea que el personaje salte .
---------------	--

Tabla resumen funcionalidad.

Funcionalidad	Valoración	Puntuación
Salto.	Fácil	7.5

Movimiento personaje en 8 direcciones.

El proceso para la creación de un personaje que se mueva en 8 direcciones es el siguiente:

1. Insertar el Sprite del personaje.

Ruta/comandos	Panel izquierdo, botón secundario sobre opción Srpite -> Create Sprit -> Load Sprite y seleccionar la imagen.
---------------	---

2. Creación del objeto personaje.

Ruta/comandos	Panel izquierdo , botón secundario sobre opción Objects -> y Create Object.
Variables	<p>Sprite: Habrá que seleccionar el creado anteriormente.</p> <p>Visible: Determina si el objeto es visible en la escena o permanece oculto.</p> <p>Solid: Si el objeto es sólido podrá colisionar con otros objetos del mismo tipo en la escena.</p> <p>Persistent: Seleccionar si se desea que el objeto sea persistente en el ciclo de ejecución del juego.</p> <p>Use Physics: Posibilidad de usar las funciones físicas de Game Maker.</p> <p>Depth: Profundidad del objeto</p> <p>Parent: En GM existen herencias entre objetos, si este objeto tiene un padre indicar aquí.</p> <p>Mask: Mascará usada para las colisiones con otros objetos.</p>

3. Añadir al objeto creado el comportamiento de movimiento en 8 direcciones.

Ruta/comandos	En la misma pesataña de creación de objeto -> Add Evento y añadirmos evento keyboard -> Seleccionar cada una de las teclas y le añadimos el evento movimiento en horizontal io vertical. Por ejemplo para la derecha: Seleccionar Right ->En acciones acción Elegir Jump to X 4 (X horizontal , positivo derecha y negativo hacia la izquierda Y vertical, positivo hacia abajo positivo hacia arriba) y marcar la opción relativo esto significa que se movera hacia la derecha desde donde este cuando mayor sea el número más rápido se movera.
----------------------	--

4. Opcional: Es recomendable añadir a cada dirección un controlador para que no supere los límites de la pantalla. Por ejemplo, en el caso de que la pantalla sea de 40 de acho y 120 de alto habrá que añadir una condición if antes del evento de movimiento de la siguiente forma:

Ruta/comandos	añadir arriba de evento de movimiento una acción de tipo "Test variable" y con los siguientes datos para cada dirección: Izquierda: Variable X valor 40 y operación Mayor que. Derecha: Variable X valor room_width-40 y operación Menor que. Arriba: Variable Y valor 40 y operación Mayor que. Abajo: Variable Y valor room_height-40 y operación Menor que.
----------------------	--

Tabla resumen funcionalidad.

Funcionalidad	Valoración	Puntuación
Movimiento 8 direcciones	Fácil	10

Disparo de bala.

Para que un objeto o personaje dispare balas los pasos a realizar son los siguientes.

1. Crear objeto personaje como se ha creado en los puntos anterior.
2. Crear objeto bala.

Ruta/comandos	Secundario sobre objects y create object -> Seleccionar el Sprite de la bala -> Se añade el evento Create (Este evento sucedera cuando el objeto se crea) -> Se añade la opción al evento Set the vertical Speed con un valor negativo de -8 (conforme el número sea menor la velocidad será mayor)
----------------------	---

3. Para que la bala se dispare hay que añadir la acción al objeto que la va disparar mediante la pulsación de una tecla.

Ruta/comandos	Doble clic sobre el objeto ->Add event seleccionar Keyboard y elegir la tecla que va disparar y se le inerta el evento create instance con el objeto bala.
----------------------	--

4. Es recomendable añadir a la bala la opción de eliminación cuando este fuera de la pantalla.

Ruta/comandos	Doble clic sobre el objeto bala -> vento tipo Outside Room dentro de la categoría otros -> Destroy the instance.
----------------------	--

Para mejorar el disparo de la bala y que no se estén creando constante se le puede añadir un reloj con una variable para que solo se puedan disparar balas en intervalos. Para realizar esta implementación los pasos son los siguientes:

1. Crear el evento Create en el objeto y a este evento hay que enlazarle la acción Set Variable para que al crearse el objeto se cree también una variable llamada por ejemplo can_shoot (esta variable hará de reloj para disparar o no) con un valor de true.
2. Modificamos las acciones del evento de disparo de la bala añadiendo en primer lugar la condición de que únicamente dispare si la variable can_shoot tiene el valor true Para ello añadimos la acción Test variable comprobando que la variable can_shoot tiene el valor true, para que el disparo sea un subevento tenemos que colocar la acción test en el lugar 1 insertar rodeando el disparo un star block y end block (apartado de acciones control y dentro de este other) con ello el disparo solo se realizará cuando se cumpla la condición impuesta.
3. Se introduce en este bloque la acción set variable can_shoot false.
4. Junto a las acciones anteriores también dentro del bloque se añade una cronómetro o alarma que va empezar a contar desde el momento en que se disparó , set alarm en acciones de tipo timing con un valor de 15 steps en la alarma 0.
5. Insertar nuevo evento de tipo alarma que examine la alarma 0 (Add Event->Alarm->Seleccionar alarm 0). Este evento sucederá cuando el cronómetro

incluido en el paso anterior llegue a 15 steps (no segundos).

6. Al evento anterior habrá que añadirle una acción set variable y volver a poner la variable can_shoot a true.

Con esto tendríamos un control sobre el disparo para evitar el disparo de infinitas balas de manera continua.

Tabla resumen funcionalidad.

Funcionalidad	Valoración	Puntuación
Disparo	Fácil	7.5

Colisión con objeto.

Los pasos para crear la colisión son los siguientes:

1. . Añadir el evento colisión, véase Figura 81, a uno de los objetos de la colisión.

Ruta/comandos	Doble click sobre el objeto-> Add Event ->Colision-> Seleccionar el segundo objeto de la colisión.

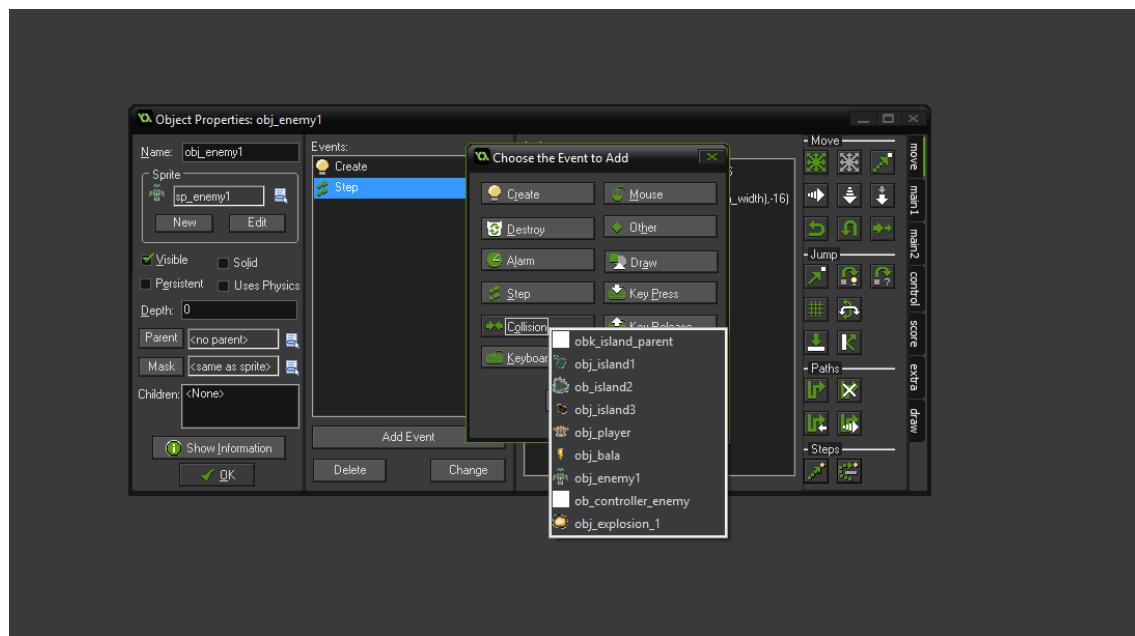


Figura 93: GM: Evento colisión

Tabla resumen funcionalidad.

Funcionalidad	Valoración	Puntuación
Colisión	Muy fácil	10

Destrucción de objeto tras colisión.

Para destruir objeto tras colisión partimos de la funcionalidad anterior, con dos objetos y un evento colisión entre ellos y después le seguirían los siguientes pasos:

1. Al evento de colisión se le añade la acción de destruir instancia.

Ruta/comandos	Direcciones
	Doble clic sobre el objeto -> clic sobre el evento Colision -> Seleccionar acción Destroy instance dentro de las acciones relativas a objetos -> En el menú contextual elegir la opción Self en el selector Applies to, si se quiere que también se destruya se crea una nueva opción destroy instance y se marca la opción Other.

Tabla resumen funcionalidad.

Funcionalidad	Valoración	Puntuación
Destrucción tras colisión	Muy fácil	10

Resumen de funciones básicas y puntuación del subcriterio.

La siguiente tabla resumen las funcionalidades, su dificultad y su puntuación.

Función	Dificultad	Puntuación
Creación de un proyecto	Muy fácil	10
Insertar fondo a una escena.	Muy fácil	10
Movimiento del personaje en 2 direcciones	Fácil	7.5
Salto de personaje en 2 direcciones	Fácil	7.5

Movimiento de personaje en 8 direcciones	Muy fácil	10
Disparo de bala.	Fácil	7.5
Colisión entre objetos.	Muy Fácil.	10
Destrucción de Objeto tras colisión.	Muy Fácil.	10
PUNTUACIÓN FINAL		72.5
FACTOR DE CORRECIÓN		0.85
PUNTUACIÓN CON FACTOR DE CORRECIÓN		61.62

Tabla 147: GM resumen de funciones básicas

Por lo visto en la tabla anterior la puntuación de las funciones básicas es de 61.62 sobre 80, que escalada al valor del criterio funciones básica es de 9.25.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Funciones básicas	0-12	9.25

Tabla 148: GM subcriterio funciones básicas

Puntuación subcriterio lógica de programación necesaria

Game maker permite trabajar de dos formas dos formas diferentes:

3. Game Maker modo Drag and Drop

En este modo todas las funcionalidades se crean a partir de la interfaz visual de programa: seleccionado opciones, arrastrando elementos o eligiendo las funcionalidades mediante desplegable.

4. Game Maker modo programación.

Gracias al lenguaje de programación y de creación de scripts propio de GM, denominado GML se puede crear las funcionalidades sin hacer uso de la interfaz visual.

Ambos modos no son excluyentes y pueden convivir dentro de un mismo proyecto, por lo que permite implementar partes algunas partes con programación visual y otras con GML. Por ello el programa se adapta a la percepción a toda clase de usuarios y niveles.

Como se acaba de explicar , GM permite trabajar sin hacer uso de lógica de programación por lo que la puntuación en este apartado es la máxima.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Lógica de programación necesaria	0-4	4

Tabla 149: Puntuación subcriterio lógica de programación necesaria

Resumen puntuación funcionamiento y dificultad de uso

En la tabla siguiente se exponen la puntuación de todos los apartados que componen el criterio funcionamiento y dificultad de uso.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Funcionamiento General	0-4	4
Primeros Pasos: Creación de un pequeño proyecto.	0-4	3
Funciones básicas	0-12	9.25
Lógica de programación necesaria	0-4	4
TOTAL		20.25

Tabla 150: GM puntuación criterio funcionamiento y dificulta de uso

Criterio 4: Editor Gráfico/Visual.

Los datos obtenidos por GM en la encuesta que valora su interfaz son los mostrados en la siguiente tabla.

Concepto	Puntuación Máxima	Puntuación mínima	Media (Sobre 5)
Rapidez al inciarse	4	2	2,875
El programa se muestra rápido	4	3	3,875
La interfaz es atractiva	5	4	4,375
Tiene un aspecto cuidado y moderno	5	4	4,625
Puntuación diseño del programa	5	4	4,5
Los iconos representan bien los elementos Y/o funciones	5	3	4,625
A priori parece sencillo de usar	5	4	3,5
Consumo de recursos	5	4	4,875
MEDIA TOTAL		4.15/5	
PUNTUACIÓN PONDERADA		6.64/8	

Tabla 151: GM resultado encuesta interfaz

Una vez realizada la ponderación de la puntuación para adaptarla a la importancia del criterio la puntuación es la que muestra la siguiente tabla.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Editor Gráfico/Visual	Clave	6.64

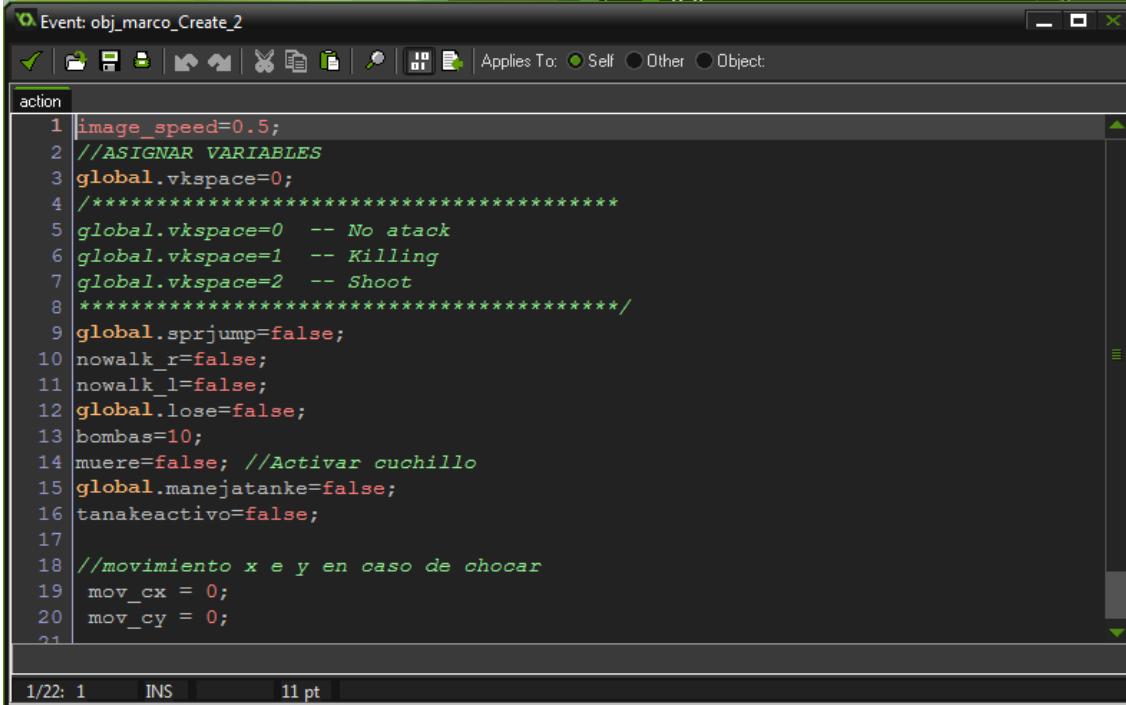
Tabla 152: Criterio: Editor gráfico/visual

Criterio 5: Escalabilidad.

Puntuación subcriterio posibilidad de creación de funciones, scripts y/o lenguaje de programación propio.

Como se ha visto en el apartado de funcionamiento GM tiene dos modos de funcionamiento no excluyentes entre sí: el modo Drag and Drop y el modo

programación. En este modo se trabajará con el lenguaje de programación propio GML, véase Figura 82.



```

Event: obj_marco_Create_2
action
1 image_speed=0.5;
2 //ASIGNAR VARIABLES
3 global.vkspace=0;
4 ****
5 global.vkspace=0 -- No attack
6 global.vkspace=1 -- Killing
7 global.vkspace=2 -- Shoot
8 ****
9 global.sprjump=false;
10 nowalk_r=false;
11 nowalk_l=false;
12 global.loose=false;
13 bombas=10;
14 muere=false; //Activar cuchillo
15 global.manejatanke=false;
16 tanakeactivo=false;
17
18 //movimiento x e y en caso de chocar
19 mov_cx = 0;
20 mov_cy = 0;
21

```

1/22: 1 INS 11 pt

Figura 94: Lenguaje de programación GML

GML es un lenguaje basado en C++ , aunque también se pueden encontrar referencias a Pascal²⁶.

Las principales características de GML son las siguientes:

- La sintaxis es muy similar a la de C, pero no es tan estricta ya que GML es un lenguaje muy flexible ya que cuenta con diversas sintaxis válidas.
- Cuenta con una gran cantidad de funciones predefinidas que se corresponden con las funcionalidades que se pueden implementar vía Drag and Drop.
- Solo tiene dos tipos de variables: Strings, cadenas de caracteres ASCII; y valores reales con signo de punto flotante.
- Tiene tres tipos de variables por su ámbito de aplicación : variables instancia, propias de una clase en particular; variables locales, solo accesibles dentro del

²⁶ lenguaje creado por el profesor suizo Niklaus Wirth entre los años 1968 y 1969 y publicado en 1970, utilizando la programación estructurada y estructuración de datos. Lenguaje de programación fuertemente tipado.

ámbito de un script o función; y las variables globales que son accesibles desde cualquier punto del proyecto.

GML es un lenguaje muy potente que maximiza el tiempo dedicado a la creación de los juegos y permite que los proyectos sean mucho más complejos y completos. Además su sintaxis parecida a C++ , hace que cualquier usuario con conocimientos de este lenguaje se encuentre cómodo y pueda adaptarse a él de manera rápida.

Pero además Game Maker tiene disponibles dentro de la tienda, vista en el punto anterior, algunos complementos que permiten no utilizar GML sino directamente C++, aunque no es recomendable ya que no tendríamos disponibles todas las funciones predefinidas de GML, es una opción más con la que cuentan los usuarios para expandir las funcionalidades base.

Como se ha podido comprobar GM tiene muchas posibilidades para crear funciones y script propios, unido a la potencia del lenguaje GML y a que se tiene la posibilidad de usar otros lenguajes para la creación de estos nuevos scripts la puntuación este apartado es máxima.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Posibilidad de crear funciones y scripts propios y/o lenguaje de programación propio.	0-8	8

Tabla 153: GML 2: Puntuación subcriterio creación de funciones propias.

Puntuación subcriterio Add-ons y plug-ins

Game maker dispone además de la posibilidad de crear funciones propias, de una tienda de plugins y addons todos ellos aprobados por la desarrolladora del programa. Los complementos disponibles se encuentras clasificados en las siguientes categorías:

- **Animation:** Conjuntos de animaciones y recursos gráficos para usar en los juegos que se desarrollen con GM. Todos ellos se pueden utilizar para fines comerciales y están especialmente optimizados para GM.
- **Audio:** Dentro de esta categoría se encuentran recursos sonoros. Se subdivide en música de ambiente, música, efectos y otros.
- **Background:** Conjuntos de tiles para crear elementos para los proyectos, tales como plataformas, paisajes y fondos.

- **Demos:** Proyecto de GM que contienen juegos completos o funcionalidades, que el usuario puede usar a modo de ejemplo o como base de sus proyectos.
- **Extensiones:** Dotan a GM de nuevas características con las que no cuenta de base. Es quizás la categoría más importante de addons disponibles. Entre ellos se encuentran funcionalidades tan importantes como la integración con Facebook, el uso de archivos .avi en los proyectos, control de versiones del juego o un motor de juego en línea.
- **Fonts:** Diferentes fuentes tipográficas para usar en los proyectos.
- **Included files:** Incluyen más tipos de archivos en los soportados por GM de serie.
- **Object:** Objetos de GM con comportamientos predefinidos. Por ejemplo movimientos tipo plataformas, generación aleatoria de romos u movimiento con pantalla táctil.
- **Path:** Asistentes con menú contextual para la creación de sprites y fondos.
- **Project:** Juegos completos con su archivo de proyecto que se puede usar de base para crear juegos.
- **Rooms:** Escenarios prediseñados.
- **Scripts:** Códigos GML que implementan diversas funcionalidades. Dentro esta categoría se encuentran scripts de IA, de cámaras, de efectos, de dispositivos de entrada y salida, de físicas, de red, de animación , de audio y otros tipos no incluidos en ninguna de las anteriores categorías.
- **Servicios:** Colección de servicios externos que se pueden unir a Game Maker. Por ejemplo, añadir el SDK de Android o extensión de HTML5.
- **Shaders:** Extensiones gráficas para los proyectos.
- **Sprites:** Paquetes de sprites de personajes y elementos gráficos coherentes entre si.

En la tabla siguiente se muestran los tipos de plugins, la cantidad de cada uno de ellos disponible en la tienda y su precio máximo y mínimo.

Tipo de extensión	Cantidad total	Precio mínimo	Precio máximo
-------------------	----------------	---------------	---------------

Animation	62	Gratis	9.99€
Audio	150	Gratis	19.99€
Backgrounds	154	Gratis	19.99€
Demos	134	Gratis	14.99€
Extensions	296	Gratis	9.99€
Fonts	19	Gratis	0.99€
Included Files	18	Gratis	19.99€
Paths	32	Gratis	1.99€
Objetct	80	Gratis	9.99€
Project	488	Gratis	999.99€
Rooms	4	Gratis	9.99€
Scripts	552	Gratis	999.99€
Services	28	Gratis	Gratis
Shaders	72	Gratis	29.99€
Sprites	650	Gratis	49.99€

Tabla 154: GM extensiones

Como se puede comprobar en la tabla anterior GM tiene un gran cantidad de addons y plugins de todo tipo. Además, mucho de estos recursos son gratuitos. Por lo que la puntuación en este apartado debe ser máxima.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Add-ons y plug-ins	0-6	6

Tabla 155: GM: Puntuación subcriterio addons y plugins

Resumen puntuación Escalabilidad.,

La tabla siguiente muestra el resumen de la puntuación del criterio escalabilidad.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Posibilidad de crear funciones y scripts propios y/o lenguaje de programación propio.	0-8	8
Add-ons y plug-ins	0-4	6
TOTAL		14

Criterio 6: Resultados profesionales.

A continuación, se expone una lista de juegos comerciales más destacados que han sido desarrollados con Game Maker.

Hotline Miami 1 y 2.

El primero Hotline Miami llegó al mercado en octubre de 2012 ganándose en poco tiempo el favor del público y la crítica especializada. Se trata de un shooter de acción cenital en el que el objetivo del jugador es acabar con todos los enemigos de cada nivel y con el jefe final que aparece una vez derrotamos a todos ellos.

En su aspecto visual recuerda a títulos de acción cenital como Gran Theft Auto²⁷, antes de dar el salto a las 3 dimensiones y clásicos de máquinas arcade de principios de los noventa.

Destaca sobre todo en su ambientación basando sus diseños de niveles, personajes, objetos y banda sonora en los años 80 y en su alta dificultad. Además de su pulido sistema de combate, en que se puede hacer uso de toda clase de armas cuerpo a cuerpo o a distancia, apoyándose en un control casi perfecto.

En 2015 fue lanzada al mercado su segunda parte *Hotline Miami 2: Wrong Number*, imagen en la Figura 97, manteniendo todas las características del primero juego de la saga y siendo recibido por el público de la misma forma que la primera parte.



Figura 95: Hotline Miami

²⁷ (abreviado como GTA) es una serie de videojuegos, con 16 entregas lanzadas entre los años 1997 y 2015.

Genero	Acción
Lanzamiento	Octubre 2012
Nota Metacritic	85/100
Plataformas disponibles	Pc, Linux, Mac, PS3, PS Vita

Tabla 156: Hotline Miami datos

Undertale

Undertale es quizás es juego indie²⁸ más influyente de los últimos años y quizás de toda la historia de este tipo de desarrollos.

Fue nominado e incluso ganador (IGN Best of 2015 PC Gamer of the year ²⁹) a juego del año 2015 en muchas revistas especialidades en videojuego y en premios del sector, cosa que ningún juego indie había conseguido hasta la fecha, luchando por el galardón con juegos como Metal Gear Solid 5, con un coste aproximado de 80 millones de Dólares³⁰, o The Witcher 3, con 81\$ millones de Dolarés invertidos¹⁰. Sorprende más aún ya que el juego ha sido desarrollado prácticamente por una sola persona, Toby Fox, contando únicamente con apoyo de terceros para desarrollar las versiones de PS4 y PSVita.

Undertale, imagen del juego Figura 83, es un RPG con combates por turnos que destaca por su banda sonora, también de Toby Fox, sus diálogos complejos, su elaborada y adulta historia, tratando temas como la sexualidad, la homosexualidad y la violencia. Además, el juego permite al jugador determinar que hacer con los enemigos con los que se encuentra si matarlos, dejarlos vivos o mediante una serie de comandos durante el combate conseguir su amistad, esta forma de actuar por parte del jugador varía la historia y su final.

²⁸ son videojuegos creados por individuos o pequeños grupos, sin apoyo financiero de distribuidores. Los videojuegos independientes a menudo se centran en la innovación y se basan en la distribución digital.

²⁹ http://www.ign.com/wikis/best-of-2015/PC_Game_of_the_Year

^{30,10} https://en.wikipedia.org/wiki/List_of_most_expensive_video_games_to_develop



Figura 96: Undertale

Undertale	
Genero	RPG, Puzzle
Lanzamiento	Septiembre 2015
Nota Metacritic	92/100
Plataformas disponibles	Pc, Linux, Mac, PS4, PS Vita

Tabla 157: Undertale datos

Spelunky

El caso de Spelunky es un caso distinto a la mayoría de juegos mencionados en este proyecto y puede ser muy ejemplarizante de lo que pueden llegar a realizar este tipo de herramientas, sus desarrolladores no tenían al principio ninguna pretensión comercial y lanzaron su juego como freeware ³¹en 2009, pero vendo el éxito del juego y que su idea al principio sencilla había alcanzado una calidad alta en todos sus apartados para ser comercializado, en 2013, después de realizar toda una serie de mejoras relanzaron esta vez de manera comercial.

Spelunky es un juego de exploración y plataformas 2D con generación de niveles procedural, imagen de un nivel en Figura 84, por lo que cada partida es diferente.

³¹ tipo de software que se distribuye sin costo, disponible para su uso, pero que mantiene el copyright

Cuenta con planteamiento clásico ya que no hay ni guardado ni puntos de control ³²por lo que si el jugador muere debe comenzar de nuevo la aventura.



Figura 97: Spelunky

Genero	Acción, Plataformas
Lanzamiento	Diciembre 2015
Nota Metacritic	90/100
Plataformas disponibles	Pc, Xbox, PS3, PS Vita.

Tabla 158: Spelunky datos

Nuclear Throne

Nuclear Throne es un Roguelike³³ con influencia de los arcades de los 80 y 90, lanzado al mercado a finales del año 2015. Cosechando unas buenas críticas de los medios especializados del sector por su ajustada dificultad su preciso control y sus escenarios procedurales.

³² Instante o lugar de un videojuego donde la partida se guarda automáticamente, y desde donde se retoma la acción en caso de que el jugador pierda o sea eliminado.

³³ género orientado a que el jugador se adentre en una mazmorra en la que se crea de manera procedural cada nivel



Figura 98: Nuclear Throne

Datos del juego	
Genero	Acción, Roguelike
Lanzamiento	Septiembre 2013
Nota Metacritic	88/100
Plataformas disponibles	Pc, Mac, PS4

Figura 99: Nuclear Throne datos

Puntuación subcriterio principales juegos comerciales desarrollados.

Como se ha visto anteriormente Game Maker es una herramienta con potencial para crear juegos comerciales y de éxito. Los ejemplos mostrados no son solo juegos a la venta creados con Game Maker, sino que además son casos de éxito dentro del desarrollo de videojuegos indie y valorados como los mejores del género, por lo que la puntuación en este sentido es la máxima.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Principales juegos desarrollados	0-8	8

Tabla 159: GM subcriterio principales juegos desarrollados

Puntuación subcriterio diferenciación

Como se ha visto los juegos de Game Maker pertenecen a diferentes géneros como roguelikes, plataformas, acción e incluso aventuras gráficas, como Gods Will Be Watching Figura XX. En el aspecto visual también se encuentra una gran diferenciación de estilos gráficos y artísticos,



Figura 100: Gods will be watching

Se puede concluir que con Game Maker se pueden crear juegos comerciales de estilos y géneros muy diferentes, por tanto, la puntuación en este apartado es la más alta posible.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN

Diferenciación	0-4	4
-----------------------	-----	---

Tabla 160: GM subcriterio diferenciación

Resumen puntuación resultados profesionales.

En la siguiente tabla se muestra el resumen de la puntuación el criterio resultados profesionales.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Principales juegos desarrollados.	0-8	8
Diferenciación	0-4	4
TOTAL		12

Tabla 161: GM criterio resultados profesionales

Criterio 7: Estado actual.

Game Maker se encuentra actualmente activo y con actualizaciones constante, aunque dese que se puso a la venta Game Maker 2, los desarrolladores están abandonando progresivamente la versión 1, sobre todo a lo relativo a nuevas funcionalidades.

La última versión con nuevas funcionalidades fue lanzada en Mayo de 2017 y la principal característica que añadió fue la total compatibilidad con Game Maker 2, permitiendo que proyecto creados con la primera versión fueran totalmente compatibles con la nueva.

ESTADO	ACTIVO
Última versión	16 de mayo 2017
Frecuencia de actualizaciones	A menudo

Tabla 162: GM estado actual

Resumen puntuación criterio Estado actual

Como se ha visto Game Maker se encuentra en estado activo pero la salida de una nueva versión de programa y la migración a ella que están intentando impulsar los desarrolladores hace que el programa vaya a reducir poco a poco sus actualizaciones. Aunque la versión 2 es totalmente compatible con todo lo explicado en el presente proyecto, ya que se trata fundamentalmente de una gran actualización, la puntuación en este apartado debe ser reducida, la mitad en cada apartado salvo en el de relativo a la fecha desde la última versión ya que data de hace relativamente poco tiempo.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Estado actual	0-2	1.5
Diferenciación	0-1	0.5
Fecha desde la última actualización.	0-1	0.5
TOTAL		2.5

Tabla 163: GM puntuación estado actual

Criterio 8: Sistemas operativos soportados y requerimientos.

Los juegos creados con Game Maker se pueden exportar prácticamente a la totalidad de los sistemas operativos y consolas como se puede comprar en la tabla siguiente.

Sistema/Plataforma	Exportación.
Windows	Si
Linux	Si
IOs	Si
Android	Si
HTML 5	Si
Flash	Si
Windows Store (Mobile y Store escritorio)	Si
Sistemas PlayStation (PS3, PS-VITA, PS4)	SI
Sistemas Xbox (XBOX360, XBOX ONE)	Si
Sistemas Nintendo (3DS, Wii, WiiU)	Si
RESUMEN Y PUNTUACIÓN PONDERADA	
TOTAL SISTEMAS	9 DE 10
PUNTUACIÓN PONDERADA	7.2

Tabla 164: GM sistemas de exportación

Resumen puntuación sistemas operativos soportados y posibilidades de exportación.

Con GM se pueden exportar nuestros proyectos a todas las plataformas posibles, únicamente quedaría por cubrir los sistemas Nintendo, por lo que la puntuación en este apartado es prácticamente la máxima.

Una vez ponderada la puntuación la siguiente tabla muestra el resultado de GM en este criterio.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Posibilidades de exportación: Sistemas y plataformas	0-8	7.2
TOTAL		7.2

Tabla 165: GM criterio posibilidades de exportación

Criterio 9: Licencia.

El modelo de negocio de Game Maker se basa en el uso de licencias y módulos, mientras que otros motores cobran regalías, un porcentaje de la venta de cada juego va destinado a los desarrolladores del motor (Unity, Unreal Engine).

Con la adquisición de una licencia el usuario puede realizar todos los proyectos que quiera y además publicarlos sin ningún tipo de restricción ni pagos adicionales. Esta licencia tiene carácter personal e indefinido.

Game Maker tiene una licencia gratuita y una versión profesional.

Game Maker Studio free versión.

Aparecida en 2014 la versión gratuita de GameMaker supone un intento de abrir el SDK's a más usuarios. Esta versión limitada en cuanto a funcionalidades puede suponer un primer acercamiento al programa con el que poder crear pequeños proyectos y juego para familiarizarnos con el entorno de desarrollo.

Las limitaciones fundamentales de la versión gratuita de GML son las siguientes:

- No permite el uso de plugins, addons ni extensiones.
- Los juegos son únicamente ejecutables dentro del programa, no es posible exportarlos a ejecutables ni a ningún sistema.
- Limitaciones en el número de objetos: 15 objetos, 20 Sprites , 5 Rooms, 10 sonidos , 2 tilesets, 10 Scripts, 5 Paths, 2 extensiones .
- Imposibilidad de crear grupos de texturas y de audio.
- Programación con GML deshabilitada.
- Incompatibilidad con los plugins de la tienda oficial.

Puntuación subcriterio versión gratuito.

Las limitaciones de la versión gratuita de GM impiden un uso más allá de probar de manera poco profunda el funcionamiento del programa. El número de objetos que permite usar es demasiado escaso para incluso proyectos que pueda llevar a cabo un usuario novato en el programa. Por lo que la puntuación en este apartado es muy baja.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Modo gratuito	0-2	0.5

Tabla 166: GM subcriterio modo gratuito

Game Studio profesional.

Con un precio de 99€, la versión profesional elimina todas las restricciones de la versión gratuita y además permite la exportación a Windows, Linux y Mac.

El precio medio de las licencias es de 179.25€ , por lo que la diferencia entre el precio de la licencia más avanzada de GM y este precio medio es de 79.25 €. Según el baremo definido en las consideraciones previas, existe más de 50€ de diferencia por lo que ha GM le corresponde el 100% de la puntuación.

CTF tiene un coste máximo de licencia de 219.99€. La diferencia con el precio medio de las licencias de los programas analizados (179.25€) es de 40.74€. Con el baremo definido antes de comenzar el análisis a CTF le corresponde un 25% de total de la puntuación.

CRITERIO	PUNTUACIÓN MAX-MIN	PUNTUACIÓN
Precio de licencias	0-4	4

Tabla 167: GM puntuación subcriterio precio de licencias

Puntuación de subcriterio precio módulo de exportación,

En la siguiente tabla se muestra el precio de los módulos de exportación disponibles para GM.

Sistema/Plataforma	Precio
HTML 5	149,99€
Android e IO's	399,99€
Windows Store	399,99€
Xbox One (Un año)	799,99€
PS4 (Un año)	799,99€

Tabla 168: GM precio módulos de exportación

Los módulos de exportación que tienen influencia en la puntuación de este criterio son los de HTML 5, Android e Ios, que se muestran en la siguiente tabla, junto al precio medio de los mismos.

Sistema/Plataforma	Precio
HTML 5	149,99€
Android e IO's	399,99€
PRECIO MEDIO TOTAL	183.32€

Tabla 169: GM precio medio modulos con influencia en la puntuación

El precio medio de los módulos de exportación con influencia el criterio de precio de licencias es de 183.32€, y su diferencia con el precio medio (39.9€) es de 143€ por lo que la puntuación según los baremos definidos es 0.

Resumen puntuación licencias.

CRITERIO	IMPORTANCIA	PUNTUACIÓN MAX-MIN
Precio de Licencias	0-4	4
Modo de prueba o free	0-2	0.5
Precio módulos de exportación	0-2	0
TOTAL		4.5

Tabla 170: GM puntuación licencias

Valoración final y resumen de puntuación

La puntuación final de Game Maker queda como muestra la siguiente tabla. Estos datos se analizarán más en profundidad en el apartado análisis de resultados del proyecto.

CRITERIO	PUNTUACIÓN MÁXIMA	PUNTUACIÓN	Porcentual por apartados
Requisitos e instalación	6	5,16	86,00%
Documentación, tutoriales y comunidad.	16	15	93,75%
Funcionamiento y dificultad de uso	24	20,25	84,38%
Editor Gráfico/Visual	8	6,64	83,00%
Escalabilidad	14	14	100,00%
Resultados profesionales	12	12	100,00%
Estado actual	4	2,5	62,50%
Sistemas operativos soportados y requerimientos.	8	7,2	90,00%
Licencias	8	4,5	56,25%
TOTAL	100	87,25	

Tabla 171: GM Puntuación final

5.3. Programación en Game Maker vs Programación C++ SFML.

5.3.1. Razón de la comparación.

Una vez se ha seleccionado un framework entre los analizados gracias a los criterios definidos previamente hay que compararlo con la programación de un juego en C++.

El objeto de este estudio comparado es valorar si resulta eficiente el uso de estas herramientas frente a las herramientas clásicas de programación de videojuegos usando un lenguaje de programación como ser C o Java.

Se va a valorar el tiempo de desarrollado de ambos juegos y el consumo de recursos de cada uno de ellos.

5.3.2. Requisitos juego a desarrollar.

El juego que se va crear en C++ y Game Maker es un clon de clásico Space Invaders³⁴, véase Figura 101, de nombre Space Invasors.



Figura 101: Space Invaders

Space Invasors tendrá los siguientes requisitos de funcionamiento:

- Al igual que en el original, el jugador encarnará a una nave que tiene que evitar que los enemigos detruyan tanto a las estructuras defensivas que están por encima del jugador como al propio jugador.
- El grupo de enemigos se encontrará en la parte superior de la pantalla y se moverá en conjunto de izquierda a derecha y hacia abajo cuando toque una de los laterales de la pantalla.

³⁴ Videojuego arcade diseñado por Toshihiro Nishikado y lanzado al mercado en 1978.

- El jugador estará situado en la parte inferior de la pantalla, podrá disparar balas a los enemigos con los que intentará destruirles y se moverá pulsando las telcas direccionales de teclado de izquierda a derecha.
- Los enemigos a su vez lanzarán balas de manera aleatoria hacia el jugador.
- Encima del jugador habrá cuatro estructuras, llamadas *builds* o edificios, que por un lado, impedirán que el jugador dispare por debajo de ellas y por otro recibirán los impactos de los enemigos.
- El jugador y cada enemigo tendrá 3 vidas y las estructuras 4. Cada vez que estos objetos vean reducidas sus vidas el Sprite debe cambiarse para mostrar el estado en que encuentran.
- El juego podrá estar en cuatro estados, 1: el menú principal donde el usuario debe pulsar la barra espaciadora para comenzar el juego; el juego en si mismo; la pantalla de muerte; y el estado partida ganada cuando el jugador haya acabado con todos los enemigos.
- Ambos juegos usarán los mismos Sprites, fondos y resolución de pantalla.

5.3.3. Desarrollo juego SFML.

Antes de desarrollar el juego en SFML hay que instalar SFML en el equipo. La explicación de la instalación de SFML se encuentra en el Anexo II: Instalación de SFML en Visual Studio.

Estructura general del juego.

En el siguiente diagrama se muestra la estructura general del juego, con una Interfaz de la que heredan cuatro clases de juego: Hive Enemy, Enemy Ship, Player y Build; la clase Game y la clase bala. A esto se añade un archivo Config y una clase Colision.

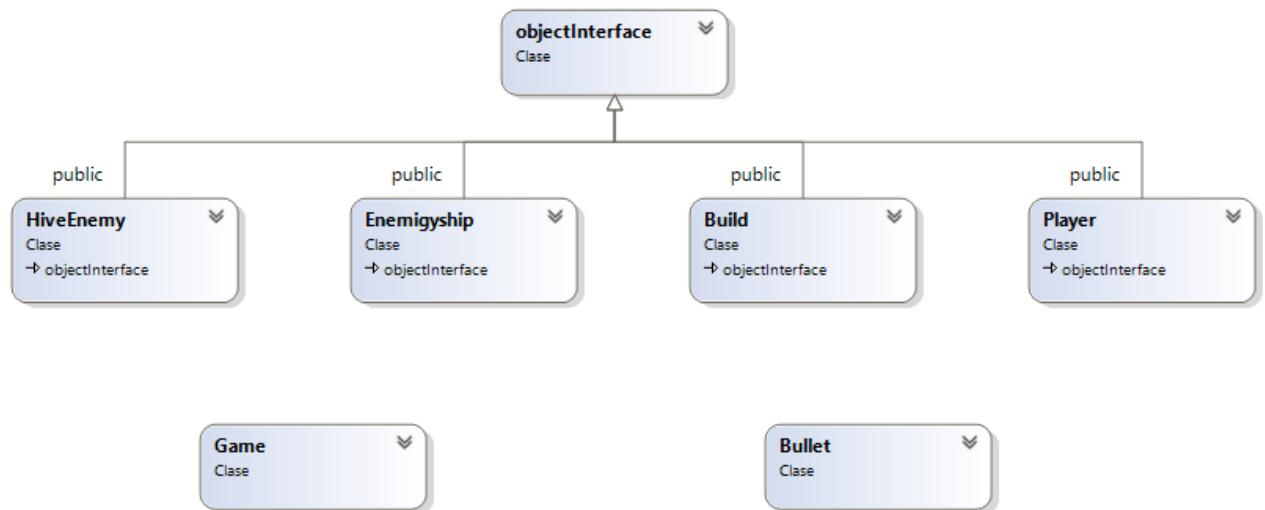


Figura 102: Diagrama general juego C++

A continuación, se explica la funcionalidad, las variables y método de cada una de las clases y archivos.

Archivo Config

El archivo Config contiene las constantes que son accesibles desde todo el programa, véase Figura 103.

```

1 #pragma once
2
3 namespace constants
4 {
5     const int GLOBAL_SCREEN_WIDTH = 1200;
6     const int GLOBAL_SCREEN_HEIGHT = 800;
7     const int GLOBAL_PLAYER_VELOCITY = 200;
8     const int GLOBAL_ENEMY_VELOCITY = 150;
9     const int GLOBAL_ENEMY_V_VELOCITY = 150;
10    const int GLOBAL_BULLET_VELOCITY = 150;
11
12 } // namespace constants
13
14
  
```

Figura 103: Constantes de config

- **GLOBAL_SCREEN_WIDTH:** Define el ancho de pantalla.
- **GLOBAL_SCREEN_HEIGHT:** Define el largo de la ventana de juego.

- **GLOBAL_PLAYER_VELOCITY:** Define la velocidad de movimiento del jugador.
- **GLOBAL_ENEMY_VELOCITY:** Velocidad de movimiento horizontal de los enemigos.
- **GLOBAL_ENEMY_V_VELOCITY:** Indica la velocidad de movimiento vertical del enemigo.
- **GLOBAL_BULLET_VELOCITY:** Velocidad de las balas del juego.

Clase Game.

Es la clase que controla el bucle general del juego y controla los distintos estados en que puede estar (Menú principal, juego, muerte y victoria). Se encarga también de pasar el tiempo delta a los objetos para que el juego funcione a la misma velocidad en cualquier equipo. El siguiente diagrama contiene las variables y métodos de la clase Game.

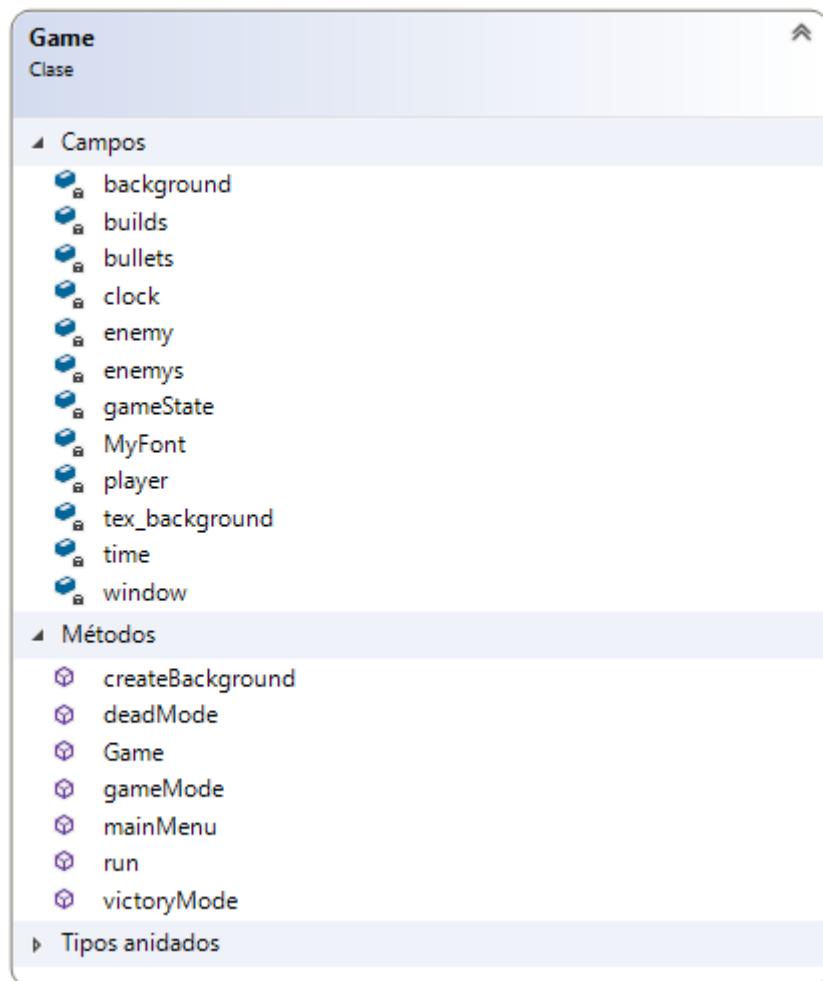


Figura 104: Diagrama Clase Game

Las variables de la clase Game como muestra la figura 104 son las siguientes:

VARIABLE	TIPO	FUNCIÓN
window	SFML Window	Ventana del juego.
clock	SFML Clock	Medir el tiempo
time	Time	Calcular el tiempo
tex_background	SFML Textura	Textura del fondo de pantalla
*player	Player	Jugador.
bullets	Bulletlist	Lista que contiene las balas en el juego.
*enemys	HiveEnemy	Conjunto de enemigos.
MyFont	SFML fuente	Fuente del texto que se imprime en pantalla.
gameState	Int	Variable que indica en qué estado se encuentra el juego: Menú, Juego, Muerte o Victoria.
Background	SFML Sprite	Sprite del fondo.

Tabla 172: Variables clase Game

Las tablas siguientes muestran los métodos de la clase Game su función y sus variables de entrada y salida.

Game	
Variables de entrada	<ul style="list-style-type: none"> • Unsigned int width: Ancho de pantalla. • Unsigned int height: Largo de pantalla. • bool verticalSync: Sincronización vertical si o no. • string fondo: Dirección del fondo de pantalla.
Variables de salida	Ninguna.

Función	Constructor del juego.
Funcionamiento	<ul style="list-style-type: none"> • Crear la pantalla con las variables de entrada, y activa o no la sincronización vertical. • Crear el background con el fondo pasado por variable. • Crear el jugador y el enjambre de enemigos (HiveEnemy). • Crear los edificios mediante un bucle for.

Tabla 173: Método game de la clase Game

run	
Variables de entrada	Ninguna.
Variables de salida	Ninguna.
Función	Lanzador del juego.
Funcionamiento	Mientras la ventana esta abierta en un bucle comprueba el estado de la variable gameState y dependiendo del valor de esta ejecuta: mainMain, gameMode, deadMode o victoryMode.

Tabla 174:: Método run de la clase Game

createBackground	
Variables de entrada	<ul style="list-style-type: none"> • string resource: Dirección del fondo de juego.
Variables de salida	Ninguna.
Función	Crear el fondo del juego.
Funcionamiento	Carga desde la ruta indicada en la variable resource el fondo.

Tabla 175: Método createBackground de la clase Game

mainMenu	
Variables de entrada	Ninguna
Variables de salida	Ninguna.
Función	Se encarga de controla el juego en el estado menú.
Funcionamiento	<ul style="list-style-type: none"> • Carga la fuente desde un archivo. • Crea los dos textos que van a mostrarse en pantalla. • Comprueba si se ha pulsado la tecla espacio, si se ha pulsado cambia el estado a 2 (estado de juego principal). • Pinta en pantalla los textos y el fondo.

Tabla 176: Método mainMenu de la clase Game

gameMode	
Variables de entrada	Ninguna
Variables de salida	Ninguna.
Función	Se encarga de controla el juego en el estado juego.
Funcionamiento	<ul style="list-style-type: none"> • Calcular el tiempo delta para pasarlo a los métodos de los objetos. • Recorrer la lista de balas para comprobar si están vivas, la que no lo este la eliminará. • Mismo proceso con la lista de edificios. • Llamar a los métodos update de jugador y de él ejambre de enemigos, de las balas y de los edificios. • Llamar a los métodos de pintar (draw) del jugador, los edificios, las balas y el ejambre de enemigos.

Tabla 177: Método gameMode de la clase Game

deadMode	
Variables de entrada	Ninguna
Variables de salida	Ninguna.
Función	Se encarga de controla el juego en el estado muerte.
Funcionamiento	Funcionamiento igual que el método MainMenu pero con otro texto distinto.

Tabla 178: Método deadMode de la clase Game

victoryMode	
Variables de entrada	Ninguna
Variables de salida	Ninguna.
Función	Se encarga de controla el juego en el estado victoria.
Funcionamiento	Funcionamiento idéntico al de MainMenu y deadMode pero con otro texto.

Tabla 179: Método victoryMode de la clase Game

Clase Player.

La clase Player o jugador es la encargada de gestionar todas las acciones del jugador, desde su movimiento pasando por el disparo y la gestión de su vida.

En el siguiente diagrama se muestran las variables y funciones de la clase player.

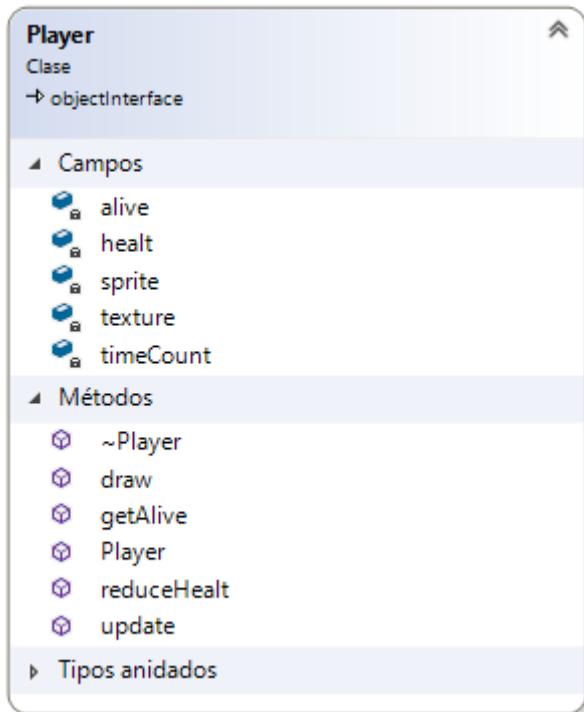


Figura 105: Diagrama Clase Player

La tabla siguiente detalla las variables de la clase Player.

VARIABLE	TIPO	FUNCIÓN
sprite	SFML Sprite	Sprite del jugador que se va pintar en pantalla.
texture	SFML Texture	Textura (imagen) que va llevar el Sprite del jugador.
alive	Bool	Booleano que indica si el jugador esta vivo o no.
healt	Int	Cantidad de vida que tiene el jugador.
timeCount	Float	Variable usada para la dar al disparo de balas un intervalo de tiempo.

Tabla 180: Variables clase Player

Los métodos de la clase player se describen en las tablas siguientes.

Player	
Variables de entrada	<ul style="list-style-type: none"> • int posX: Posición horizontal del jugador. • int posY: Posición en el eje Y del jugador. • string resource: Dirección de la imagen que se va usar como Sprite.
Variables de salida	Ninguna.
Función	Constructor del jugador.
Funcionamiento	<ul style="list-style-type: none"> • Carga la imagen de la ruta pasada en la variable resource como textura y esta textura se la asigna al Sprite. • Escala el Sprite. • Posiciona el Sprite en la posición indicada por las variables posX y posY de entrada. • Asigna 3 de vida y a la variable alive se le da el valor true.

Tabla 181: Método player de la clase Player

Update	
Variables de entrada	<ul style="list-style-type: none"> • float delta: Variable de tiempo para la interpolación del movimiento. • list<Bullets*> bullets: lista de balas en juego.
Variables de salida	<ul style="list-style-type: none"> • list<Bullets*> bullets: devuelve la lista de balas actualizada.
Función	Actualizar el jugador en cada ciclo de ejecución, así como evaluar los posibles eventos de pulsación de teclas.

Funcionamiento	<ul style="list-style-type: none"> • Primero aumenta la variable TimeCount. • Comprueba si se ha pulsado la tecla izquierda o derecha, si ha sido así y no supera los límites de la pantalla mueve el Sprite en la dirección correspondiente con la velocidad indicada en el archivo config.h multiplicado por el tiempo delta. • Si se ha pulsado la tecla espacio y el timeCount es mayor de 1 (intervalo de tiempo entre bala y bala) se creará una nueva bala y se introducirá en la lista de balas. Además, se volverá a poner la variable timeCount a cero. • Se recorre la lista de balas y se comprueba, llamando a un método estático de la clase Colision, si alguna de ellas está colisionando con el jugador se llama al método reduceHealt y al estado alive de esa bala se la asigna false. • Devuelve la lista de balas actualizada.
-----------------------	--

Tabla 182 :Método update de la clase Player

Draw	
Variables de entrada	<ul style="list-style-type: none"> • SFML RenderWindow window: Ventana actual del juego.
Variables de salida	Ninguna.
Función	Pintar el jugador en pantalla.
Funcionamiento	Pinta el Sprite del jugador en la ventana pasada por parámetro.

Tabla 183: Método draw de la clase Player

ReduceHealt	
Variables de entrada	Ninguna.
Variables de salida	<ul style="list-style-type: none"> • Bool alive: Devuelve el estado vivo o no del jugador.
Función	Reducir la vida del jugadot y asignar el nuevo Sprite.
Funcionamiento	<ul style="list-style-type: none"> • Reduce en una unidad la vida. • Dependiendo del valor de la vida cambia el Sprite y devuelve el estado vivo o no.

Tabla 184: Método reduceHealt de la clase Player

getAlive	
Variables de entrada	Ninguna
Variables de salida	<ul style="list-style-type: none"> • bool alive: Variable que indica si el jugador esta en estado vivo o no.
Función	Devolver la variable vivo del jugador.
Funcionamiento	<ul style="list-style-type: none"> • Hace un return de la variable de clase alive.

Tabla 185: Método getAlive de la clase Player

Clase HiveEnemy

La clase HiveEnemy o ejambre en enemigos se encarga de controlar el movimiento conjunto de los enemigos y de actualizarlos y pintarlos en pantalla. La clase Game no interactua directamente con los enemigos, si no que lo hace con HiveEnemy y este lo hace con cada uno de ellos. La estructura de la clase HiveEnemy viene definida por el siguiente diagrama.

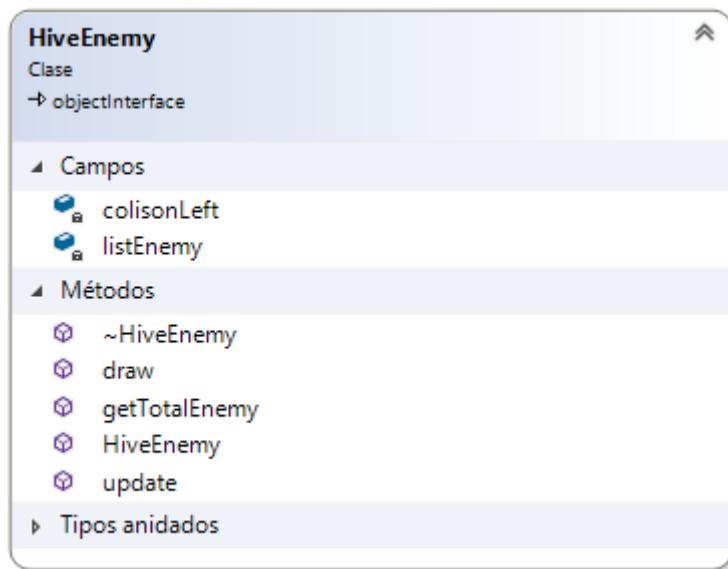


Figura 106: Diagrama Hive Enemy

Visto el diagrama las variables de la clase HiveEnemy son dos:

VARIABLE	TIPO	FUNCTION
colisionLeft	Bool	Indica si alguno de los enemigos está colisionando con los límites de la pantalla por la izquierda.
listEnemy	Enemy List	Lista de enemigos en el juego.

Tabla 186: Variables clase HiveEnemy

Las funciones de la clase HiveEnemy son los descritos en las siguientes tablas.

HiveEnemy	
Variables de entrada	Ninguna
Variables de salida	Ninguna.
Función	Constructor del ejambre de enemigos.

Funcionamiento	<ul style="list-style-type: none"> Asigna a la variables colisionLeft el valor true, para que el movimiento horizontal comience a la derecha. Mediante un bucle doble <i>for</i> crear tres filas de enemigos con 6 enemigos en cada una.
-----------------------	---

Tabla 187: Método HiveEnemy de la clase HiveEnemy

update	
Variables de entrada	<ul style="list-style-type: none"> float delta: Variable de tiempo para la interpolación del movimiento. list<Bullets*> bullets: lista de balas en juego.
Variables de salida	<ul style="list-style-type: none"> list<Bullets*> bullets: devuelve la lista de balas actualizada.
Función	Actualiza cada uno de los enemigos que componen el ejambre de enemigos.
Funcionamiento	<ul style="list-style-type: none"> Recorre la lista de enemigos para comprobar si alguno esta colisionando con los límites de la pantalla y así controlar el cambio de dirección. En función del resultado anterior se mueven los enemigos en una dirección u en otra con la velocidad que marque la constante de Config.h y el tiempo delta, teniendo en cuenta que cuando lleguen al límite izquierdo también se moverán hacia abajo. Se recorre la lista de enemigos y se llama al método update de cada uno de ellos con el tiempo delta y la lista de balas. Se comprueba si alguno de los enemigos de la lista está muerto si es así lo eliminamos de la lista y lo destruimos.



Tabla 188 :Método update de la clase Hive Enemy

draw	
Variables de entrada	<ul style="list-style-type: none"> • SFML RenderWindow window: Ventana actual del juego.
Variables de salida	Ninguna.
Función	Pintar a los enemigos en pantalla.
Funcionamiento	Recorre la lista de enemigos y llama al método pintar de cada uno de ellos pasándoles como parámetro la variable window.

Tabla 189: Método draw de la clase HiveEnemy

getTotalEnemy	
Variables de entrada	Ninguna
Variables de salida	<ul style="list-style-type: none"> • Int totalEnemy: Número de enemigos en el enjambre.
Función	Devolver el número total de enemigos en la lista.
Funcionamiento	<ul style="list-style-type: none"> • Mediante un size de la lista de enemigo se obtiene el número total de enemigos y lo devuelve.

Tabla 190: Método getTotalEnemy

Clase EnemygyShip

La clase EnemygyShip o enemigo se encarga del comportamiento de los enemigos individualmente. En el siguiente diagrama se muestran las variables y funciones de la clase.

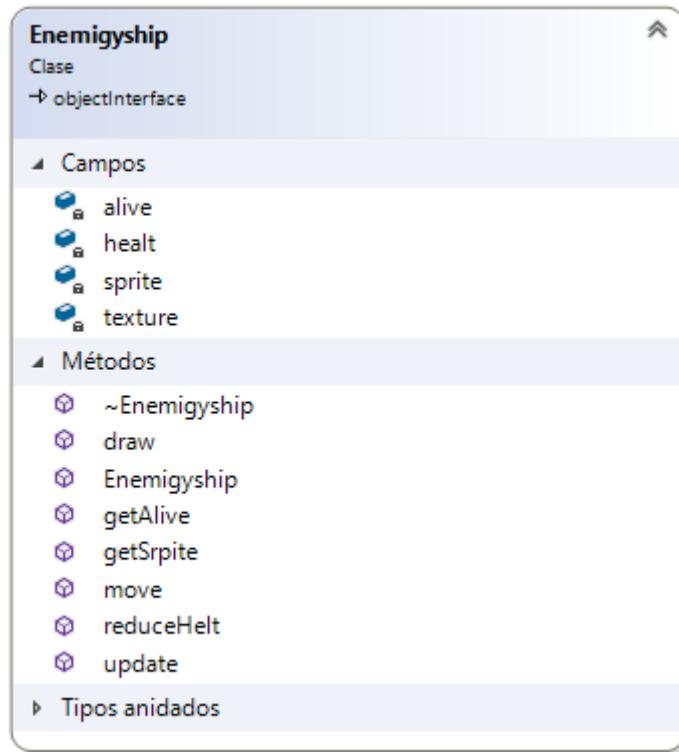


Figura 107: Diagrama EnemigyShip

La tabla siguiente describe el tipo y la función de las variables de la clase enemigo.

VARIABLE	TIPO	FUNCIÓN
sprite	SFML Sprite	Sprite del enemigo que se va pintar en pantalla.
texture	SFML Texture	Textura (imagen) que va llevar el Sprite del enemigo.
alive	Bool	Booleano que indica si el enemigo esta vivo o no.
healt	Int	Cantidad de vida que tiene el enemigo.

Tabla 191: Variables clase EnemigyShip

En cuanto a sus funciones vienen descritas por las siguientes tablas.

EnemygyShip	
Variables de entrada	<ul style="list-style-type: none"> • int posX: Posición horizontal del jugador. • int posY: Posición en el eje Y del jugador. • string resource: Dirección de la imagen que se va usar como Sprite.
Variables de salida	Ninguna.
Función	Constructor del enemigo
Funcionamiento	<ul style="list-style-type: none"> Carga la textura desde el archivo resource y la asigna al Srpite. Posiciona el enemigo en la pantalla en función de las variables de entrada PosX y PosY. Da a la variable alive el valor true y asigna 3 puntos de vida.

Tabla 192: Método *EnemygyShip* de la clase *EnemygyShip*

update	
Variables de entrada	<ul style="list-style-type: none"> • float delta: Variable de tiempo para la interpolación del movimiento. • list<Bullets*> bullets: lista de balas en juego.
Variables de salida	• list<Bullets*> bullets: devuelve la lista de balas actualizada.
Función	Actualiza cada uno de los enemigos que componen el ejambre de enemigos.
Funcionamiento	<ul style="list-style-type: none"> Se recorre la lista de balas y se comprueba, llamando a un método estático de la clase

	<p>Colision, si alguna de ellas está colisionando con el enemigo se llama al método reduceHealth y al estado alive de esa bala se la asigna false.</p> <ul style="list-style-type: none"> • Se genera un número aleatorio entre 1000 si el valor es cero el enemigo disparará y, por tanto, se creará una nueva bala en dirección al jugador, esta bala se añadirá a la lista de balas recibida como parámetro. • Se devuelve la lista de balas.
--	--

Tabla 193 :Método update de la clase EnemygyShip

draw	
Variables de entrada	<ul style="list-style-type: none"> • SFML RenderWindow window: Ventana actual del juego.
Variables de salida	Ninguna.
Función	Pintar al enemigo en pantalla.
Funcionamiento	Se pinta el enemigo en la window recibida como parámetro.

Tabla 194: Método draw de la clase EnemygyShip

move	
Variables de entrada	<ul style="list-style-type: none"> • float delta: Variable de tiempo para la interpolación del movimiento. • int direction: Dirección en la que se va a mover el enemigo.
Variables de salida	Ninguna.
Función	Mover al enemigo

Funcionamiento	En función de la variable direction recibida el enemigo se moverá en horizontal , hacia la izquierda o derecha o hacia abajo, con la velocidad que le marque la variable de config.h multiplicada por el tiempo delta.
-----------------------	--

Tabla 195: Método move de la clase EnemygyShip

reduceHealt	
Variables de entrada	Ninguna.
Variables de salida	<ul style="list-style-type: none"> Bool alive: Devuelve el estado vivo o no del jugador.
Función	Reducir la vida del jugadot y asignar el nuevo Sprite.
Funcionamiento	<ul style="list-style-type: none"> Reduce en una unidad la vida. Dependiendo del valor de la vida cambia el Sprite y devuelve el estado vivo o no.

Tabla 196: Método reduceHealt de la clase EnemygyShip

getAlive	
Variables de entrada	Ninguna
Variables de salida	<ul style="list-style-type: none"> bool alive: Variable que indica si el enemigo esta en estado vivo o no.
Función	Devolver la variable vivo del enemigo.
Funcionamiento	<ul style="list-style-type: none"> Hace un return de la variable de clase alive.

Tabla 197: Método getAlive de la clase EnemygyShip

getSprite	
Variables de entrada	Ninguna
Variables de salida	<ul style="list-style-type: none"> SFML Sprite: Sprite del enemigo.
Función	Devolver el Sprite del enemigo.
Funcionamiento	<ul style="list-style-type: none"> Hace un return de la variable Sprite de la clase.

Tabla 198: Método getSprite de la clase EnemygyShip

Clase Bullet

La clase bullet tiene el comportamiento de la bala en el juego. Puede ser disparada por tanto por los enemigos como por el jugador e ir en dirección vertical positiva y vertical negativa. Su estructura viene definida por el siguiente diagrama.

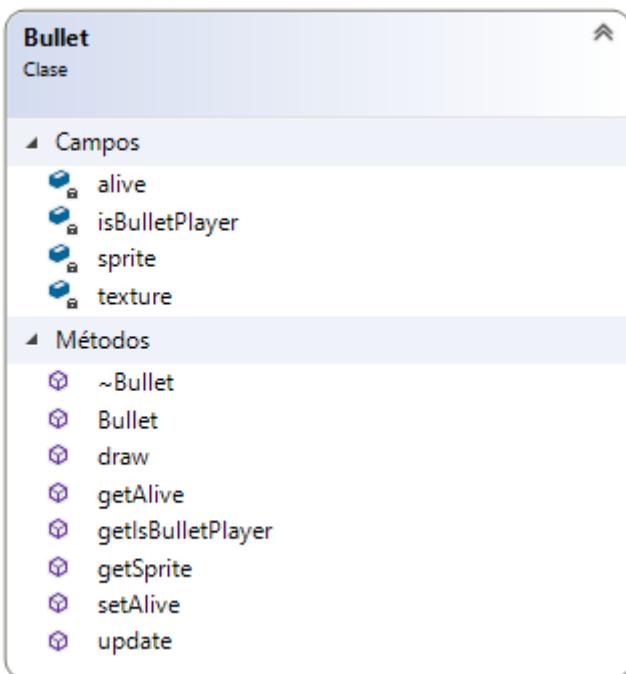


Figura 108: Diagrama clase bullet

Como se ha visto en el diagrama las variables de la clase bullet o bala son las siguientes:

VARIABLE	TIPO	FUNCIÓN
sprite	SFML Sprite	Sprite de la bala que se va pintar en pantalla.
texture	SFML Texture	Textura (imagen) que va llevar el Sprite de la bala.
alive	Bool	Booleano que indica si la bala esta viva o no.
isBulletPlayer	Bool	Booleano que indica si la bala es del jugador o no.

Tabla 199: Variables clase Bullet

En cuanto a los métodos vienen descritos en las siguientes tablas.

Bullet	
Variables de entrada	<ul style="list-style-type: none"> • int posX: Posición horizontal de la bala. • int posY: Posición en el eje Y de la bala. • bool BulletPlayer: Valor que indica si la bala es del jugador o no.
Variables de salida	Ninguna.
Función	Constructor de la bala
Funcionamiento	<ul style="list-style-type: none"> Carga la textura desde el archivo resource y la asigna al Srpite. Asigna el valor de la variable recibida BulletPlayer a la variable de clase isBulletPlayer. Posiciona la bala en la pantalla en función de las varaiables de entrada PosX y PosY. Si la variable no es del jugador la rota 180 grados.

Tabla 200: Método *HiveEnemy* de la clase *Bullet*

update	
Variables de entrada	<ul style="list-style-type: none"> • float delta: Variable de tiempo para la interpolación del movimiento.
Variables de salida	Ninguna
Función	Actualiza la bala en cada ciclo del reloj, la mueva y detecta si esta fuera o no de los límites de la pantalla.
Funcionamiento	<ul style="list-style-type: none"> • Detecta si la bala está fuera de los límites de la pantalla, tanto superior como inferior, si es así pone su variable alive a false. • En función de si la bala es del enemigo o del jugador la mueve hacia arriba o hacia abajo, con la velocidad que indica la constante del archivo config.h por el tiempo delta.

Tabla 201 :Método update de la clase Bullet

draw	
Variables de entrada	<ul style="list-style-type: none"> • SFML RenderWindow window: Ventana actual del juego.
Variables de salida	Ninguna.
Función	Pintar la bala en pantalla.
Funcionamiento	Se pinta el Sprite de la bala en la window recibida como parámetro.

Tabla 202: Método draw de la clase Bullet

getSprite	
Variables de entrada	Ninguna
Variables de salida	<ul style="list-style-type: none"> SFML Sprite: Sprite de la bala.
Función	Devolver el Sprite de la bala.
Funcionamiento	<ul style="list-style-type: none"> Hace un return de la variable Sprite de la clase.

Tabla 203: Método getSprite de la clase Bullet

setAlive	
Variables de entrada	<ul style="list-style-type: none"> bool alive: Variable que indica si la bala esta en estado viva o no.
Variables de salida	Ninguna
Función	Cambiar el estado alive de la bala
Funcionamiento	<ul style="list-style-type: none"> Asigna el valor recibido a la variable de clase alive.

Tabla 204: Método getAlive de la clase Bullet

getAlive	
Variables de entrada	bool alive: Variable que indica si la bala esta en estado viva o no.
Variables de salida	<ul style="list-style-type: none"> Ninguna
Función	Devolver la variable alive de la bala.
Funcionamiento	<ul style="list-style-type: none"> Hace un return de la variable de clase alive.

Tabla 205: Método getAlive de la clase Bullet

getIsBulletPlayer	
Variables de entrada	Ninguna
Variables de salida	<ul style="list-style-type: none"> bool isBulletPlayer: Variable que indica si la bala es del jugador o no.
Función	Devolver la variable isBulletPlayer de la clase.
Funcionamiento	<ul style="list-style-type: none"> Hace un return de la variable isBulletPlayer de clase.

Tabla 206: Método *getAlive* de la clase Bullet

Clase Build

La clase Build es la encargada de dar comportamiento los edificios que se encuentran entre los enemigos y el jugador. Estos edificios como se ha visto en los requisitos tienen como principal función servir como escudos con tres vidas a los disparos de los enemigos. El siguiente diagrama muestra su estructura.

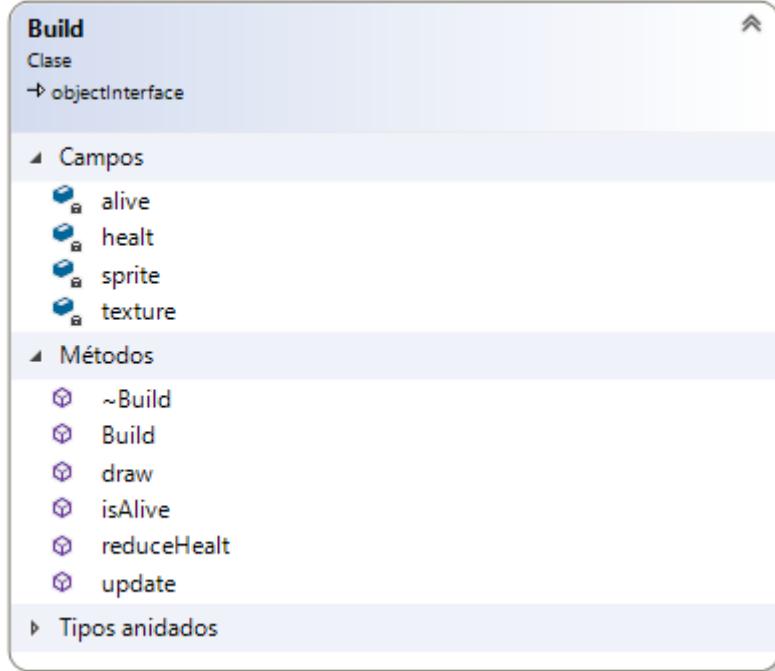


Figura 109: Diagrama de la clase Build

A continuación, en la tabla, se explican las variables de la clase Build.

VARIABLE	TIPO	FUNCIÓN
sprite	SFML Sprite	Sprite del edificio que se va a pintar en pantalla.
texture	SFML Texture	Textura (imagen) que va a llevar el Sprite del edificio.
alive	Bool	Booleano que indica si el edificio está vivo o no.
healt	Int	Vida del edificio.

Tabla 207: Variables clase Build

En las tablas siguientes se explican los métodos visto en el diagrama de la clase Build

Build	
Variables de entrada	<ul style="list-style-type: none"> • int posX: Posición horizontal del edificio. • int posY: Posición en el eje Y del edificio. • string resource: Ruta de la imagen que se va a usar como Sprite.
Variables de salida	Ninguna.
Función	Constructor del edificio
Funcionamiento	<ul style="list-style-type: none"> Carga la textura desde el archivo resource y la asigna al Sprite. Asigna el valor true a la variable alive y 4 a la variable healt.

Tabla 208: Método HiveEnemy de la clase Build

update	
Variables de entrada	<ul style="list-style-type: none"> • float delta: Variable de tiempo para la interpolación del movimiento. • list<Bullets*> bullets: lista de balas en juego.
Variables de salida	<ul style="list-style-type: none"> • list<Bullets*> bullets: devuelve la lista de balas actualizada.
Función	Actualiza el estado del edificio.
Funcionamiento	<ul style="list-style-type: none"> • Se recorre la lista de balas y se comprueba, llamando a un método estático de la clase Colision, si alguna de ellas está colisionando con el edificio y no es una bala del jugador se llama al método reduceHealt y al estado alive de esa bala se la asigna false. • En el caso de que exista colisión y la bala sea del jugador únicamente se destruye la bala. • Se devuelve la lista de balas.

Tabla 209 :Método update de la clase Build

draw	
Variables de entrada	<ul style="list-style-type: none"> • SFML RenderWindow window: Ventana actual del juego.
Variables de salida	Ninguna.
Función	Pintar al edificio en pantalla.
Funcionamiento	Se pinta el edificio en la window recibida como parámetro.

Tabla 210: Método draw de la clase Build

reduceHealt	
Variables de entrada	Ninguna.
Variables de salida	<ul style="list-style-type: none"> Bool alive: Devuelve el estado vivo o no del edificio
Función	Reducir la vida del edificio y asignar el nuevo Sprite.
Funcionamiento	<ul style="list-style-type: none"> Reduce en una unidad la vida del edificio. Dependiendo del valor de la vida cambia el Sprite y devuelve el estado vivo o no.

Tabla 211: Método reduceHealt de la clase Build

isAlive	
Variables de entrada	bool alive: Variable que indica si el edificio esta en estado viva o no.
Variables de salida	<ul style="list-style-type: none"> Niguna
Función	Devolver la variable alive del <i>build</i> .
Funcionamiento	<ul style="list-style-type: none"> Hace un return de la variable alive de clase.

Tabla 212: Método getAlive de la clase Build

Clase Colision

Para la implementación de Space Invasors se ha utilizado la librería de Colisión de SFML³⁵. Esta clase cuenta con varios métodos estáticos que detectan la colisión entre objetos.

En el programa creado se hace uso de la función *PixelPerfectTest*, que recibe dos Sprites y devuelve true o false dependiendo si existe colisión o no.

Estructura de datos especiales usadas: Tipo listas.

Para una mejor gestión de memoria se han creado listas personalizadas para la gestión de los objetos del juego. A continuación, se explica como se crean y se usan las listas tipo bullet, relativas a las balas y para el resto de objetos el proceso es el mismo.

³⁵ <https://github.com/SFML/SFML/wiki/Source:-Simple-Collision-Detection-for-SFML-2>

1. Include necesarios.

```
#include <list>
```

Figura 110: Include necesario para listas

2. Definición de listas personalizadas y declaración de variables.

```
45
46     //Definimos las listas de balas
47     typedef std::list<Bullet*> bulletsList;
48     //DEFINIMOS LOS ITERADORES
49     typedef std::list<Bullet*>::iterator BulletIndex;
50
51     //Creamos las variables
52     bulletsList bullets;
53     BulletIndex beginBullet, endBullet;
```

Figura 111: Listas: Definición y declaración

3. Recorrer y hacer uso de los métodos de la bala.

```
192 }
193
194 //RECORRER ITERADOR DE BALAS
195
196 //Primer elemento de la lista
197 firstBullet = bullets.begin();
198
199 //Ultimo elemento de la lista
200 lastBullet = bullets.end();
201
202 //Mientras el primer y el ultimo elemento sean distintos bucle
203 while (firstBullet != lastBullet)
204 {
205     //Si queremos operar con el objeto debemos hacer asignarlo a un objeto tipo bullet
206     Bullet* bullet = (*firstBullet);
207     //Acción del objeto
208     bullet->update(deltaTime);
209     //Pasamos a la siguiente bala
210     firstBullet++;
211 }
212 }
```

Figura 112: Recorrer lista de balas y usar método

4. Eliminar balas en la lista (En este caso es importante realizar las acciones en el orden indicado sino fallará la ejecución).

```

135 //borramos bala's maestras
136 //RECORRER ITERADOR DE BALAS
137
138 //Primer elemento de la lista
139 firstBullet = bullets.begin();
140 //Ultimo elemento de la lista
141 lastBullet = bullets.end();
142
143 //Bucle mientras sean distintos
144 while (firstBullet != lastBullet)
{
145     // Asignamos a bala.
146     Bullet* bullet = (*firstBullet);
147
148     //Pasamos al siguiente elemento
149     firstBullet++;
150
151     //Lo sacamos de la lista
152     bullets.remove(bullet);
153     //Eliminamos la bala
154     delete bullet;
155
156 }
157

```

Figura 113: Eliminar bala de la lista

Resumen de juego en C++ y SFML.

En la siguiente tabla se resume los datos del juego realizado con C++.

Tiempo invertido	Consumo medio cpu	Consumo medio Ram	FPS
18h 45m	2%	74,30Mb	60

Tabla 213: Resumen juego C++

Los archivos fuentes del juego se pueden consultar en el github de Óscar Sanchez³⁶.

³⁶ https://github.com/oscarvha/SFML_space_invaders

5.3.4. Desarrollo juego Game Maker.

A continuación, se va a explicar las principales características y objetos del juego realizado en Game Maker.

La estructura del juego va estar basada en *rooms*, que son la estructura básica o pantallas de una aplicación GM. Space invasors cuenta con las siguientes *rooms*.

- Room Main Menu.
- Room Game.
- Room Dead.
- Room Victory.

Room Main Menu.

Room que se va ejecutar en primer lugar y que mostrará el mensaje de inicio de partida, véase Figura 102.



Figura 114: Room Main Menu

Objetos en Room Main menú.

1. Objeto text

El objeto va tener dos funciones principales:

- Mostrar el mensaje de inicio de juego.
- Desencadenar el evento pulsación de tecla espacio que va cambiar a la Room del juego, véase Figura 103.

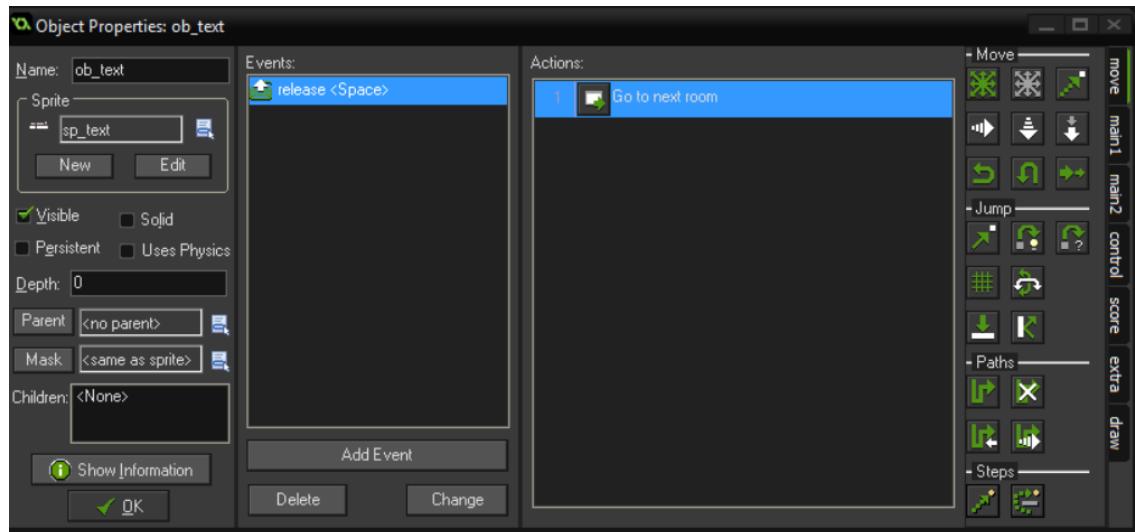


Figura 115: Evento cambio de Room

Room Game.

Constituye el juego en si mismo. Va contener toda la lógica y objeto que permiten que el juego tenga las funcionalidades descritas en los requisitos.

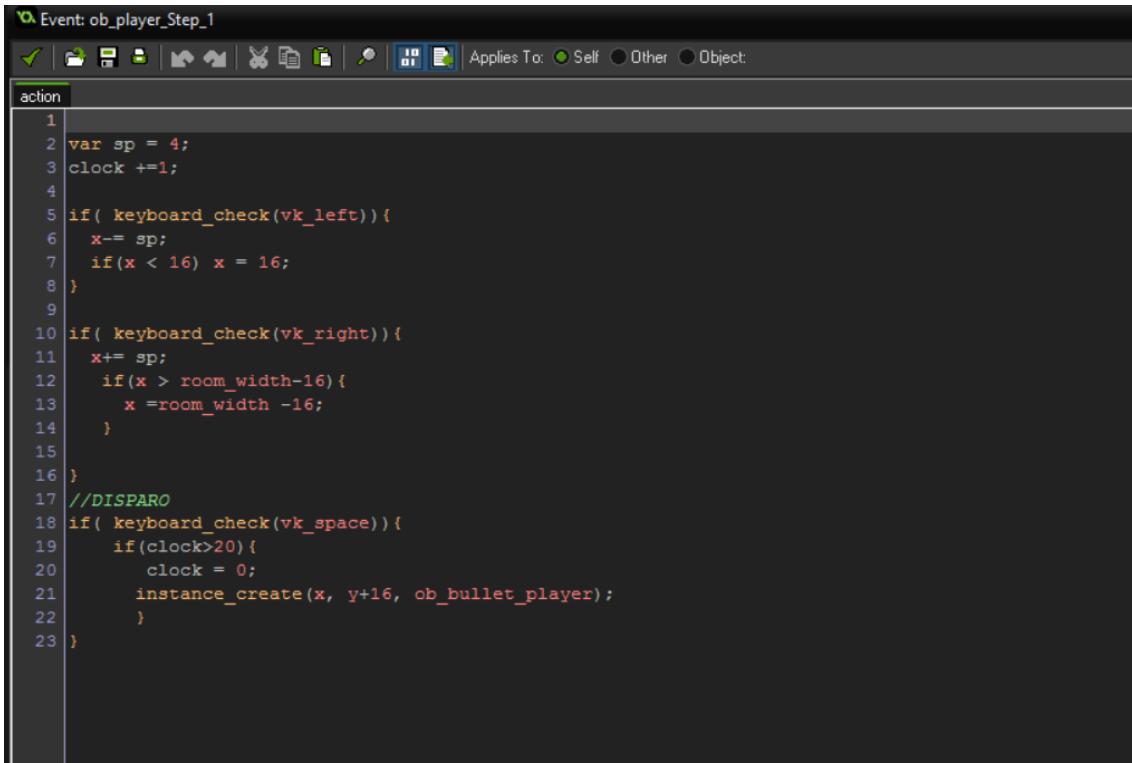
Objetos en Room Game.

1. Objeto Player.

El objeto player se va encargar de controlar el movimiento del jugador, el disparo de balas por parte de este y el estado vivo o muerto del mismo.

Para ellos va tener los siguientes eventos y funcionalidades.

- **Evento Create:** Este evento se va desencadenar cuando el jugador sea creado y va tener asociado las acciones: poner el estado vivo al jugador; darle tres puntos de vida; crear una variable tipo reloj con valor 0 que será utilizada en el disparo de balas.
- **Evento Step:** En cada ciclo de juego se van a ejecutar las siguientes acciones: ejecutar un código que va a realizar las acciones de movimiento y disparo, véase Figura 103; y comprobar si el jugador está vivo, sino es así cambiará a la Room de muerte.



```

Event: ob_player_Step_1
action
1
2 var sp = 4;
3 clock +=1;
4
5 if( keyboard_check(vk_left)) {
6     x-= sp;
7     if(x < 16) x = 16;
8 }
9
10 if( keyboard_check(vk_right)) {
11     x+= sp;
12     if(x > room_width-16) {
13         x = room_width -16;
14     }
15 }
16 }
17 //DISPARO
18 if( keyboard_check(vk_space)) {
19     if(clock>20){
20         clock = 0;
21         instance_create(x, y+16, ob_bullet_player);
22     }
23 }

```

Figura 116: Jugador movimiento y disparo

- **Evento colisión con bala enemiga:** En este evento se va ejecutar el código que reduce la vida del jugador en 1 puntos y cambiará el Sprite, y destruir la bala que ha colisionado con el jugador.

2. Objeto Bala Jugador y objeto bala enemigo

En el juego en Game Maker por facilidad de implementación se ha distinguido entre bala del jugador y bala del enemigo. Ambos van a tener los mismos eventos, por un lado, un evento de movimiento en el caso del jugador movimiento hacia arriba y en del enemigo hacia abajo, y por otro, un evento que va destruir la bala si se encuentra fuera de la pantalla.

3. Objeto director de juego.

En GM, es muy habitual y recomendable el uso de objetos directores que son aquellos que están destinados a controlar los parámetros del juego como por ejemplo, el número de enemigos en pantalla, su lugar de creación, los objetos en pantalla, etc.

En el caso de Space Invaders el objeto director tiene los siguientes:

- **Evento create:** Se va encargar de crear los enemigos y edificios del juego y de definir las variables globales de la dirección del movimiento de los enemigos, véase Figura 106.

```

action
1 global.movetoleft = true;
2 global.movetodown = false;
3
4 for(var i =1; i<5; i++){
5     for(var e= 1; e<8; e++) {
6         instance_create(e*70 +300,i*70,ob_enemy)
7     }
8 }
9
10 for(var e= 0; e<6; e++) {
11     instance_create(e*200+70,580,ob_build)
12 }
13
14

```

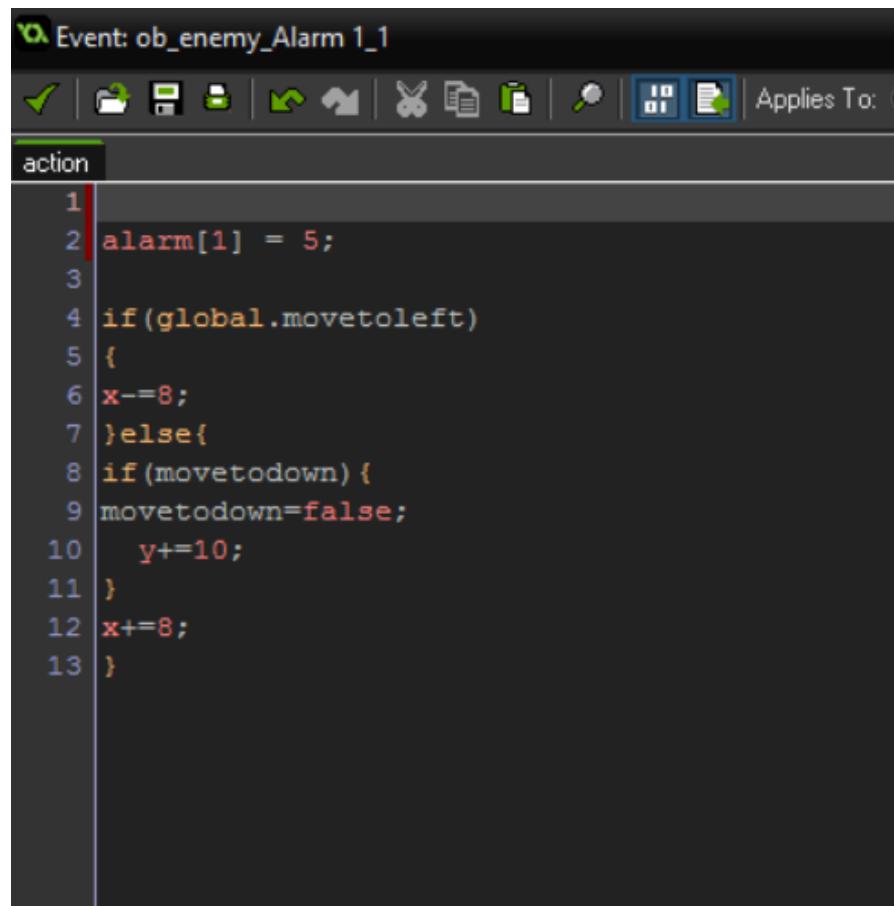
Figura 117: Creación de enemigos y edificios

- **Evento Step:** En este evento se va comprobar en cada ciclo de ejecución si existen enemigos en juego, sino existen lanzará la room de victoria.

4. Objeto enemigo.

El objeto enemigo tiene los eventos siguientes:

- **Evento créate:** Va dar a el enemigo la vida, el estado vivo, crear una alarma que se lance cada 5 ciclos de ejecución y poner definir la variable de movimiento hacia abajo.
- **Evento alarma 1:** Se va encargar de generar el movimiento del enemigo, véase Figura 106, cada 5 ciclos de ejecución.



The screenshot shows the Construct 2 event editor for an event named "Event: ob_enemy_Alarm 1_1". The interface includes a toolbar with various icons like checkmark, file operations, and search. Below the toolbar is a header bar with tabs and a dropdown labeled "Applies To:". The main area is a code editor titled "action" with the following pseudocode:

```

1
2 alarm[1] = 5;
3
4 if(global.movetoleft)
5 {
6 x-=8;
7 }else{
8 if(movetodown) {
9 movetodown=false;
10 y+=10;
11 }
12 x+=8;
13 }

```

Figura 118: Movimiento de enemigos

- **Evento Steep:** Va ejecutar el código que se va encargar de generar disparos aleatorios por parte del enemigo y comprobar si ha llegado a los límites de la pantalla y evaluar si el enemigo está vivo o no, si no lo está destruirá la instancia.
- **Evento Colision con bala de jugador:** Si el enemigo colisiona con una bala de jugador se va quitar una vida al enemigo y se va destruir la bala.

5. Objeto edificio.

El objeto edificio o *build* va contar con los siguientes eventos, véase Figura 107:

- **Evento create:** Al crear el edificio se le van a dar 4 puntos de vida y su estado alive se pondrá a true.
- **Evento Step:** En cada ciclo de ejecución del juego se va comprobar si el objeto está vivo, sino lo está se destruirá la instancia.

- **Evento colisión con bala de personaje:** Si se produce una colisión con una bala del jugador está se destruira.
- **Evento colisión con bala de enemigo:** A parte de destruirse el objeto bala al colisionar con una bala enemiga también su vida será reducida en 1 punto y se cambiará el Sprite.

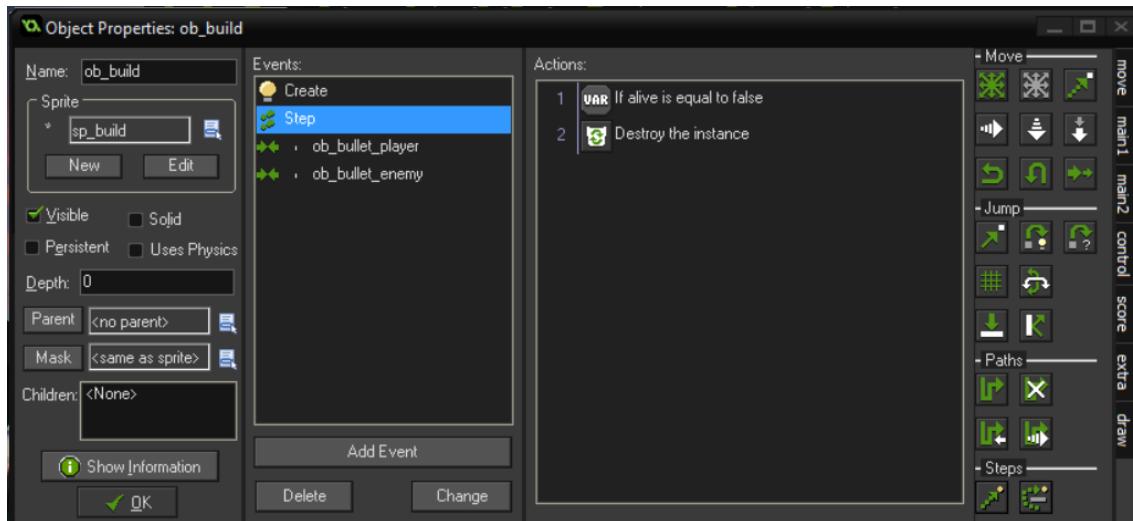


Figura 119: Eventos objeto edificio

Room Victory y room dead.

La room de victoria y la room dead van a contar ambas con un mensaje, aunque con texto distinto que muestre si se ha ganado, matando a todos los enemigos o si ha muerto por que el jugador se ha quedado sin vidas.

Resumen de juego en Game Maker.

En la siguiente tabla se resume los datos del juego realizado con GM.

Tiempo invertido	Consumo medio cpu	Consumo medio Ram	FPS
2h 50m	3%	59,50Mb	60

Tabla 214: Space Invaders Game Maker datos

6. Análisis de Resultados.

6.1. Resultados de análisis de frameworks de creación de videojuegos

En cuerpo del trabajo se analizado cuatro frameworks de creación de videojuegos bajo unos criterios definidos previamente. En este apartado, se van a analizar estos resultados de cada uno de los programas en los siguientes aspectos:

- Valoración general del programa.
- Aspectos positivos.
- Aspectos negativos.
- Uso recomendado.

De cara a la valoración de los criterios de cada programa se considerará que la puntuación de cada uno de ellos tiene la consideración que muestra la siguiente tabla.

CONSIDERACIÓN CRITERIOS	
Entre 0 y 50%	Criterio considerado malo
Entre 50 y 70%	Criterio con consideración normal
Más de 70%	Criterio con consideración buena.

Tabla 215: Consideración de criterios.

Este análisis se va realizar en orden ascendente desde el programa con la puntuación más baja a la más alta. En la siguiente figura se muestra un gráfico con la puntuación total sobre 100 de los programas.

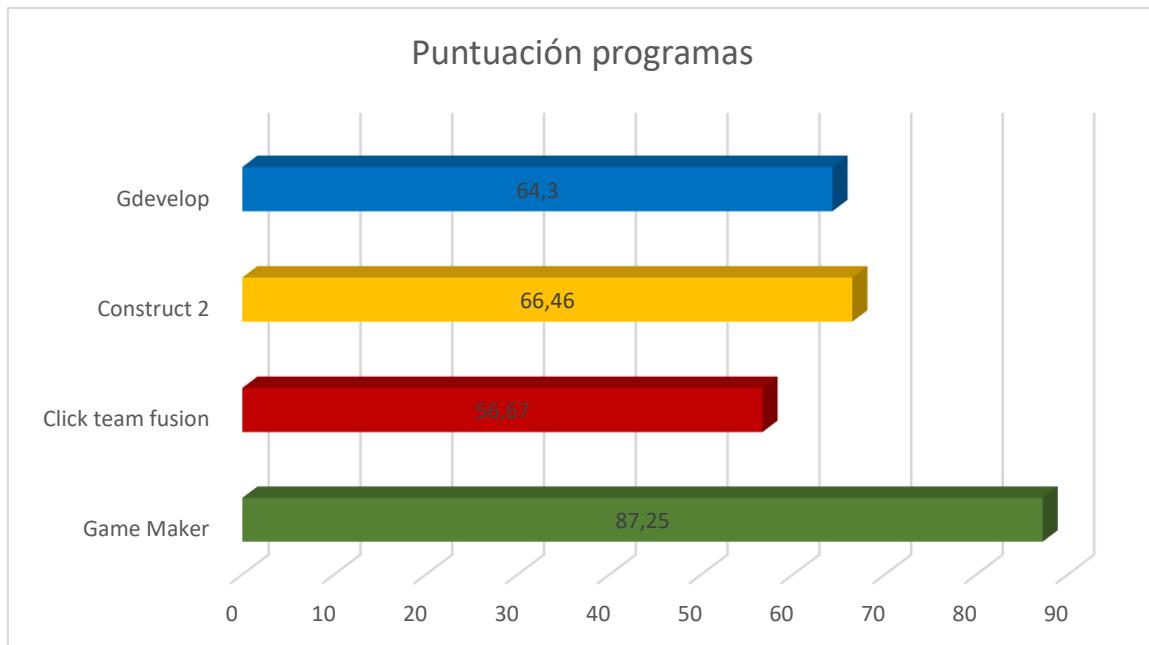


Figura 120: Puntuación programas

Vista la puntuación el orden será el siguiente: Click Team Fusión, Gdevelop , Construct 2 y Game Maker.

6.1.1 Análisis resultados Click Team Fusion

Si hay un programa de los analizados que se quedan bastante lejos del resto ese es CTF. Su puntuación total ha sido de 56,67 sobre 100.

En el siguiente gráfico se detalla la puntuación en los diferentes criterios.

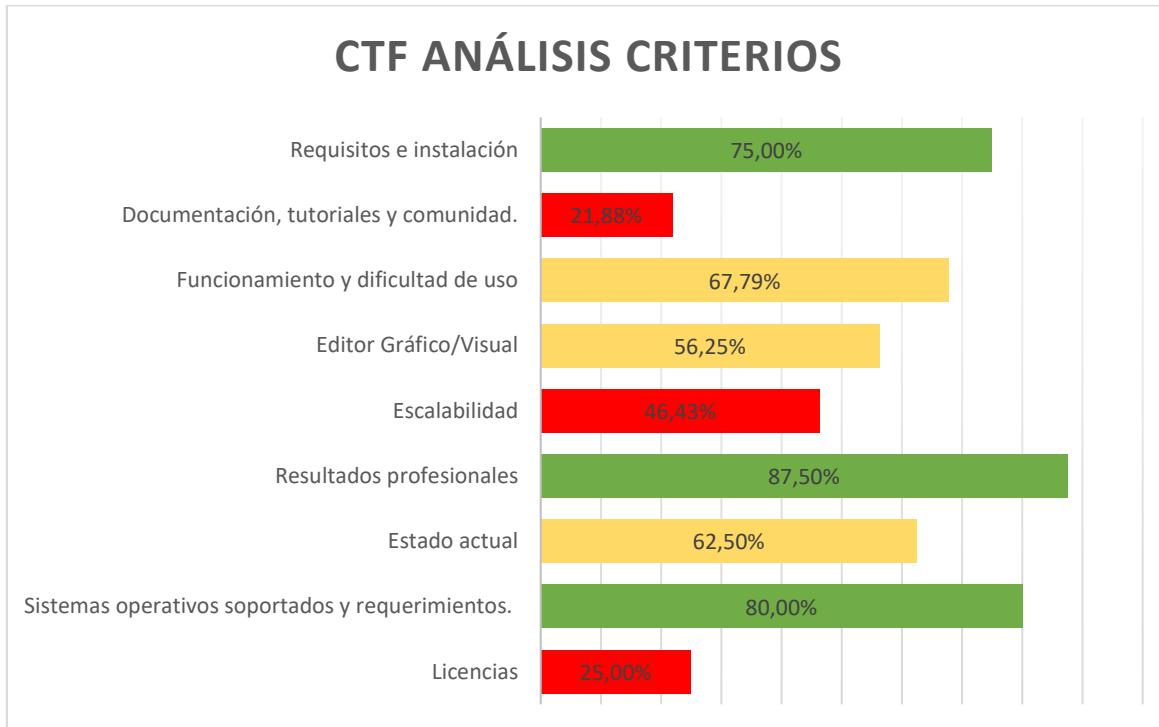


Figura 121: Gráfico CTF criterios

Como muestra el siguiente gráfico CTF tiene 3 criterios considerados buenos, 3 de consideración normal y 3 definidos como malos.

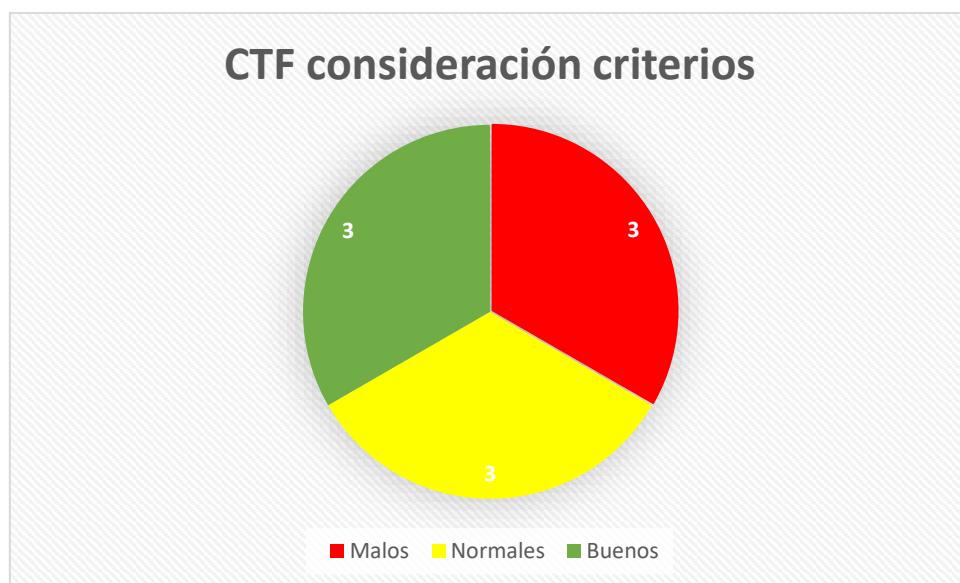


Figura 122: Consideración criterios

CTF tiene los siguientes aspectos positivos:

- CTF tienen unos buenos resultados profesionales. Durante el análisis se ha podido comprobar que existen juegos comerciales de éxito que han sido desarrollados con la herramienta.
- Junto con Game Maker, es el programa que mayores posibilidades de exportación tiene, lo que garantiza que los juegos creados puedan ser jugados en multitud de plataformas y sistemas.

En cuanto a los aspectos más negativos de CTF destacan los siguientes:

- La documentación que ofrece es escasa, de mala calidad y con multitud de errores y fallos.
- Los tutoriales oficiales, por su gran cantidad de errores no ofrecen las posibilidades de aprendizaje para el usuario que se podría esperar.
- Comparativamente con el resto de programas, es el que tiene unos precios tanto de licencias como de módulos más elevados.
- La escalabilidad es escasa. Aunque cuenta con una gran cantidad de addons y plugins disponibles para su compra, la creación de nuevas funcionalidades por parte del usuario es muy costosa y sin apenas documentación para ello.

Por los errores en su documentación oficial y en los tutoriales CTF no se puede recomendar a usuarios sin experiencia de programación o en este tipo de programas, ya que tendrá que aprender mediante ensayo y error, y sin saber si la forma en que implementa las funcionalidades es la más adecuada.

Su sistema de eventos basado en rejilla, véase Figura 103, hace que su funcionamiento sea diferente al resto, mientras q en los otros si que pueda haber ciertos patrones de uso parecidos que permiten cierto aprendizaje compartido entre ellos. Este sistema de eventos puede resultar adecuado para proyectos pequeños, pero se vuelve demasiado caóticos y difícil de gestionar en proyectos de tamaño medio o grande.

All the events All the objects											
1	• collides with the background										
2	• Speed of <= 0										
3	• Speed of > 0										
4	• Always										
5	• collides with the background										
6	• New condition										

Figura 123: Sistema de eventos CTF

Por todo lo visto CTF es el programa menos recomendable de todos los analizados, no contando con las herramientas que permiten a los usuarios aprender correctamente su funcionamiento. Únicamente es recomendable para usuarios que quieran probar un programa con una forma de trabajar distinta al resto y se adapten rápidamente a su sistema de eventos.

6.1.2 Análisis resultados Gdevelop.

La puntuación de Gdevelop es de 64,3 sobre 100. En la siguiente figura se muestra la puntuación que ha obtenido en cada uno de los criterios.

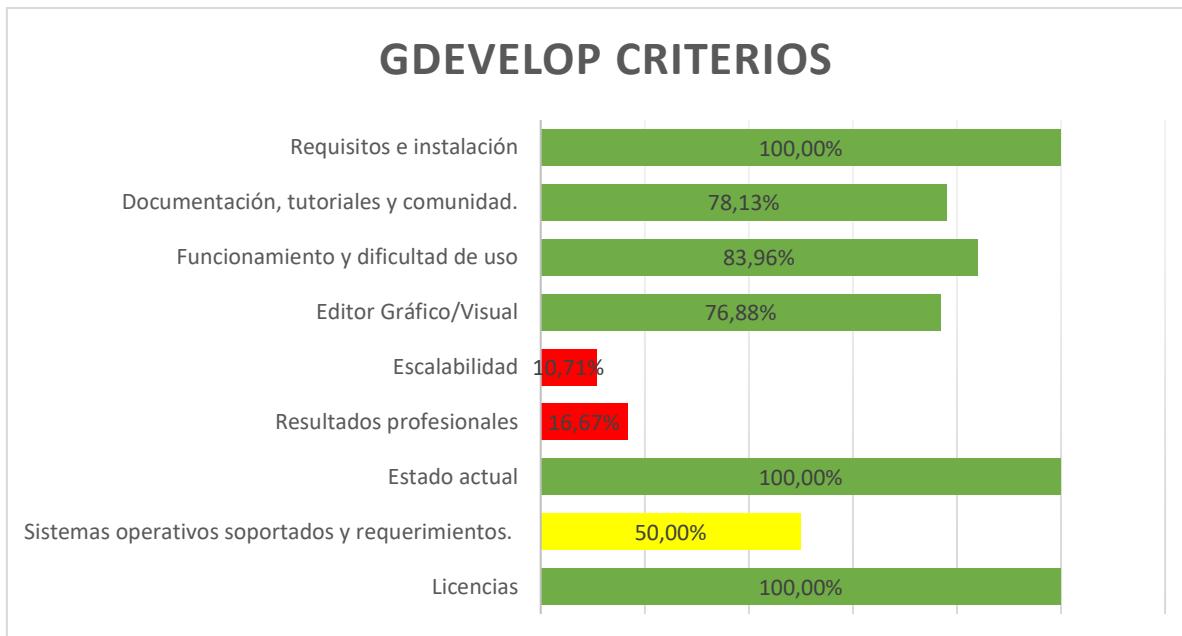


Figura 124: Gdevelop criterios

Como se puede comprobar en el siguiente gráfico Gdevelop solo tiene un criterio mal y uno normal, en el resto consigue una buena puntuación.

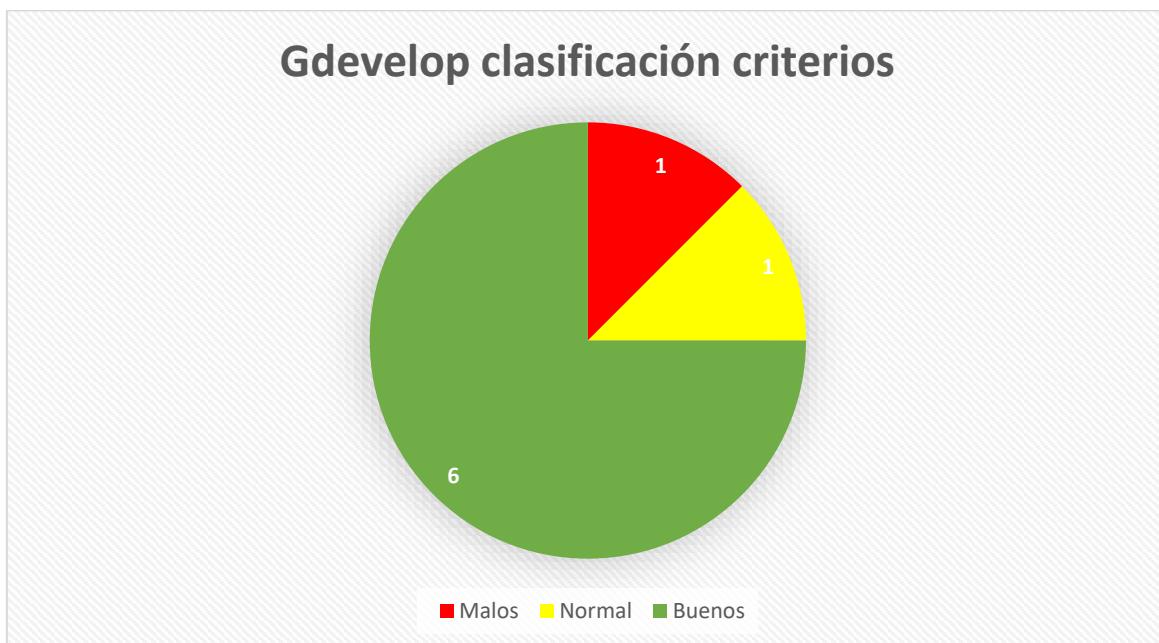


Figura 125: Gdevelop clasificación criterios

El programa destaca sobretodo en los siguientes aspectos:

- El programa es libre y de código abierto, por lo que es totalmente gratuito.
- La documentación disponible es extensa y bien realizada.
- No es necesario tener ningún tipo de conocimiento previo, ni de lógica de programación ni de herramientas de este tipo.
- Muy sencillo de usar y de implementar las funcionalidades básicas de los juegos.
- Su curva de aprendizaje es asequible para todo tipo de usuarios.
- Los requisitos de funcionamiento son los más bajos de todo el estudio y además se puede usar en multitud de plataformas e incluso en versión web.

Por el contrario, tiene también los aspectos negativos siguientes:

- Las posibilidades de escalabilidad son nulas ya que es muy difícil la creación de nuevas funcionalidades y tampoco dispone de plugins ni addons.
- No existen juegos comerciales realizados con la herramienta.
- No es una herramienta destinada a realizar grandes proyectos.

Gdevelop es quizás el programa más sencillo de usar de los analizados. Las funcionalidades se crean de forma rápida y cuenta con multitud de ayudas visuales para que los usuarios entiendan el funcionamiento de la herramienta.

Es un programa adecuado para usuarios sin experiencia en programación ni en este tipo de herramientas que quieran iniciarse en la creación de videojuegos. Con Gdevelop podrán sin demasiado esfuerzo, gracias a los buenos manuales y tutoriales, crear proyectos de tamaño pequeño y medio prácticamente desde las primeras horas. Si el objeto del estudio hubiera sido encontrar el mejor framework para usuarios sin ningún tipo de conocimiento Gdevelop hubiera sido el vencedor.

Si bien no es recomendable para usuarios avanzados y con intención de crear grandes proyectos, ya que el programa no está pensado para ello.

6.1.3. Análisis resultados Construct 2

Construct 2 ha obtenido un 66,46 sobre 100. En el siguiente gráfico se muestra la puntuación de Construct 2 en los diferentes criterios.

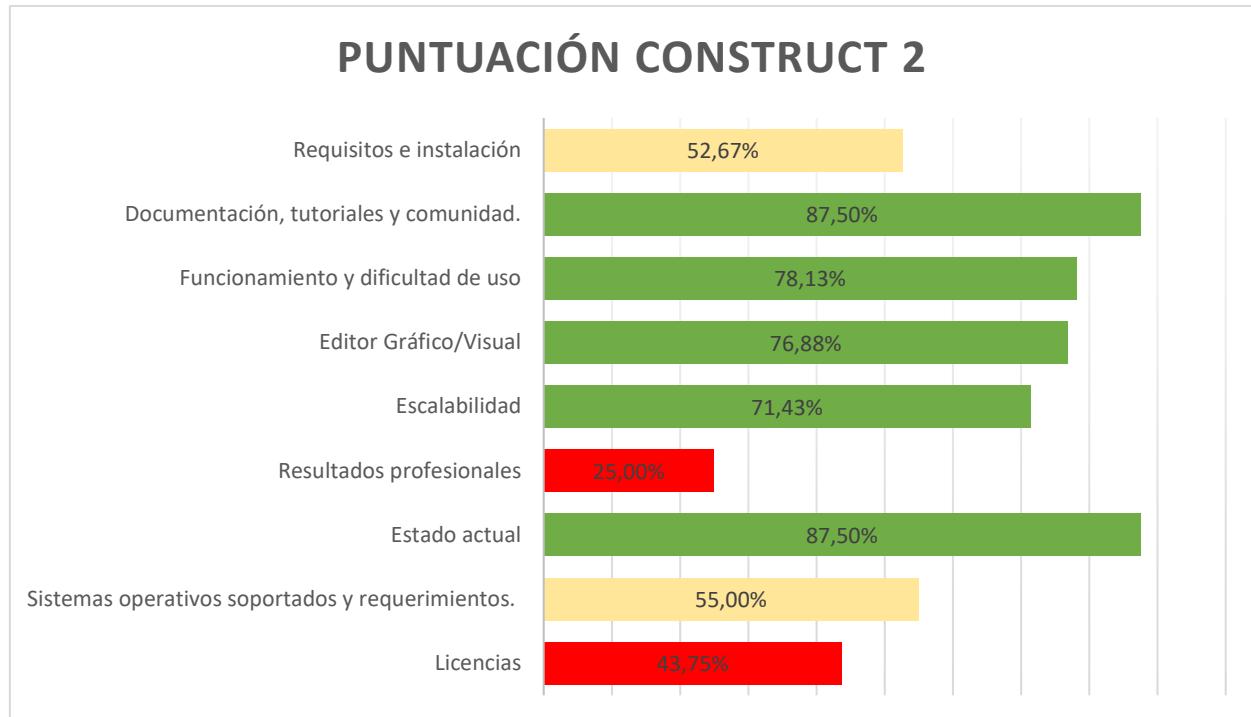


Figura 126: Construct 2 puntuación criterios

Como se puede comprobar Construct 2 es un programa bastante equilibrado ya que solo suspende en dos criterios y dos tienen una consideración normal en el resto su calificación es buena, véase Figura 107.

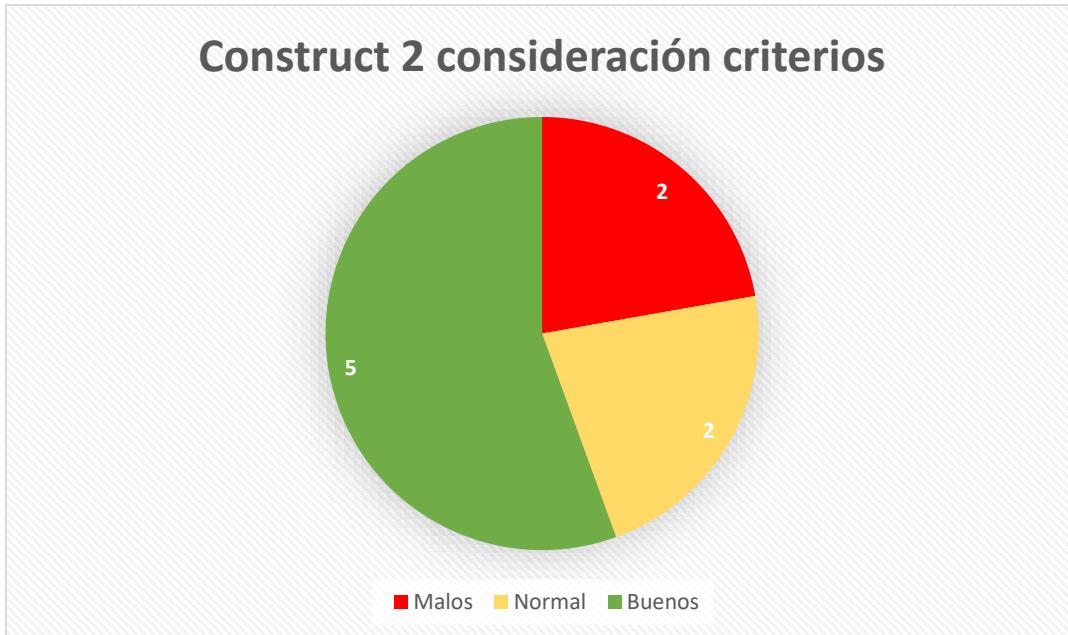


Figura 127: Construct 2 consideración criterios

Construct 2 destaca en los siguientes aspectos:

- Tiene gran cantidad de recursos para el aprendizaje, para todo tipo de usuarios y niveles, lo que facilita el acercamiento al programa.
- Los tutoriales oficiales analizados son los más completos y correctos de todos los analizados.
- El programa es muy escalable, haciendo uso de Javascript y Json para aumentar las funcionalidades base.
- La licencia gratuita permite un uso casi completo del programa y las limitaciones que tienen no impiden trabajar y crear proyectos con cierta complejidad.

Por el contrario, tiene los siguientes puntos negativos:

- Los requisitos de instalación son los más altos de todo el estudio, ya que para funcionar requiere 2GB de RAM y 1Ghz de procesador.
- Su uso profesional es meramente anecdótico, no se han encontrado ejemplo de juegos comerciales realizados con Construct 2.
- Tiene la licencia más cara de todas las del resto de programas, con un precio de 405.99€.

- La nomenclatura del programa ésta basada en la lógica de programación , lo que puede dificultar en un primer momento el uso a usuarios que no conocen ésta lógica.

Su modo gratuito combinado con su gran documentación y la cantidad de buenos tutoriales convierte a Construct 2 en una de las mejores opciones para usuarios que quieran inciarse en el desarrollo de juegos con este tipo de programas. Aunque es uso de nomenclatura de programación hace recomendable un cierto conocimiento de la lógica de la misma, aunque no es una dificultad excesivamente grande hay que tenerlo en cuenta a la hora de inciarse con Construct 2.

Además, por sus altas dosis de escalabilidad que permite la creación de nuevas funciones y scripts es un programa adecuado, también para usuarios avanzados en programación, sobre todo en Javascript y JSON. Pero la ausencia de juegos comerciales desarrollados con Construct hace que no sea adecuado si se busca crear proyectos de este tipo.

6.1.4. Análisis resultados Game Maker.

Game Maker es el programa que ha obtenido una puntuación más alta con 87.25 sobre 100. La puntuación en la mayoría de los criterios como muestra el gráfico siguiente es bastante alta.



Figura 128: Game Maker puntuación criterios

El programa no suspende en ningún criterio analizado y solo tiene dos clasificados como normal, el resto tienen una calificación buena, véase Figura

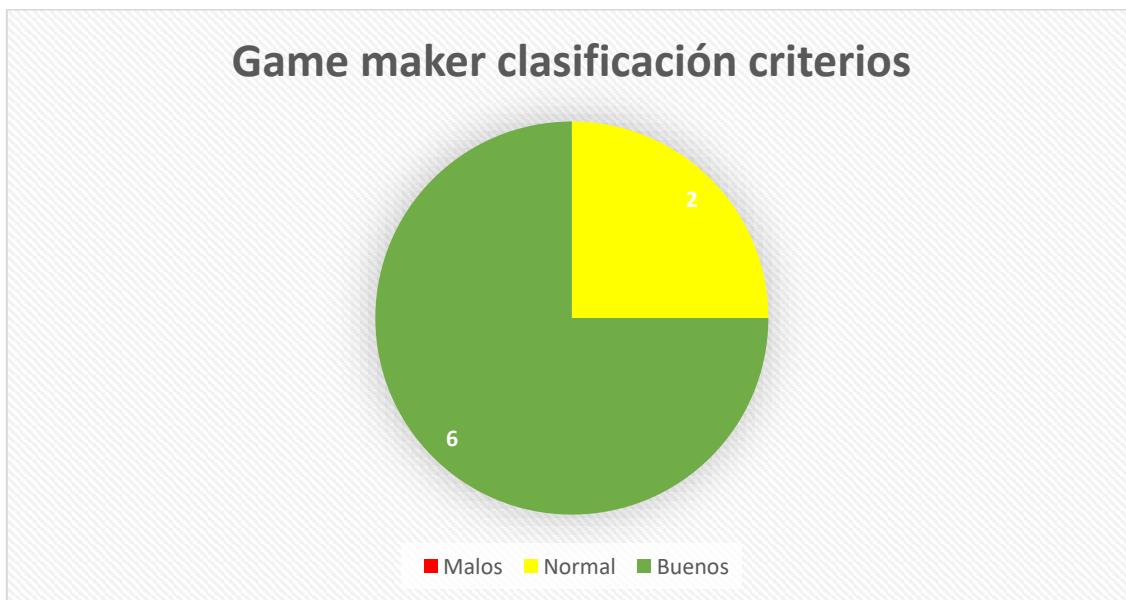


Figura 129: Game maker clasificación criterios

Game maker destaca en los siguientes aspectos:

- Dispone de muchos tutoriales y proyectos ejemplo que además algunos de ellos se pueden visualizar dentro del programa.
- Las posibilidades de exportación cubren prácticamente todas las plataformas y sistemas.
- Tiene dos modos de funcionar complementarios, modo programación y modo drag and drop lo que permite adaptarse a todo tipo de usuarios.
- Su lenguaje propio de programación GML da al programa una escalabilidad total.
- Ha demostrado con la multitud de juegos comerciales con gran éxito de público y crítica que es un programa adecuado para grandes proyectos.

Pero también tiene aspectos negativos como los citados a continuación:

- El precio de sus módulos de exportación es bastante alto.
- Su modo gratuito tiene demasiadas limitaciones lo que impide que sea una opción.
- La versión Game Maker 2 ésta apunto de ser lanzada lo que obligará a realizar una futura migración, aunque los proyectos serán compatibles y la implementación de las funcionalidades serán muy parecidas.

Game Maker es un programa recomendado para todo tipo de usuarios y niveles. Su escalabilidad, junto a sus dos modos de funcionar, drag and drop y modo programación hace que se adapte perfectamente al nivel del usuario. Hay que tener en cuenta que el modo gratuito, por sus limitaciones, no es una opción por lo que habrá que hacer un desembolso y comprar la licencia que tiene un precio de 99€.

Hay que indicar que independientemente del nivel de usuario el aprendizaje de la herramienta lleva irremediablemente a aprender su lenguaje de programación GML. Con GML se pueden llegar a realizar las funcionalidades más avanzadas que realizadas con el modo drag and drop serían muy difícil de realizar o muy costosas. Pero el uso de GML es progresivo conforme el nivel del usuario avanza con lo que su aprendizaje no resulta excesivamente costoso para usuarios sin conocimientos de programación.

6.2 Resultados comparación videojuego en C++ vs videojuegos en Game Maker.

6.2.1 Resultados visuales

Los resultados visuales de ambos juegos son prácticamente idénticos como muestran las siguientes imágenes.



Figura 130: Space Invaders C++

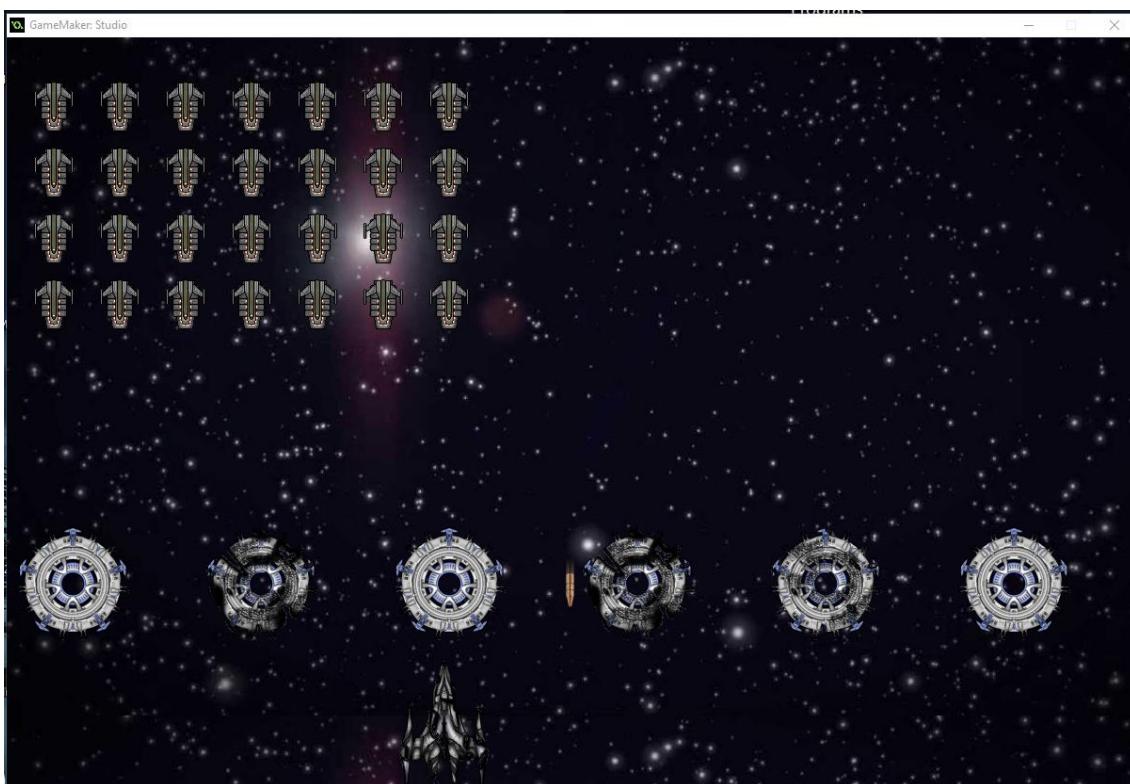


Figura 131: Space Invaders Game Maker

6.2.2 Rendimiento

El rendimiento de los juegos es bastante parecido en ambos desarrollos, aunque como muestra la tabla siguiente, el consumo de memoria es algo más bajo en GM y el cpu en SFML.

Aplicación	Consumo medio cpu	Cosumo medio Ram	FPS
Game Maker ejecutable	3%	59,50Mb	60
SFML Ejecutable	2%	74,30Mb	60

Tabla 216: Rendimiento Space Invader C++ y GM

Si bien en un primer momento se podía pensar que en Game Maker el uso de memoria y el de CPU iba a ser mayor, se ha podido comprobar que una vez realizada la implementación los resultados son muy parecidos e incluso GM hace uso de menos memoria que C++ y SFML.

Las posibles razones de este menor consumo de memoria de GM y prácticamente el mismo de CPU puede encontrarse en las siguientes:

- Al contrario de lo pensado, GM introduce menos capas intermedias o las que introduce no consumen apenas recursos.
- La estructura del código de GM es muy eficiente, situa los elementos en su lugar evitando fugas de memoria y consumos excesivos. Su gestión de eventos y objetos esta pensada para ahorrar recursos.
- El consumo de recursos en C++ depende exclusivamente de la experiencia del programador y de su capacidad para ahorrar memoria y uso de cpu. Mientras que en GM toda esa gestión se hace automáticamente, por lo es prácticamente invulnerable a implementaciones poco eficaces.

6.2.3 Tiempo invertido.

Donde se ven mayores diferencias es en el tiempo invertido para la realización de los juegos en las dos plataformas como muestra la tabla siguiente.

Aplicación	Tiempo invertido
Game Maker ejecutable	2h 50m
SFML Ejecutable	18h 45m

Tabla 217: Tiempo invertido Space Invasors

El juego en GM se ha desarrollado en menos de tres horas y el juego de C++ en casi 19 horas.

Las principales razones que explican esta diferencia de tiempo son las siguientes:

- Game Maker esta pensado para crear juegos de una forma rápida y eficaz, por lo que todo el entorno ayuda a esa velocidad de desarrollo. Mientras que el desarrollo en SFML parte de un programa en C++ de carácter generalista y no pensando en un principio para el desarrollo de juegos.
- En el juego en C++ y SFML se ha tenido que crear todo el bucle del juego y la gestión del tiempo para que funcione a la misma velocidad en todos los equipos. Por el contrario, en GM la gestión del bucle de juego y del tiempo ya esta realizada y hay que implementarla.
- El sistema de eventos y acciones en GM ahorra mucho tiempo de desarrollo.

6.2.4 Resumen final y conclusión.

Como se ha podido comprobar GM ofrece un rendimiento mejor en memoria y unos tiempos de desarrollo mucho más cortos que el juego en la versión C++. Las ventajas principales del desarrollo de GM se encuentran en la velocidad de implementación y la buena gestión de recursos de la CPU y de memoria.

Se recomienda el uso de GM para la creación de juegos a desarrolladores y usuarios que únicamente quieran crear juegos y no les importe aprender un programa y lenguaje de uso únicamente en un sistema.

Pese al mayor tiempo de implementación en SFML, es recomendable para usuarios que no quieran quedarse únicamente en desarrollar juegos, sino que quieran en un futuro crear otras aplicaciones y sistemas. Además, al tratarse de un programa en C++ se está aprendiendo un lenguaje de programación universal y que puede usarse en multitud de entornos y sistemas, incluso con en GM con un plugin.

7. Conclusiones.

7.1. Objetivos cumplidos.

Con la realización del presente proyecto se ha conseguido la consecución de los objetivos marcados.

Se han analizados los principales frameworks de este tipo disponibles y se ha llegado a la conclusión de que el más adecuado para un uso generalista y con más posibilidades de escalabilidad es Game Maker.

Game Maker dispone de una gran cantidad tutoriales, proyectos ejemplo y manuales para que el aprendizaje sea posible a todos los niveles. Además, cuenta con dos modos de funcionar, cubriendo tanto a usuarios con conocimientos de programación como sin ellos, modo programación y modo Drag and Drop. Existen también bastantes ejemplos de juegos comerciales desarrollados con Game Maker como, por ejemplo, Undertale, Hotline Miami o Spelunky, por lo que su uso para grandes proyectos es posible y existen desarrolladores trabajando profesionalmente con él.

Se ha realizado un proyecto con C++ y otro con Game Maker y se ha comparado ambas formas de implementación como se marcaba en los objetos fundamentales del proyecto. Las conclusiones están lejos de lo esperado, ya que en un primer momento se pensaba que los consumos de memoria y recurso iba a ser bastante mayor en GM, pero no ha sido así. Los resultados han sido bastante parecido, incluso con un menor consumo de memoria por parte de GM, con un tiempo de implementación mucho menor.

Por lo visto anteriormente se ha conseguido completar todos los objetivos propuestos al inicio del proyecto.

¿Qué es un kit de desarrollo de software (SDK)? (2013). Obtenido de 4 R Soluciones:

<http://www.4rsoluciones.com/blog/que-es-un-kit-de-desarrollo-de-software-sdk-2/>

2d and 3d in Game Maker Studio. (2015). Obtenido de Like 100 bears:

<http://www.like100bears.com/writing/2d-3d-in-gamemaker-studio>

2D Game Creation. (2017). Obtenido de Unity 3D:

<https://unity3d.com/es/learn/tutorials/s/2d-game-creation>

Adventure Game Studio. (s.f.). Obtenido de <http://www.adventuregamestudio.co.uk/>

Angel 2D. (2013). Obtenido de <https://angel2d.com/>

Angel 2D vs SFML. (2017). Obtenido de Slant:

https://www.slant.co/versus/1059/1100/~angel2d_vs_sfml

Autores, V. (2016). *Programmers Guide.* Obtenido de Cocos 2D X:

<http://www.cocos2d-x.org/docs/ProgrammersGuide.pdf>

Candil, D. (2012). *Cuatro motores gráficos para perder el miedo y lanzarse al desarrollo de videojuegos.* Obtenido de Vida Extra:

<https://www.vidaextra.com/listas/4-motores-graficos-para-perder-el-miedo-y-lanzarse-al-desarrollo-de-videojuegos>

Click Team Fusion 2.5. (2017). Obtenido de <http://www.clickteam.com/clickteam-fusion-2-5>

Click Team Fusion Review. (2017). Obtenido de Slant:

<https://www.slant.co/options/6626/~clickteam-fusion-2-5-review>

Clickteam Fusion 2.5. (2016). Obtenido de Click Wiki:

https://clickwiki.net/wiki/Clickteam_Fusion_2.5

Cocos 2D X. (2017). Obtenido de <http://www.cocos2d-x.org/>

Comunidad Game Maker. (2011). Obtenido de

<http://www.comunidadgm.org/index.php>

Construct 2 vs Game Develop, why nobody using it? (2014). Obtenido de Scirra Forum:

https://www.scirra.com/forum/construct-2-vs-game-develop-why-nobody-using-it_t115815?sid=ebb38308144e95d89880be87f23d776f&start=10

Contact Us. (2014). Obtenido de Air Scapte Game:

<http://airscapegame.com/index.html#contactus>

Curso de Game Maker. (2017). Obtenido de Edutin: <https://edutin.com/curso-de-GameMaker-y-GML-2018>

Eclipse Origins. (2013). Obtenido de RPG Maker Net:

<https://rpgmaker.net/engines/eclipseor/>

Engine Click Team Fusion. (2016). Obtenido de Taringa:

<http://www.taringa.net/comunidades/creatusjuegos/9178104/Engine-Clickteam-Fusion-2-5-Multimedia-Fusion.html>

Experiencia de juego 3D. (2016). Obtenido de Docs Unity:

<https://docs.unity3d.com/es/current/Manual/Overview2D.html>

Extending Construct 2 with Javascript. (2011). Obtenido de Scirra Forum:

https://www.scirra.com/forum/extending-construct-2-with-javascript_t60269

Extension MMF. (s.f.). Obtenido de Dark Wire: <http://dark-wire.com/store/extlist.php>

Fernández, A. L. (2013). *Curso de creación de videojuegos con Game Maker.* Obtenido de <http://toodaim.blogspot.com.es/2013/02/curso-gamemaker-crea-videojuegos.html>

Game Engine Game Develop. (2011). Obtenido de Taringa:

<https://www.taringa.net/posts/hazlo-tu-mismo/13168821/Game-Engine-Game-Develop.html>

Game Maker blog. (2014). Obtenido de

<http://gammemakerblog.com/2014/06/07/gamemaker-studio-standard-is-now-completely-free/>

Game Maker Vs Unity. (2016). Obtenido de Yises Blog:

<http://yises.com/blog/2016/03/28/unity-vs-gamemaker-cual-escoger/>

Game Showcase. (2016). Obtenido de HTML 5 Game Dev:

<http://www.html5gamedevs.com/forum/8-game-showcase/>

Games dev. (2017). Obtenido de <http://www.gamedev.es/>

Gdevelop change list. (2017). Obtenido de Compile Games Forum:

<http://forum.compilegames.net/viewtopic.php?f=18&t=8532>

Gdevelop Documentation. (2015). Obtenido de Wiki Compile Games:

<http://www.wiki.compilegames.net/doku.php/es/gdevelop/documentation>

Gdevelop Road Map. (2015). Obtenido de Trello:

<https://trello.com/b/qf0IM7k8/gdevelop-roadmap>

Getting Started. (2016). Obtenido de Crosswalk Project: https://crosswalk-project.org/documentation/getting_started.html

GNU Lesser General Public License. (2016). Obtenido de GNU Org:

<https://www.gnu.org/licenses/lgpl.html>

GODOT Engine. (2017). Obtenido de <https://godotengine.org/>

Guide to Intalling extensions. (2017). Obtenido de Community Click Team Fusion:

<https://community.clickteam.com/threads/83607-Guide-to-installing-Extensions>

Gutiérrez, J. J. (s.f.). *¿Qué es un framework web?*. Obtenido de
http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf

Hektor Programador. (2016). Obtenido de <http://blog.hcosta.info>

How Distribute your Games. (2015). Obtenido de Wiki Compile Games:
<http://wiki.compilgames.net/doku.php/gdevelop/tutorials/howtodistribute>

Indie Games CTF. (2017). Obtenido de <http://indiegames.clickteam.com/>

License Zlib. (2017). Obtenido de Open Source Org:
<https://opensource.org/licenses/Zlib>

Linietsky, J. (2016). *GODOT en Español*. Obtenido de <http://godot-doc-en-espanol.readthedocs.io/es/latest/>

López, D. (2017). *Aprende Game Maker*. Obtenido de
<http://www.aprendegamemaker.com/>

Lua ORG. (s.f.). Obtenido de <http://www.lua.org/>

Multimedia Fusion Examples. (2008). Obtenido de Castles of Britain:
<http://www.castles-of-britain.com/mmfexamples-0.htm>

Peinado, C. C. (s.f.). *Breve historia de motores de juegos*. Obtenido de
<https://www.gameit.es/breve-historia-de-motores-de-juegos-episodio-i/>

Repositorio Universidad de Alicante. (2017). Obtenido de
<http://rua.ua.es/dspace/handle/10045/36932>

Review Mortal Melon. (2015). Obtenido de Nintendo Life:
http://www.nintendolife.com/reviews/wiiu-eshop/mortar_melon

Scirra License. (2017). Obtenido de Scirra:
<https://www.scirra.com/store/licensing#license2>

Spalding, S. (2014). *Game Maker Tutoiales*. Obtenido de
<http://gammakertutorials.com/>

Store Extension. (2017). Obtenido de Click Team Store:
<http://clickstore.clickteam.com/plugins/extensions>

The Lua-MMF2 Combo. (2009). Obtenido de Create Games: <http://www.create-games.com/article.asp?id=1968>

The Sprites Resource. (2000). Obtenido de <https://www.sprites-resource.com/>

Torque 2D: Getting Started 1 . (s.f.). Obtenido de You tube:
<https://www.youtube.com/watch?v=JCHQ1LCn4dk>

True Valhalla. (2017). Obtenido de <https://www.truevalhalla.com/blog>

Turnes, Y. (2013). *Game Dic.* Obtenido de <http://www.gamerdic.es/>

Useful Resources. (2017). Obtenido de Click Team Fusion:
<http://www.clickteam.com/resources>

¿Qué es un kit de desarrollo de software (SDK)? (2013). Obtenido de 4 R Soluciones:
<http://www.4rsoluciones.com/blog/que-es-un-kit-de-desarrollo-de-software-sdk-2/>

2d and 3d in Game Maker Studio. (2015). Obtenido de Like 100 bears:
<http://www.like100bears.com/writing/2d-3d-in-gamemaker-studio>

2D Game Creation. (2017). Obtenido de Unity 3D:
<https://unity3d.com/es/learn/tutorials/s/2d-game-creation>

Adventure Game Studio. (s.f.). Obtenido de <http://www.adventuregamestudio.co.uk/>

Angel 2D. (2013). Obtenido de <https://angel2d.com/>

Angel 2D vs SFML. (2017). Obtenido de Slant:
https://www.slant.co/versus/1059/1100/~angel2d_vs_sfml

Autores, V. (2016). *Programmers Guide.* Obtenido de Cocos 2D X:
<http://www.cocos2d-x.org/docs/ProgrammersGuide.pdf>

Candil, D. (2012). *Cuatro motores gráficos para perder el miedo y lanzarse al desarrollo de videojuegos.* Obtenido de Vida Extra:
<https://www.vidaextra.com/listas/4-motores-graficos-para-perder-el-miedo-y-lanzarse-al-desarrollo-de-videojuegos>

Click Team Fusion 2.5. (2017). Obtenido de <http://www.clickteam.com/clickteam-fusion-2-5>

Click Team Fusion Review. (2017). Obtenido de Slant:
<https://www.slant.co/options/6626/~clickteam-fusion-2-5-review>

Clickteam Fusion 2.5. (2016). Obtenido de Click Wiki:
https://clickwiki.net/wiki/Clickteam_Fusion_2.5

Cocos 2D X. (2017). Obtenido de <http://www.cocos2d-x.org/>

Comunidad Game Maker. (2011). Obtenido de
<http://www.comunidadgm.org/index.php>

Construct 2 vs Game Develop, why nobody using it? (2014). Obtenido de Scirra Forum:
https://www.scirra.com/forum/construct-2-vs-game-develop-why-nobody-using-it_t115815?sid=ebb38308144e95d89880be87f23d776f&start=10

Contact Us. (2014). Obtenido de Air Scapte Game:
<http://airscapegame.com/index.html#contactus>

Curso de Game Maker. (2017). Obtenido de Edutin: <https://edutin.com/curso-de-GameMaker-y-GML-2018>

Eclipse Origins. (2013). Obtenido de RPG Maker Net:
<https://rpgmaker.net/engines/eclipseor/>

Engine Click Team Fusion. (2016). Obtenido de Taringa:
<http://www.taringa.net/comunidades/creatusjuegos/9178104/Engine-Clickteam-Fusion-2-5-Multimedia-Fusion.html>

Experiencia de juego 3D. (2016). Obtenido de Docs Unity:
<https://docs.unity3d.com/es/current/Manual/Overview2D.html>

Extending Construct 2 with Javascript. (2011). Obtenido de Scirra Forum:
https://www.scirra.com/forum/extending-construct-2-with-javascript_t60269

Extension MMF. (s.f.). Obtenido de Dark Wire: <http://dark-wire.com/store/extlist.php>

Fernández, A. L. (2013). *Curso de creación de videojuegos con Game Maker.* Obtenido de <http://toodaim.blogspot.com.es/2013/02/curso-gamemaker-crea-videojuegos.html>

Game Engine Game Develop. (2011). Obtenido de Taringa:
<https://www.taringa.net/posts/hazlo-tu-mismo/13168821/Game-Engine-Game-Develop.html>

Game Maker blog. (2014). Obtenido de
<http://gammemakerblog.com/2014/06/07/gamemaker-studio-standard-is-now-completely-free/>

Game Maker Vs Unity. (2016). Obtenido de Yises Blog:
<http://yises.com/blog/2016/03/28/unity-vs-gamemaker-cual-escoger/>

Game Showcase. (2016). Obtenido de HTML 5 Game Dev:
<http://www.html5gamedevs.com/forum/8-game-showcase/>

Games dev. (2017). Obtenido de <http://www.gamedev.es/>

Gdevelop change list. (2017). Obtenido de Compile Games Forum:
<http://forum.compilegames.net/viewtopic.php?f=18&t=8532>

Gdevelop Documentation. (2015). Obtenido de Wiki Compile Games:
<http://www.wiki.compilegames.net/doku.php/es/gdevelop/documentation>

Gdevelop Road Map. (2015). Obtenido de Trello:
<https://trello.com/b/qf0IM7k8/gdevelop-roadmap>

Getting Started. (2016). Obtenido de Crosswalk Project: https://crosswalk-project.org/documentation/getting_started.html

GNU Lesser General Public License. (2016). Obtenido de GNU Org: <https://www.gnu.org/licenses/lgpl.html>

GODOT Engine. (2017). Obtenido de <https://godotengine.org/>

Guide to Intalling extensions. (2017). Obtenido de Community Click Team Fusion: <https://community.clickteam.com/threads/83607-Guide-to-installing-Extensions>

Gutiérrez, J. J. (s.f.). *¿Qué es un framework web?* . Obtenido de http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf

Hektor Programador. (2016). Obtenido de <http://blog.hcosta.info>

How Distribute your Games. (2015). Obtenido de Wiki Compile Games: <http://wiki.compilgames.net/doku.php/gdevelop/tutorials/howtodistribute>

Indie Games CTF. (2017). Obtenido de <http://indiegames.clickteam.com/>

License Zlib. (2017). Obtenido de Open Source Org: <https://opensource.org/licenses/Zlib>

Linietsky, J. (2016). *GODOT en Español.* Obtenido de <http://godot-doc-en-espanol.readthedocs.io/es/latest/>

López, D. (2017). *Aprende Game Maker.* Obtenido de <http://www.aprendegamemaker.com/>

Lua ORG. (s.f.). Obtenido de <http://www.lua.org/>

Multimedia Fusion Examples. (2008). Obtenido de Castles of Britain: <http://www.castles-of-britain.com/mmfexamples-0.htm>

Peinado, C. C. (s.f.). *Breve historia de motores de juegos.* Obtenido de <https://www.gameit.es/breve-historia-de-motores-de-juegos-episodio-i/>

Repositorio Universidad de Alicante. (2017). Obtenido de <http://rua.ua.es/dspace/handle/10045/36932>

Review Mortal Melon. (2015). Obtenido de Nintendo Life: http://www.nintendolife.com/reviews/wiiu-eshop/mortar_melon

Scirra License. (2017). Obtenido de Scirra: <https://www.scirra.com/store/licensing#license2>

Spalding, S. (2014). *Game Maker Tutoriales.* Obtenido de <http://gammakertutorials.com/>

Store Extension. (2017). Obtenido de Click Team Store: <http://clickstore.clickteam.com/plugins/extensions>

The Lua-MMF2 Combo. (2009). Obtenido de Create Games: <http://www.create-games.com/article.asp?id=1968>

The Spriters Resource. (2000). Obtenido de <https://www.spriters-resource.com/>

Torque 2D: Getting Started 1 . (s.f.). Obtenido de You tube:
<https://www.youtube.com/watch?v=JCHQ1LCn4dk>

True Valhalla. (2017). Obtenido de <https://www.truevalhalla.com/blog>

Turnes, Y. (2013). *Game Dic.* Obtenido de <http://www.gamerdic.es/>

Useful Resources. (2017). Obtenido de Click Team Fusion:
<http://www.clickteam.com/resources>

Se ha encontrado la mejor herramienta de carácter generalista y para todo tipo de usuario, que ha sido Game Maker.

Anexo I: Datos completos encuesta sobre frameworks de creación de videojuegos.

1. Proceso de instalación.

A continuación, se muestran los resultados del proceso de instalación de cada uno de los frameworks.

1.1. Datos instalación Gedvelop.

En la siguiente tabla se muestran los datos de tiempo instalación de los 8 sujetos del estudio y su tiempo medio.

SUJETO	TEMPO (Horas, segundo, minutos)
Sujeto 1	0:06:32
Sujeto 2	0:01:23
Sujeto 3	0:02:42
Sujeto 4	0:05:50
Sujeto 5	0:05:40
Sujeto 6	0:02:40
Sujeto 7	0:06:15
Sujeto 8	0:02:10
Tiempo medio	0:04:09

Tabla 218: Tiempo instalación sujetos estudio.

1.2. Datos instalación Construct 2

En la tabla siguiente se muestra los tiempos de instalación de Construct dos para los 8 sujetos que han participado en la encuesta.

SUJETO	TEMPO (Horas, segundo, minutos)
Sujeto 1	0:02:40
Sujeto 2	0:01:20
Sujeto 3	0:01:40

Sujeto 4	0:02:15
Sujeto 5	0:02:40
Sujeto 6	0:02:10
Sujeto 7	0:03:40
Sujeto 8	0:01:10
Tiempo medio	0:02:12

Tabla 219: Tiempo instalación sujetos Construct 2

1.3. Datos instalación CTF.

En la tabla siguiente se muestra los tiempos de instalación de CTF dos para los 8 sujetos que han participado en la encuesta.

SUJETO	TIEMPO (Horas, segundo, minutos)
Sujeto 1	0:06:50
Sujeto 2	0:03:02
Sujeto 3	0:02:40
Sujeto 4	0:03:13
Sujeto 5	0:05:20
Sujeto 6	0:06:15
Sujeto 7	0:07:20
Sujeto 8	0:02:10
Tiempo medio	0:04:36

Tabla 220: Datos instalación tiempo CTF

1.4. Datos instalación Game Maker.

En la tabla siguiente se muestra los tiempos de instalación de GM dos para los 8 sujetos que han participado en la encuesta.

SUJETO	TEMPO (Horas, segundo, minutos)
Sujeto 1	0:04:10
Sujeto 2	0:04:20
Sujeto 3	0:03:50
Sujeto 4	0:03:50
Sujeto 5	0:05:50
Sujeto 6	0:05:45
Sujeto 7	0:07:50
Sujeto 8	0:02:30
Tiempo medio	0:04:46

Tabla 221: Tiempo instalación GM encuestas

2. Valoración de proceso de instalación.

En el siguiente apartado se exponen los datos de la valoración general del proceso de instalación para los diferentes programas por las personas participantes en el estudio.

2.1. Valoración del proceso de instalación de Gdevelop.

En la tabla siguiente se muestra la valoración de los sujetos del estudio de la sencillez del proceso de instalación de Gdevelop, con una escala de 0 a 10 donde cero es extrema complicación y 10 máxima sencillez.

Sujeto	Valoración sencillez (0-10)
Sujeto 1	10
Sujeto 2	10
Sujeto 3	10
Sujeto 4	8
Sujeto 5	9
Sujeto 6	9
Sujeto 7	8
Sujeto 8	10

Promedio	9,25
----------	------

Tabla 222: Valoración sencillez proceso de instalación Gdevelop

La funcionalidad del proceso de instalación se ha valorado mediante una escala que mide si el programa guía al usuario o no, donde 0 indica que los pasos de la instalación no están bien indicados y que el usuario se ha encontrado perdido durante la instalación y 10 el programa guía de forma correcta y explica detalladamente todos los casos.

Sujeto	Valoración guía en el proceso de instalación (0-10)
Sujeto 1	10
Sujeto 2	10
Sujeto 3	10
Sujeto 4	8
Sujeto 5	9
Sujeto 6	9
Sujeto 7	7
Sujeto 8	9
Puntuación media	9

Tabla 223: Valoración guía proceso de instalación Gdevelop

2.2. Valoración proceso de instalación Construct 2.

Las dos tablas siguientes muestran la valoración de la sencillez del proceso de instalación de Construct 2 y la valoración de si el instalador guía correctamente a los usuarios durante el proceso.

Sujeto	Valoración sencillez (0-10)
Sujeto 1	9
Sujeto 2	8

Sujeto 3	8
Sujeto 4	6
Sujeto 5	8
Sujeto 6	7
Sujeto 7	7
Sujeto 8	9
Puntuación Media	7,75

Tabla 224: Valoración sencillez proceso de instalación Construct 2

Sujeto	Valoración guía en el proceso de instalación (0-10)
Sujeto 1	9
Sujeto 2	10
Sujeto 3	8
Sujeto 4	9
Sujeto 5	7
Sujeto 6	8
Sujeto 7	7
Sujeto 8	9
Puntuación media	8,375

Tabla 225: Valoración guía proceso de instalación Constrcut 2

2.3. Valoración proceso de instalación Click Team Fusion.

Las dos tablas siguientes muestran la valoración de la sencillez del proceso de instalación de CTF y la valoración de si el instalador guía correctamente a los usuarios durante el proceso.

Sujeto	Valoración sencillez (0-10)
---------------	------------------------------------

Sujeto 1	9
Sujeto 2	10
Sujeto 3	10
Sujeto 4	7
Sujeto 5	10
Sujeto 6	9
Sujeto 7	8
Sujeto 8	10
Promedio	9,125

Tabla 226: Valoración sencillez proceso de instalación CTF

Sujeto	Valoración guía en el proceso de instalación (0-10)
Sujeto 1	9
Sujeto 2	10
Sujeto 3	10
Sujeto 4	8
Sujeto 5	10
Sujeto 6	10
Sujeto 7	8
Sujeto 8	10
Puntuación media	9,375

Tabla 227: Valoración guía en el proceso de instalación CTF

2.4. Valoración proceso de instalación Game Maker.

Las dos tablas siguientes muestran la valoración de la sencillez del proceso de instalación de GM y la valoración de si el instalador guía correctamente a los usuarios durante el proceso.

Sujeto	Valoración sencillez (0-10)
Sujeto 1	7
Sujeto 2	7
Sujeto 3	9
Sujeto 4	8
Sujeto 5	7
Sujeto 6	9
Sujeto 7	7
Sujeto 8	9
Puntuación Media	7,875

Tabla 228: Game Maker sencillez proceso de instalación encuestas

Sujeto	Valoración guía en el proceso de instalación (0-10)
Sujeto 1	9
Sujeto 2	10
Sujeto 3	9
Sujeto 4	10
Sujeto 5	8
Sujeto 6	9
Sujeto 7	8
Sujeto 8	9
Puntuación media	9

Tabla 229: Valoración guía proceso de instalación GM encuestas

3. Valoración de la interfaz.

3.1. Valoración interfaz de Gdevelop.

En la tabla siguiente se muestra la valoración de la interfaz en los distintos puntos de la encuesta para el programa Gdevelop.

Concepto	1	2	3	4	5	TOTAL	MEDIA
Rapidez al iniciarse	0	0	0	1	7	39	4,875
El programa se muestra rápido	0	0	0	3	5	37	4,625
La interfaz es atractiva	0	0	2	4	2	32	4
Tiene un aspecto cuidado y moderno	0	1	4	2	1	27	3,375
Puntuación diseño del programa		2	5	1	0	23	2,875
Los iconos representan bien los elementos Y/o funciones	0		4	4	0	28	3,5
A priori parece sencillo de usar	0	1	2	3	2	30	3,75
Consumo de recursos	0	0	4	3	1	29	3,625
TOTAL	0	4	21	21	18	245	30,625
MEDIA							3,828125

Tabla 230: Valoración Interfaz Gdevelop

3.2. Valoración interfaz de Construct 2.

En la tabla siguiente se muestra la valoración de la interfaz en los distintos puntos de la encuesta para el programa Construct 2.

Concepto	1	2	3	4	5	TOTAL	MEDIA
Rapidez al iniciarse	0	0	1	5	2	33	4,125
El programa se muestra rápido	0	0	1	4	3	34	4,25

La interfaz es atractiva	0	1	3	4	0	27	3,375
Tiene un aspecto cuidado y moderno	0	1	2	4	1	29	3,625
Puntuación diseño del programa	0	1	1	6	0	29	3,625
Los iconos representan bien los elementos Y/o funciones	0		2	2	4	34	4,25
A priori parece sencillo de usar	0	1	1	6	0	29	3,625
Consumo de recursos	0	0	3	4	1	30	3,75
TOTAL	0	4	14	35	11	245	30,625
MEDIA							3,828125

Tabla 231: Gdevelop resultado encuesta interfaz

3.3. Valoración interfaz de Click Team Fusion.

En la tabla siguiente se muestra la valoración de la interfaz en los distintos puntos de la encuesta para el programa CTF.

Concepto	1	2	3	4	5	TOTAL	MEDIA
Rapidez al iniciarse	0	2	5	1	0	23	2,875
El programa se muestra rápido	0	0	3	5	0	29	3,625
La interfaz es atractiva	1	3	4	0	0	19	2,375
Tiene un aspecto cuidado y moderno	2	4	2	0	0	16	2

Puntuación diseño del programa	1	4	3	0	0	18	2,25
Los iconos representan bien los elementos Y/o funciones	0	3	4	1	0	22	2,75
A priori parece sencillo de usar	0	2	3	3	0	25	3,125
Consumo de recursos	0	0	3	5	0	29	3,625
TOTAL	4	18	27	15	0	181	22,625
MEDIA							2,828125

Tabla 232: Valoración Interfaz CTF

1.1. Valoración interfaz de Game Maker

En la tabla siguiente se muestra la valoración de la interfaz en los distintos puntos de la encuesta para el programa GM.

Concepto	1	2	3	4	5	TOTAL	MEDIA
Rapidez al iniciarse	0	2	5	1	0	23	2,875
El programa se muestra rápido	0	0	1	7	0	31	3,875
La interfaz es atractiva	0	0	0	5	3	35	4,375
Tiene un aspecto cuidado y moderno	0	0	0	3	5	37	4,625
Puntuación diseño del programa		0	0	4	4	36	4,5
Los iconos representan bien los elementos Y/o funciones	0	0	1	1	6	37	4,625
A priori parece sencillo de usar	0	0	5	2	1	28	3,5

Consumo de recursos	0	0	0	1	7	39	4,875
TOTAL	0	2	12	2 4	2 6	266	33,25
MEDIA							4,15625

Tabla 233: Valoración Interfaz GM encuestadas

Anexo II: Instalación y configuración de SFML en Visual Studio 2017.

1. Compilación de SFML compatible con Visual Studio Instalado.

Para llevar a cabo la instalación de SFML en Visual Studio 2017, se debe instalar previamente el programa Cmake³⁷, para ello lo descargarmos desde la web oficial³⁸.

Una vez con Cmake en el equipo los pasos que habrá que seguir para instalar SFML en el equipo son los siguientes:

- Generar proyecto Visual Studio con SFML vía Cmake.
- Compilar proyecto generado en Visual Studio.

1.1. Generar proyecto Visual Studio vía CMake.

Descargar y descomprimir la última versión de SFML desde su Github³⁹, ya que la descarga desde la web oficial no incluye los archivos necesarios para realizar la compilación de la librería.

Abrir Cmake y en la opción *Where is the source code*, seleccionar SFML y pulsar la opción *configure* y marcar las opciones igual que en la Figura 112.

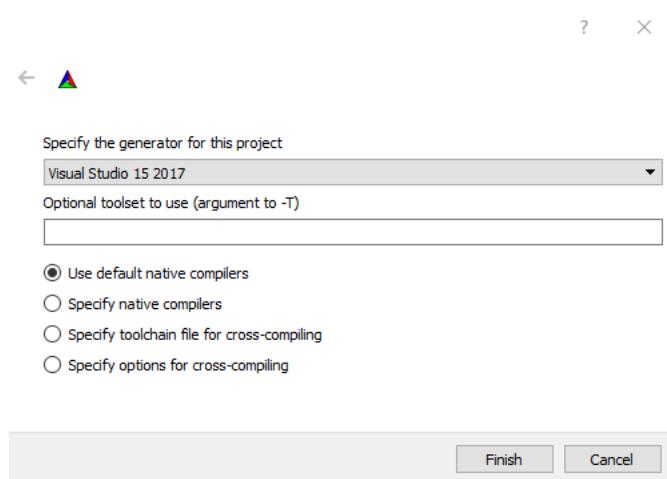


Figura 132: Configuración Cmake

³⁷ Herramienta multiplataforma de generación o automatización de código, diseñada para construir, probar y empaquetar software. CMake se utiliza para controlar el proceso de compilación del software usando ficheros de configuración sencillos e independientes de la plataforma.

³⁸ <https://cmake.org/download/>

³⁹ <https://github.com/SFML/SFML>

Un vez terminada la configuración aparecerá el menú de la Figura 113.

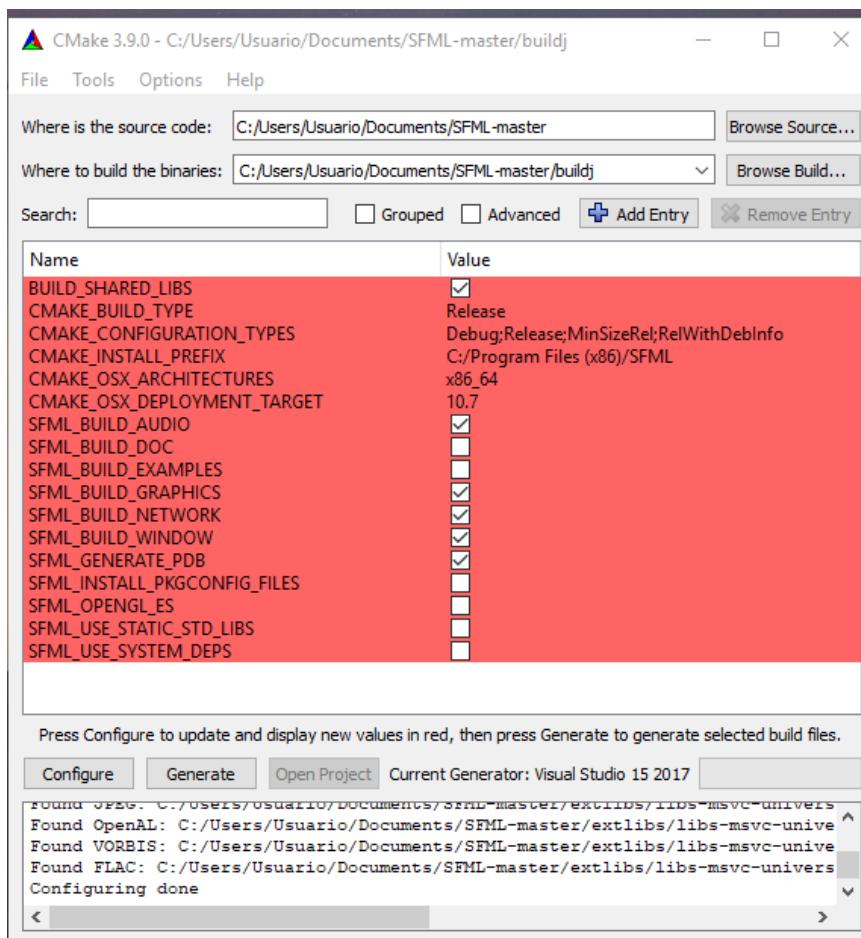


Figura 133: Cmake SFML

Las opciones por defecto que se deben marcar en este menú para realizar la compilación son las siguientes:

- *Where is the build the binaries.*
Archivo donde se va generar la solución Visual Studio.
- *BUILD_SHARED_LIBS.*
- *CMAKE_BUILD_TYPE.*
En esta opción se debe insertar uno de los valores de CMAKE_CONFIGURATION_TYPES. Las dos configuraciones tipo más utilizadas son:
Debug: Se utiliza el entorno de desarrollo, durante la fase de creación y programación del juego se deberá usar esta configuración.
Release: Es la versión ejecutable del SFML, se utiliza una vez el desarrollo ha terminado y se desea compilar la versión final de la solución para uso en otro

equipos.

- *SFML_BUILD_AUDIO.*
Librerías SFML relativas a Audio.
- *SFML_BUILD_GRAPHICS.*
Librerías SFML relativas a graficos.
- *SFML_BUILD_NETWORK.*
Librerías SFML de herramientas online.
- *SFML_BUILD_WINDOWS.*
Librerías SFML relativas a la gestión de ventanas.

El resto de opciones son librerías y elementos accesorios que cada usuario decidirá en cada caso.

Si el proceso ha funcionado correctamente en la carpeta destino se habrá generado un proyecto Visual Studio los siguientes archivos, véase Figura 114.

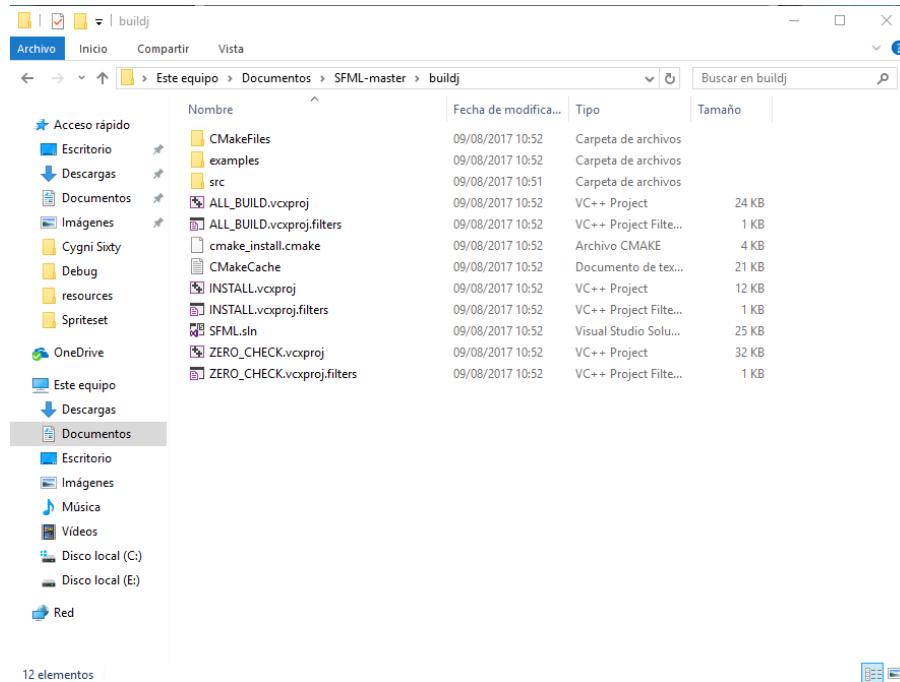


Figura 134: Carpeta generada por Cmake

1.2 Compilar solución generada en Visual Studio.

En la carpeta generada por CMake se debe haber generado una solución Visual Studio SFML.sln.

Habrá que abrir la solución que contendrá todos los archivos SFML, véase Figura 115.

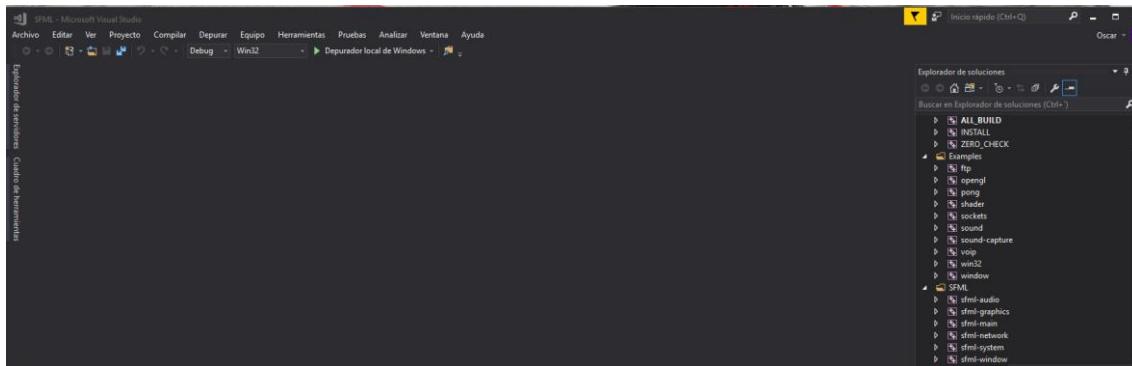


Figura 135: Proyecto Visual Studio.

Con el proyecto abierto configuramos el modo en que se va generar las librerías, véase Figura 116. Este modo debe coincidir con el indicado en *CMAKE_BUILD_TYPE* de la configuración Cmake, en cuanto al sistema de Windows 32 o 64 se deberá seleccionar el mismo con el que se vaya a compilar posteriormente el juego. Una vez configurado este apartado activar la opción *depurador local Windows*.

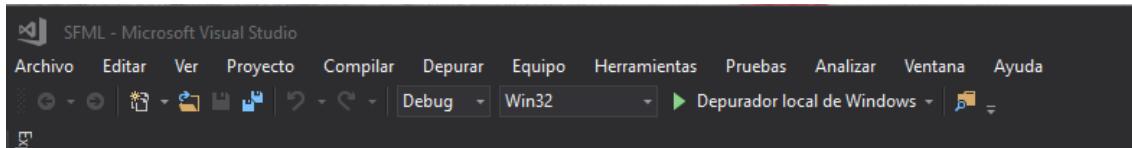


Figura 136: Depurador Visual Studio

Una vez finalizado si el proceso ha funcionado correctamente se habrá generado una carpeta lib y dentro de ésta una Release o Debug (dependiendo el modo que se haya seleccionado previamente) con los archivos lib y dll.

2. Configurar proyecto Visual Studio.

Ahora que ya se tienen las librerías compiladas para el sistema operativo y Visual Studio con el que se está trabajando, habrá que crear un nuevo proyecto Visual Studio y configurarlo de la siguiente forma para trabajar con SFML.

Ir a proyecto y propiedades, véase Figura 117, y configurar las siguientes opciones:

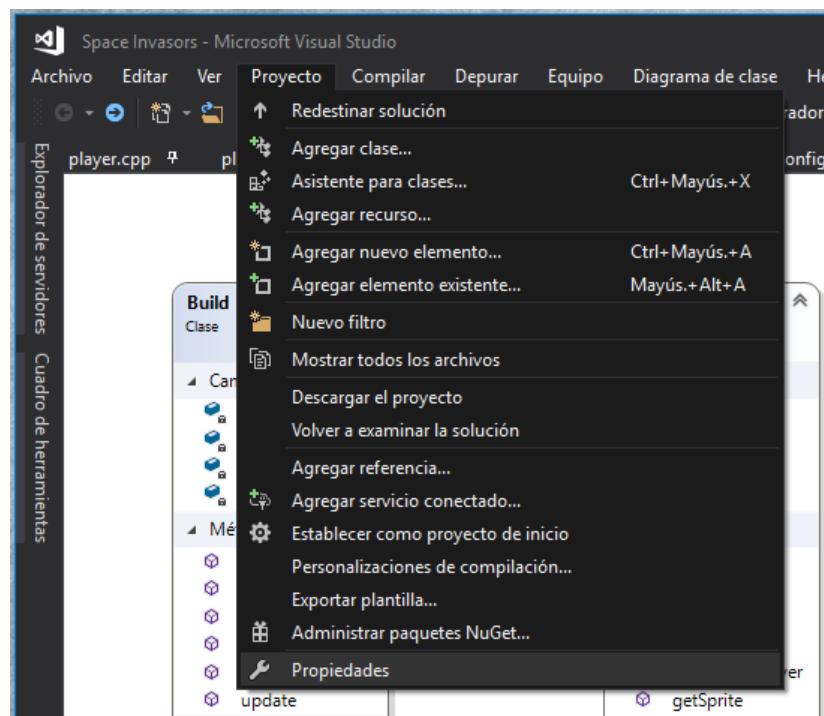


Figura 137: Opciones de proyecto Visua Studio

- Opción C++ , General y archivos de inclusión , Figura 118: Indicar la ruta de la carpeta include de SFML (descargado de github).

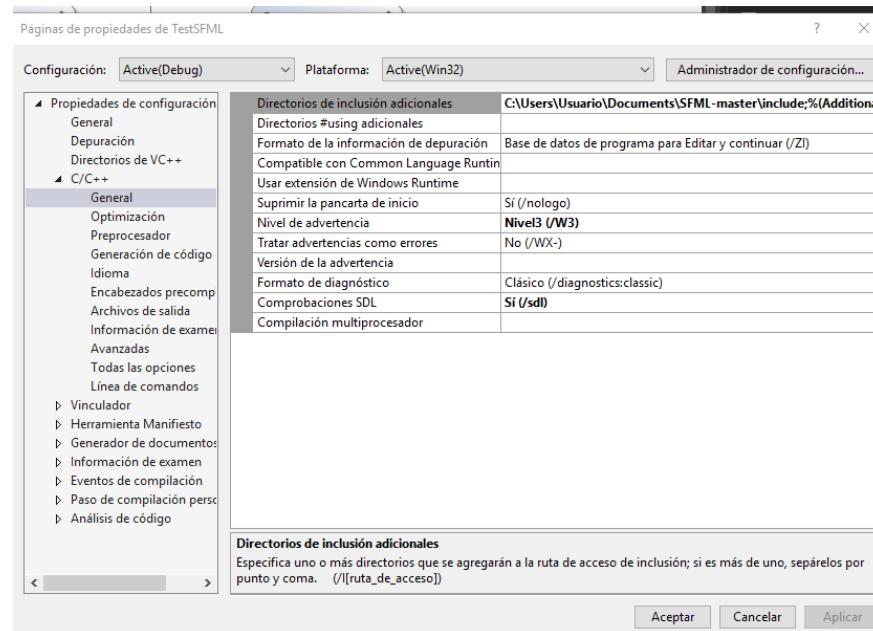


Figura 138: Visual Studio archivos de inclusión

- Viculador, General, y en Directorios de bibliotecas adicionales, véase Figura 119, indicar la ruta la carpeta Relase o Debug, en casa caso, que se ha generado con Cmake y Visual Studio en el paso anterior.

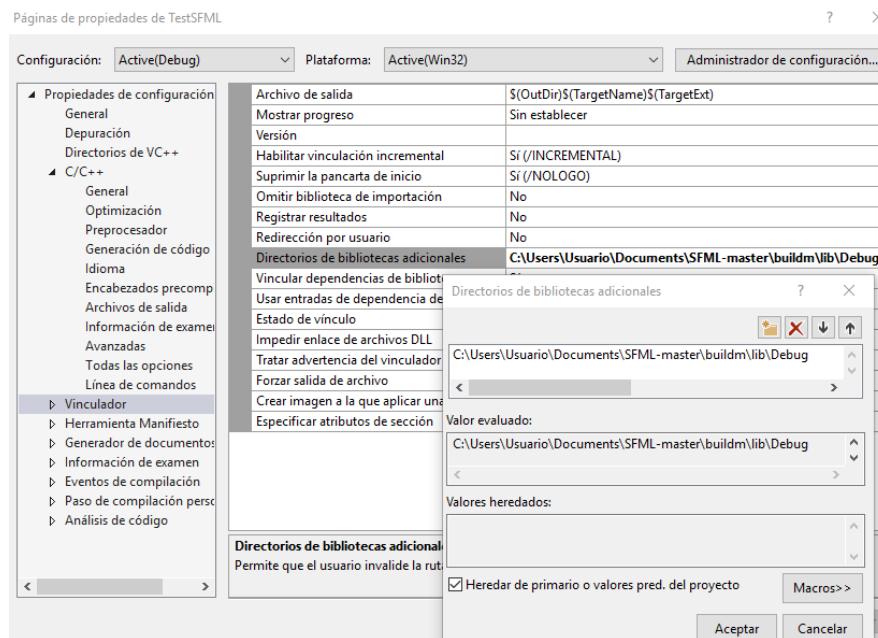


Figura 139: Visual Studio directorio de bibliotecas adicionales

- Vinculador, Entrada y en dependencias, véase Figura 120, adicionales si se trata de SFML en modo debub lo siguiente: sfml-graphics-d.lib, sfml-main-d.lib, sfml-audio-d.lib, sfml-system-d.lib, sfml-window-d.lib. Si se trata de modo Release se añade lo siguiente: sfml-graphics.lib, sfml-main.lib, sfml-audio.lib, sfml-system.lib, sfml-window-d.lib.

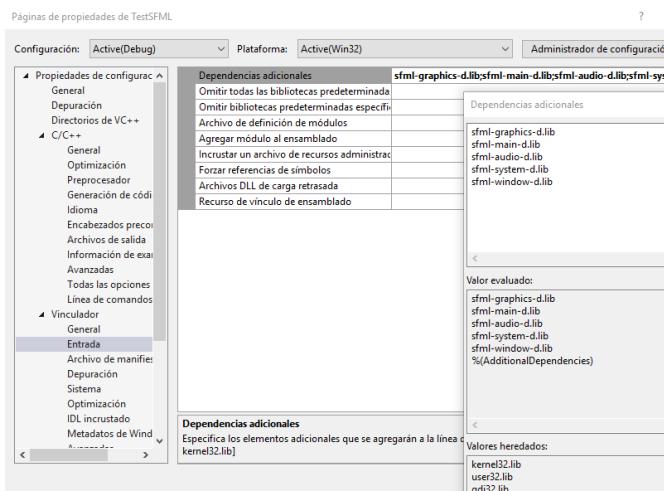


Figura 140: Visual Studio dependencias adicionales

Una vez configuradas las propiedades del proyecto solo quedará incluir los archivos .dll generador por Cmake en la carpeta de Visual Studio.

Una vez realizado el proceso ejecutar el siguiente código para comprobar que SFML esta configurado correctamente.

```
#include <SFML/Graphics.hpp>
int main()
{
    sf::RenderWindow window(sf::VideoMode(200, 200), "SFML works!");
    sf::CircleShape shape(100.f);
    shape.setFillColor(sf::Color::Green);

    while (window.isOpen())
    {
        sf::Event event;
        while (window.pollEvent(event))
        {
            if (event.type == sf::Event::Closed)
                window.close();
        }

        window.clear();
        window.draw(shape);
        window.display();
    }
    return 0;
}
```

Figura 141: Código test SFML

Bibliografía

¿Qué es un kit de desarrollo de software (SDK)? (2013). Obtenido de 4 R Soluciones:

<http://www.4rsoluciones.com/blog/que-es-un-kit-de-desarrollo-de-software-sdk-2/>

2d and 3d in Game Maker Studio. (2015). Obtenido de Like 100 bears:

<http://www.like100bears.com/writing/2d-3d-in-gamemaker-studio>

2D Game Creation. (2017). Obtenido de Unity 3D:

<https://unity3d.com/es/learn/tutorials/s/2d-game-creation>

Adventure Game Studio. (s.f.). Obtenido de <http://www.adventuregamestudio.co.uk/>

Angel 2D. (2013). Obtenido de <https://angel2d.com/>

Angel 2D vs SFML. (2017). Obtenido de Slant:

https://www.slant.co/versus/1059/1100/~angel2d_vs_sfml

Autores, V. (2016). *Programmers Guide.* Obtenido de Cocos 2D X:

<http://www.cocos2d-x.org/docs/ProgrammersGuide.pdf>

Candil, D. (2012). *Cuatro motores gráficos para perder el miedo y lanzarse al desarrollo de videojuegos.* Obtenido de Vida Extra:

<https://www.vidaextra.com/listas/4-motores-graficos-para-perder-el-miedo-y-lanzarse-al-desarrollo-de-videojuegos>

Click Team Fusion 2.5. (2017). Obtenido de <http://www.clickteam.com/clickteam-fusion-2-5>

Click Team Fusion Review. (2017). Obtenido de Slant:

<https://www.slant.co/options/6626/~clickteam-fusion-2-5-review>

Clickteam Fusion 2.5. (2016). Obtenido de Click Wiki:

https://clickwiki.net/wiki/Clickteam_Fusion_2.5

Cocos 2D X. (2017). Obtenido de <http://www.cocos2d-x.org/>

Comunidad Game Maker. (2011). Obtenido de

<http://www.comunidadgm.org/index.php>

Construct 2 vs Game Develop, why nobody using it? (2014). Obtenido de Scirra Forum:

https://www.scirra.com/forum/construct-2-vs-game-develop-why-nobody-using-it_t115815?sid=ebb38308144e95d89880be87f23d776f&start=10

Contact Us. (2014). Obtenido de Air Scapte Game:

<http://airscapetegame.com/index.html#contactus>

Curso de Game Maker. (2017). Obtenido de Edutin: <https://edutin.com/curso-de-GameMaker-y-GML-2018>

Eclipse Origins. (2013). Obtenido de RPG Maker Net:

<https://rpgmaker.net/engines/eclipseor/>

Engine Click Team Fusion. (2016). Obtenido de Taringa:

<http://www.taringa.net/comunidades/creatusjuegos/9178104/Engine-Clickteam-Fusion-2-5-Multimedia-Fusion.html>

Experiencia de juego 3D. (2016). Obtenido de Docs Unity:

<https://docs.unity3d.com/es/current/Manual/Overview2D.html>

Extending Construct 2 with Javascript. (2011). Obtenido de Scirra Forum:

https://www.scirra.com/forum/extending-construct-2-with-javascript_t60269

Extension MMF. (s.f.). Obtenido de Dark Wire: <http://dark-wire.com/store/extlist.php>

Fernández, A. L. (2013). *Curso de creación de videojuegos con Game Maker.* Obtenido de <http://toodaim.blogspot.com.es/2013/02/curso-gamemaker-crea-videojuegos.html>

Game Engine Game Develop. (2011). Obtenido de Taringa:

<https://www.taringa.net/posts/hazlo-tu-mismo/13168821/Game-Engine-Game-Develop.html>

Game Maker blog. (2014). Obtenido de

<http://gammemakerblog.com/2014/06/07/gammemaker-studio-standard-is-now-completely-free/>

Game Maker Vs Unity. (2016). Obtenido de Yises Blog:

<http://yises.com/blog/2016/03/28/unity-vs-gammemaker-cual-escoger/>

Game Showcase. (2016). Obtenido de HTML 5 Game Dev:

<http://www.html5gamedevs.com/forum/8-game-showcase/>

Games dev. (2017). Obtenido de <http://www.gamedev.es/>

Gdevelop change list. (2017). Obtenido de Compile Games Forum:

<http://forum.compilegames.net/viewtopic.php?f=18&t=8532>

Gdevelop Documentation. (2015). Obtenido de Wiki Compile Games:

<http://www.wiki.compilegames.net/doku.php/es/gdevelop/documentation>

Gdevelop Road Map. (2015). Obtenido de Trello:

<https://trello.com/b/qf0IM7k8/gdevelop-roadmap>

Getting Started. (2016). Obtenido de Crosswalk Project: https://crosswalk-project.org/documentation/getting_started.html

GNU Lesser General Public License. (2016). Obtenido de GNU Org:

<https://www.gnu.org/licenses/lgpl.html>

GODOT Engine. (2017). Obtenido de <https://godotengine.org/>

- Guide to Intalling extensions.* (2017). Obtenido de Community Click Team Fusion:
<https://community.clickteam.com/threads/83607-Guide-to-installing-Extensions>
- Gutiérrez, J. J. (s.f.). *¿Qué es un framework web?* . Obtenido de
http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf
- Hektor Programador.* (2016). Obtenido de <http://blog.hcosta.info>
- How Distribute your Games.* (2015). Obtenido de Wiki Compile Games:
<http://wiki.compilgames.net/doku.php/gdevelop/tutorials/howtodistribute>
- Indie Games CTF.* (2017). Obtenido de <http://indiegames.clickteam.com/>
- License Zlib.* (2017). Obtenido de Open Source Org:
<https://opensource.org/licenses/Zlib>
- Linietsky, J. (2016). *GODOT en Español.* Obtenido de <http://godot-doc-en-espanol.readthedocs.io/es/latest/>
- López, D. (2017). *Aprende Game Maker.* Obtenido de
<http://www.aprendegamemaker.com/>
- Lua ORG.* (s.f.). Obtenido de <http://www.lua.org/>
- Multimedia Fusion Examples.* (2008). Obtenido de Castles of Britain:
<http://www.castles-of-britain.com/mmfexamples-0.htm>
- Peinado, C. C. (s.f.). *Breve historia de motores de juegos.* Obtenido de
<https://www.gameit.es/breve-historia-de-motores-de-juegos-episodio-i/>
- Repositorio Universidad de Alicante.* (2017). Obtenido de
<http://rua.ua.es/dspace/handle/10045/36932>
- Review Mortal Melon.* (2015). Obtenido de Nintendo Life:
http://www.nintendolife.com/reviews/wiiu-eshop/mortar_melon
- Scirra License.* (2017). Obtenido de Scirra:
<https://www.scirra.com/store/licensing#license2>
- Spalding, S. (2014). *Game Maker Tutoriales.* Obtenido de
<http://gammakertutorials.com/>
- Store Extension.* (2017). Obtenido de Click Team Store:
<http://clickstore.clickteam.com/plugins/extensions>
- The Lua-MMF2 Combo.* (2009). Obtenido de Create Games: <http://www.create-games.com/article.asp?id=1968>
- The Spriters Resource.* (2000). Obtenido de <https://www.spriters-resource.com/>
- Torque 2D: Getting Started 1 .* (s.f.). Obtenido de You tube:
<https://www.youtube.com/watch?v=JCHQ1LCn4dk>

True Valhalla. (2017). Obtenido de <https://www.truevalhalla.com/blog>

Turnes, Y. (2013). *Game Dic.* Obtenido de <http://www.gamerdic.es/>

Useful Resources. (2017). Obtenido de Click Team Fusion:

<http://www.clickteam.com/resources>

