



UNIVERSIDAD
DE MÁLAGA



Realizado por Carlos Valencia Arguilea

Tutorizado por Antonio J. Fernández Leiva

Departamento de LCC

MÁLAGA, febrero de 2020



UNIVERSIDAD
DE MÁLAGA





UNIVERSIDAD
DE MÁLAGA





UNIVERSIDAD
DE MÁLAGA



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
GRADUADO EN INGENIERÍA DEL SOFTWARE

**DESARROLLO DE UN VIDEOJUEGO SERIO
COMO HERRAMIENTA PARA EL ANÁLISIS DE LA
CAPACIDAD DE TRABAJO EN EQUIPO DE
ESCOLARES**

**DEVELOPMENT OF A SERIOUS VIDEO GAME
AS A TOOL FOR THE ANALYSIS OF TEAMWORK
CAPACITY OF SCHOOLCHILDREN**

Realizado por
Carlos Valencia Arguilea
Tutorizado por
Antonio José Fernández Leiva
Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, FEBRERO DE 2020



UNIVERSIDAD
DE MÁLAGA



Resumen

El propósito de este Trabajo de Fin de Grado ha sido crear un videojuego serio que sirva como herramienta para enseñar y analizar las capacidades de trabajo en equipo, principalmente a grupos de escolares.

El proyecto se basa en un juego plataformas compuesto por niveles, en el que se requiere la interacción coordinada de varios jugadores para su resolución. Un agente externo evaluará distintos aspectos del trabajo en equipo durante el desarrollo del juego, para finalmente, mediante un breve cuestionario, obtener los informes de rendimiento del equipo y capacidades mostradas.

El objetivo es emplear el potencial de los juegos serios para sacar a la luz las capacidades que tienen los jugadores; ejerciendo como una herramienta de dinámicas de equipo y proporcionando informes psicológicos a partir del comportamiento del grupo.

Palabras clave: Videojuego, juego serio, trabajo en equipo, recurso docente, juego de plataformas.

Abstract

The purpose of this Final Degree Project has been to create a serious videogame that serves as a tool to teach and analyse the abilities of teamwork, mainly to school groups.

The project is based on a platform game composed set of levels, in which the coordinated interaction of several players is required for the resolution. An external agent evaluates different aspects of teamwork during the game, and finally, through a brief questionnaire, obtain reports on team performance and skills shown.

The objective is to use the potential of serious games to show up the capabilities that players have, exercising as a tool for team dynamics and providing psychological reports based on the group behaviour.

Keywords: Videogame, serious game, teamwork, scholar resource, platform game.

Índice

Resumen.....	1
Abstract.....	1
Índice.....	1
Índice de Figuras.....	3
Introducción	7
1.1 Ámbito	7
1.2 Objetivos	7
1.3 Motivación	8
1.4 Estructura de la memoria	8
Antecedentes	11
2.1 Desarrollo de videojuegos	11
2.2 Entorno tecnológico	11
2.2.1 Unity	12
2.3 Metodología de trabajo.....	13
2.3.1 Scrum.....	13
2.3.2 Roles Scrum	13
2.3.3 Recursos Scrum.....	14
2.4 Conceptos.....	14
Planificación y diseño del videojuego.....	15
3.1 Estructura del proyecto	15
3.1.1 Concepto	15
3.1.2 Preproducción	15
3.1.3 Producción.....	15
3.1.4 Posproducción	15
3.2 Planificación	16
3.3 Recursos	16
3.3.1 Recursos hardware	16
3.3.2 Recursos software	17
3.3.3 Recursos Humanos	17
3.4 Análisis	18
3.4.1 Actores.....	18
3.4.2 Objetivos.....	19
3.4.3 Requisitos del Sistema	20
3.4.4 Matriz de trazabilidad de requisitos	26
3.5 Modelo.....	27
3.6 Diseño	30
3.6.1 Patrón MVC	30
3.6.2 Documento de Diseño	30
3.6.3 Diseño Técnico.....	30
3.6.4 Diseño Artístico.....	32
Desarrollo del videojuego	35
4.1 Versiones del GDD	35
4.2 El producto final.....	37
4.2.1 Propósito	37
4.2.2 La atmósfera	37

4.2.3 La historia	37
4.2.4 La perspectiva	37
4.2.5 Personaje	37
4.2.6 El flujo de juego	39
4.3 Manual de usuario	42
4.3.1 Concepto	42
4.3.2 Características	42
4.3.3 Las reglas de juego.....	43
4.3.4 Interacción	43
4.3.5 La mecánica de juego.....	43
4.3.6 Los retos	44
4.4 Pruebas	44
4.4.1 Pruebas del Personaje	45
4.4.2 Pruebas de Escenas.....	45
4.4.3 Pruebas de Interfaz.....	46
4.4.4 Pruebas de Integración	47
4.6 Calidad de Software	48
4.6.1 Trazabilidad de requisitos	48
4.6.2 Validación	48
Conclusiones.....	49
5.1 Problemas encontrados.....	49
5.1.1 Cambios en el concepto del juego	49
5.1.2 Recursos educativos	49
5.2 Ampliaciones	49
5.2.1 Desarrollo de fases	50
5.2.2 Desarrollo de NPCs y agentes autónomos	50
5.2.3 Dinámicas de personaje.....	50
Bibliografía y referencias	51
Manual de Instalación	57
Guía de instalación	57
Requisitos mínimos	57
Recursos Producidos	59
Documento de Diseño del Videojuego	63

Índice de Figuras

Figura 1. Módulos Unity	12
Figura 2. Conceptos	14
Figura 3. Planificación Pykels	16
Figura 4. Planificación Preproducción	16
Figura 5. Planificación Producción	16
Figura 6. Planificación Posproducción	16
Figura 7. Ejemplo Tabla Actor	18
Figura 8. ACT-00 Game Party	18
Figura 9. ACT-01 Player 1	18
Figura 10. ACT-02 Player 2	18
Figura 11. ACT-03 Player 3	18
Figura 12. Ejemplo Tabla Objetivo	19
Figura 13. OBJ-00 Pykels	19
Figura 14. OBJ-01 Estructura del Videojuego	19
Figura 15. OBJ-02 Implementación de Pykels	19
Figura 16. OBJ-03 Creación del Entorno Gráfico	19
Figura 17. OBJ-04 Creación del Sonido de Juego	19
Figura 18. OBJ-05 Creación de la GUI	20
Figura 19. RF-01 Cámara de Juego	20
Figura 20. RF-01 Cámara de Juego	20
Figura 11. RF-02 Cámara Sigue al Personaje	20
Figura 22. RF-03 Límites de Cámara	21
Figura 23. RF-04 Cámara Efecto Parallax	21
Figura 24. RF-05 Físicas de Juego	21
Figura 25. RF-06 Control del Personaje	21
Figura 26. RF-07 Movimiento Horizontal	21
Figura 27. RF-08 Movimiento Vertical	22
Figura 28. RF-09 Control de Plataformas	22
Figura 29. RF-10 Control de Ratón	22
Figura 30. RF-11 Plataforma con Movimiento Vertical	22
Figura 31. RF-12 Plataforma con Solidez Activable	22
Figura 32. RF-13 Plataforma Giratoria	23
Figura 33. RF-14 Hojas Cayendo	23
Figura 34. RF-15 Nubes	23
Figura 35. RF-15 Portales	23
Figura 36. RF-16 Transición Entre Niveles	23
Figura 37. RF-17 límites de Nivel	24
Figura 38. RF-18 Pausar el Juego	24
Figura 39. RF-19 Reanudar el Juego	24
Figura 40. RF-20 Menú de Inicio	24
Figura 41. RF-21 Menú de Pausa	24
Figura 42. RF-22 Menú de Opciones	25
Figura 43. RF-23 Tutorial	25
Figura 44. RF-24 Nivel 1	25
Figura 45. RF-25 Nivel 2	25

Figura 46. RF-26 Nivel 3	25
Figura 47. NRF-01 Jugabilidad	26
Figura 48. NRF-02 Rendimiento	26
Figura 49. NRF-03 Compatibilidad	26
Figura 50. Matriz de Trazabilidad de Requisitos	27
Figura 51. Módulos del Sistema	28
Figura 52. Modelo del Personaje	28
Figura 53. Modelo de Escenas	29
Figura 54. Modelo GUI	29
Figura 55. Audio Manager	31
Figura 56. GUI-Menu Principal	33
Figura 57. GUI-Menu de Pausa	33
Figura 58. Pykel Walk Sprites	33
Figura 59. Pykel Idle Sprites	33
Figura 60. Tree-1	33
Figura 61. Tree-2	33
Figura 62. Bush-1	33
Figura 63. Rock-1	33
Figura 64. Platform-I	33
Figura 67. Platform-L	33
Figura 66. Platform-O	33
Figura 65. Platform-V	33
Figura 68. Block-1	34
Figura 69. Block-2	34
Figura 70. Suelo Principal	34
Figura 71. Pykel Colliders	38
Figura 72. Pykel RigidBody	38
Figura 73. PykelController Script	39
Figura 74. Pykel State Machine	39
Figura 79. Tutorial-4	40
Figura 78. Tutorial-4	40
Figura 77. Tutorial-3	40
Figura 76. Tutorial-2	40
Figura 75. Tutorial-1	40
Figura 83. Nivel1-4	41
Figura 82. Nivel1-3	41
Figura 81. Nivel1-2	41
Figura 80. Nivel1-1	41
Figura 86. Nivel1-7	41
Figura 85. Nivel1-6	41
Figura 84. Nivel1-5	41
Figura 87. Tabla Interacción	43
Figura 88. Ejemplo Tabla Caso de Prueba	45
Figura 89. TC-01 Pruebas de Físicas del Personaje	45
Figura 90. TC-02 Pruebas de Control del Personaje	45
Figura 91. TC-03 Pruebas de Animaciones del Personaje	45
Figura 92. TC-04 Pruebas de Triggers del Personaje	45
Figura 93. TC-05 Pruebas de Control de Plataformas	46

Figura 94. TC-06 Pruebas de Control del Ratón	46
Figura 95. TC-07 Pruebas de Controlador de Escenarios	46
Figura 96. TC-08 Pruebas de Transición de Escenas	46
Figura 97. TC-09 Pruebas de Cámaras de Juego	46
Figura 98. TC-10 Pruebas del Menú de Inicio	47
Figura 99. TC-11 Pruebas del Menú de Opciones	47
Figura 100. TC-12 Pruebas del Menú de Pausa	47
Figura 101. TC-13 Pruebas de Controlador de Audio	47
Figura 102. TC-14 Pruebas de Flujo del Juego	47
Figura 103. TC-15 Pruebas de Configuración del Juego	48
Figura 104. Paquete Prefabs-1	59
Figura 105. Paquete Prefabs-2	59
Figura 106. Paquete Prefabs-3	59
Figura 107. Paquete Scenes	59
Figura 108. Paquete Sprites-1	59
Figura 109. Paquete Sprites-2	60
Figura 110. Paquete Animations-1	60
Figura 111. Paquete Animations-2	60
Figura 112. Paquete Audio-1	60
Figura 113. Paquete Audio-2	60
Figura 114. Paquete GUI-1	61
Figura 115. Paquete GUI-2	61
Figura 116. Paquete GUI-3	61
Figura 117. Paquete Materials	61
Figura 118. Paquete PhysicMaterials	61
Figura 119. Paquete Particles	61
Figura 120. Paquete Scripts	62
GDD-Game design	64
GDD-Pykel Design	64
GDD-Flubber reference	64
GDD-Zac reference	64
GDD-Limbo ingame screenshot	64
GDD-Hollow Knight ingame screenshot	64
GDD-Interacción de Jugadores	64
GDD-Menú de Pausa	64
GDD-Menú Principal	64
GDD-Idle Sprites	64
GDD-Walk Sprites	64
GDD-Tree-2	64
GDD-Tree-1	64
GDD-Rock-1	64
GDD-Bush-1	64
GDD-Platform-L	64
GDD-Platform-O	64
GDD-Platform-V	64
GDD-Platform-I	64
GDD-Block-2	64
GDD-Block-1	64

GDD-Pykel Animator	64
GDD-Audio Manager	64

1

Introducción

En este capítulo se introduce el contexto del trabajo, y se explica la motivación y los objetivos del proyecto.

1.1 Ámbito

La evolución de los videojuegos en los últimos años ha sido tal, que ya ha superado en ingresos a industrias como el cine o la música juntas, a nivel mundial. Además, en España, el videojuego está reconocido como industria cultural por el Parlamento español desde el año 2009 [AEVI, 2020]. Esto provoca una tremenda expansión en todos los campos que conforman la creación de videojuegos, como pueden ser el arte, la música y por supuesto, la informática. Esta evolución también ha afectado al enfoque del videojuego, pasando de ser únicamente un recurso recreativo, a adoptar aplicaciones en ciencia, educación, estudios sociales, investigación, etc. Un ejemplo de esto es el campo referente a los juegos serios, que consisten en videojuegos implementados con un objetivo más allá del entretenimiento, como pueden ser, fines docentes, motivación de aprendizaje, análisis psicológicos, etc.

Por otra parte, cada vez más se están asociando los videojuegos con el trabajo en trabajo en equipo, ya que el planteamiento de retos y objetivos afrontados de manera colectiva sacan a la luz muchas de nuestras capacidades y debilidades a la hora de pensar y trabajar en grupo. Es por ello, que muchas empresas ya están usando los videojuegos como dinámicas de trabajo en equipo para fortalecer y analizar las capacidades que pueden ofrecer sus empleados en esta disciplina.

1.2 Objetivos

En este contexto, el proyecto que aquí se propone busca desarrollar, desde cero, un videojuego serio 2d [Videojuego serio, 2020], orientado a evaluar y mejorar la capacidad de trabajo en equipo de alumnos/as en edad escolar. El juego se dirige idealmente a alumnos en aulas docentes, donde se puedan formar varios equipos de tres, y supervisados por un profesor o tutor; pero este contexto es adaptable a cualquier situación.

El objetivo es que se superen los retos del videojuego de manera colectiva, para completar una secuencia de escenarios virtuales. Cada nivel está pensado para destacar principalmente una capacidad individual del trabajo en equipo (coordinación, cohesión, comunicación, resolución de conflictos, liderazgo, investigación, etc.), aunque el objetivo es evaluar todas las capacidades que muestran los integrantes de un equipo en cada reto. Para ello, durante el desarrollo del juego, el evaluador (profesor, padre o

tutor) debe observar el comportamiento del grupo, y al finalizar el juego realizar un cuestionario digital en el que se evalúan distintos aspectos tanto individuales como colectivos del equipo. De esta manera, una vez finalizados el juego y el cuestionario del evaluador, se genera un informe asociado al rendimiento del equipo y a las capacidades mostradas por cada integrante y a nivel colectivo.

A través de este videojuego se pretende introducir las dinámicas de equipo en entornos escolares, obteniendo grandes beneficios como la temprana familiarización con el trabajo en equipo, o el análisis de las capacidades de trabajo en equipo que se presentan a distintas edades. Pero el alcance de cualquier videojuego con esta orientación es mucho mayor, ya que se puede implementar en procesos de selección de personal cualificado para trabajar en equipo, generando una dinámica que juego que permitiría a los entrevistadores a evaluar las capacidades de manera muy eficiente. Por otro lado, este tipo de videojuego puede ayudar a tratar problemas de exclusión social, autismo, etc.

El videojuego debe ser accesible públicamente y de manera gratuita, en la medida de lo posible, de manera que pueda ser libremente utilizado como recurso pedagógico en instituciones docentes o como referencia para el desarrollo de proyectos del género. Se pretende distribuir el software bajo licencia MIT [[Opensource, 2020](#)].

1.3 Motivación

Los videojuegos han constituido una parte importante de mi vida, desde el primer *Tomb Raider* [[Tomb Raider I, 2020](#)] de la saga, hasta la actualidad, he vivido la evolución de esta industria audiovisual. Gracias a los videojuegos he aprendido idiomas, a competir, trabajar en equipo, he conocido amigos, he aprendido de tecnología, y muchas otras cosas. Me han marcado y son mis influencias juegos como *Kingdom Hearts* [[Kingdom Hearts, 2020](#)], *Tibia* [[Tibia, 2020](#)], *Witcher 3* [[Witcher 3, 2020](#)] o *Resident Evil 4* [[Resident Evil 4, 2020](#)], y actualmente busco referencia en el género *indie* [[Videojuegos Indie, 2020](#)] con videojuegos como *Limbo* [[Limbo, 2020](#)], *Inside* [[Inside, 2020](#)], *Binding of Isaac* [[Binding of Isaac, 2020](#)], o *Super Meat Boy* [[Super Meat Boy, 2020](#)].

Desde joven ya mostraba curiosidad por el proceso de creación de videojuegos, generándome la motivación necesaria para investigar e iniciarme en el mundo tecnológico, con mis primeros pasos programando una versión del juego *Mastermind* [[Mastermind, 2020](#)] en C++ [[C++, 2020](#)]. Con los años mi interés por el desarrollo software fue creciendo, hasta terminar estudiando un grado en informática, para el que actualmente desarrollo este proyecto de fin de carrera.

1.4 Estructura de la memoria

A continuación se describe la estructura de este documento así como el contenido asociado a cada uno de los capítulos.

Introducción: En la introducción se acerca el contexto del trabajo al lector, desde una perspectiva general que se centra progresivamente en el desarrollo de videojuegos

serios. El contenido del capítulo se distribuye en el ámbito del trabajo, los objetivos que se pretende alcanzar y la motivación de realizar el proyecto.

Antecedentes: En los antecedentes se sitúa al lector bajo el contexto del desarrollo de videojuegos, y el entorno tecnológico requerido que demanda dicho desarrollo. Se presta especial interés a la metodología de trabajo y al motor gráfico que se va a implementar.

Planificación y diseño del videojuego: En este capítulo se define el proyecto estableciendo la planificación, recursos y estructura del mismo. Posteriormente se detalla el análisis del software realizado así como los resultados y recursos que proporciona esta etapa del desarrollo.

Desarrollo del videojuego: En el capítulo de desarrollo del videojuego se concreta el resultado del producto final y el proceso llevado a cabo para su obtención; se detallan las características del juego, funcionalidades implementadas, y se explican bajo el contexto del juego componentes importantes como la historia y los personajes.

Conclusiones: El capítulo de conclusiones resume el trabajo realizado desde una visión analítica, evaluando posibles mejoras y detallando algunos problemas relevantes afrontados durante el transcurso del desarrollo. Finalmente se comenta el alcance del videojuego y posibles contextos de uso a los que iba orientado.

Anexos: Al final de la memoria se adjunta, en la sección de apéndices, todos los documentos creados durante el desarrollo del proyecto.

2

Antecedentes

En este capítulo nos centramos en presentar algunos conceptos e información relevante que el lector debe conocer para entender el desarrollo del proyecto. Dentro de esta información se presta especial atención a la herramienta Unity3D [[Unity3D, 2020](#)], y a la metodología *Scrum* [[Scrum, 2020](#)]; las cuales constituyen partes esenciales del desarrollo de este trabajo.

2.1 Desarrollo de videojuegos

El desarrollo de videojuegos, como muchos otros productos software se ubica dentro del ámbito de procesos de desarrollo software, pero este proceso ha ido adaptándose a la evolución de la industria para conseguir potenciar todas las partes implicadas en proyectos de videojuegos.

En este sentido, la evolución en el desarrollo de videojuegos ha quedado diferenciada por dos grandes grupos. El desarrollo de videojuegos AAA referencia a los proyectos de alto despliegue económico, normalmente procedente de grandes compañías o continuaciones de títulos ya exitosos. Por otro lado, el desarrollo de videojuegos *Indie* se asocia al caso opuesto, en el que pequeños equipos de desarrollo sin gran disponibilidad de recursos producen videojuegos atractivos para el público.

En el contexto de este trabajo, se han tomado referencias y guías asociadas al género *Indie*, ya que por motivo de estar siendo desarrollado individualmente y en el ámbito de trabajo de fin de carrera es la orientación más acorde para el desarrollo de *Pykels*.

2.2 Entorno tecnológico

Técnicamente el desarrollo del proyecto demanda el empleo de un motor profesional de desarrollo de videojuegos complementado a su vez con el lenguaje de codificación. Tras investigar sobre las distintas opciones de motores gráficos se tomó la decisión de utilizar Unity 3D, bajo el lenguaje C# [[Csharp, 2020](#)]. También se requiere el empleo de software para el modelado del sistema y la elaboración de diagramas, tales como *Magic Draw* [[Magic Draw, 2020](#)] o *Draw.io* [[Draw.io, 2020](#)]. De manera adicional, se utilizará software destinado a los recursos artísticos del videojuego, como puede ser *Photoshop* [[Creative Cloud, 2020](#)] o *Krita* [[Krita, 2020](#)]. Para la gestión del proyecto se utilizan las herramientas Microsoft Project 2016 [[MS Proyect, 2020](#)] y *Trello* [[Trello, 2020](#)].

2.2.1 Unity

Unity es un motor gráfico y la herramienta de desarrollo de videojuegos más extendida. En esta sección se describe la herramienta Unity3D utilizada para el desarrollo de *Pykels*.

Módulos Unity

El motor gráfico de Unity se compone de los siguientes módulos, tal y como se muestra en la *Figura 49*. Nótese que la orientación del desarrollo con Unity se adapta tanto a sistemas 2D como 3D; destinando diferentes tipos de componentes y configuraciones para cada perspectiva.

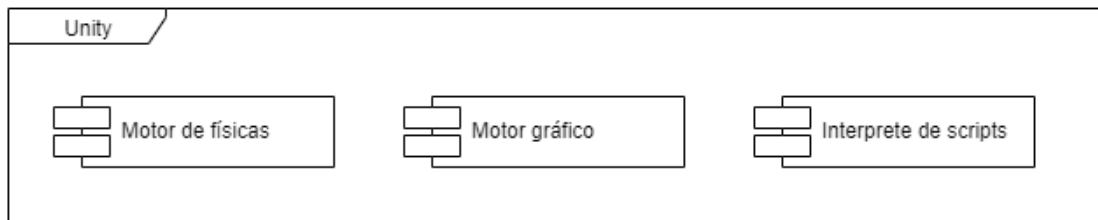


FIGURA 1. MÓDULOS UNITY

Motor gráfico: es el módulo destinado al procesamiento de imágenes del juego.

Motor de físicas: es el módulo destinado a la gestión de físicas del juego.

Interprete de scripts: es el módulo destinado al procesamiento de controladores del juego.

Componentes Unity

Unity proporciona una gran cantidad de componentes y extensiones para el desarrollo de videojuegos. Se detallan a continuación los relevantes para el desarrollo de éste proyecto.

Generales

- *GameObject*: Todos los componentes extienden este objeto.
- *Transform*: Registro de posición, rotación y escala del objeto.

Gráficos

- *SpriteRender*: Permite asociar una imagen al objeto.
- *SpriteMask*: Proporciona interacción dinámica al procesamiento de imágenes.
- *ParticleSystem*: Gestión de partículas para efectos gráficos.

Animaciones

- *Animator*: Estructura de animaciones de objeto configurable.
- *Animation*: Animación individual compuesta por *sprites* en intervalo de tiempo.

Audio

- *AudioMixerGroup*: Administrador de sonidos del juego.
- *AudioListener*: Receptor de sonido dentro de la escena de juego.
- *Sound*: Componente básico de sonido del juego.

Físicas

- *RigidBody2D*: Dota de físicas dinámicas al objeto con masa, gravedad, etc.
- *Collider*: Proporciona al objeto una estructura para colisiones.

- *CompositeCollider*: Estructura de composición de *colliders*.
- *CircleCollider2D*: Estructura de colisión circular y bidimensional.
- *BoxCollider*: Estructura de colisión cuadrada y bidimensional.

Scripts: Permiten definir clases para el control del juego.

GUI

- *Canvas*: Componente superior en cada interfaz.
- *Slider*: Barra deslizable.
- *Button*: Botón.
- *Toggle*: Cuadro de activación.
- *Image*: imagen.

Extensiones

- *Cinemachine*: Implementa cámaras virtuales.
- *CinemachineConfiner*: Establece los límites de la cámara virtual.
- *TextMeshPro*: Proporciona componentes para la interfaz de usuario.

2.3 Metodología de trabajo

La metodología elegida para el proyecto *Pykels* ha sido *Scrum*, que se basa en el desarrollo incremental ágil [[Metodologías agile, 2020](#)]. El tutor ejercerá el papel de *Scrum Master* realizándose reuniones cada dos semanas, en un proceso de desarrollo iterativo.

2.3.1 Scrum

Scrum es una metodología de trabajo ágil, que consiste en la gestión de proyectos enfocados al cliente. Al contrario de la metodología en cascada (*waterfall*) [[Waterfall VS agile, 2020](#)], las metodologías ágiles ejecutan varias iteraciones del ciclo de desarrollo, dividiendo el trabajo en pequeñas funcionalidades que desarrollar y que el cliente podrá validar.

2.3.2 Roles Scrum

Product Owner

Representante del cliente, proporciona la visión del proyecto y la información necesaria para la obtención de requisitos. Durante los procesos de calidad, el *product owner* se implicará en la validación del producto, funcionalidades, requisitos, etc.

Scrum Master

El *Scrum Master* es el encargado de ayudar al equipo a aplicar la metodología y alcanzar los objetivos del proyecto. Esto implica una responsabilidad a la hora de orientar al equipo y eliminar los impedimentos que vayan surgiendo. Éste rol difiere del jefe de proyecto, y deben ser personas distintas.

Scrum Team

Equipo de profesionales con conocimientos técnicos encargados de la ejecución del trabajo de desarrollo del proyecto.

2.3.3 Recursos Scrum

Sprint

Los *sprints* son períodos cortos de tiempo en los que se deben realizar tareas específicas del desarrollo del proyecto. Son precedidos por reuniones en las que se planifican y distribuyen dichas tareas.

User Stories

Las historias de usuario (*User Stories*) se asocian al análisis del sistema, con el objetivo de proporcionar información sobre una funcionalidad específica desde el punto de vista del usuario. Este recurso es de gran utilidad a la hora de obtener los requisitos, ya que las historias de usuario no utilizan lenguaje técnico y se obtienen directamente del *product owner*.

Product Backlog

Este documento vivo asociado a la metodología *Scrum* se utiliza para representar la lista de los requisitos que quedan por implementar para completar el sistema, ordenados por la prioridad del cliente. En cada sprint del desarrollo se utilizará el *product backlog* para seleccionar las características más importantes que se van a desarrollar en dicha franja temporal. Al finalizar la implementación e integración de cada requisito del *product backlog* se verificará con el *product owner* y en caso de ser satisfactoria se dará por completa la tarea eliminándola del documento vivo.

2.4 Conceptos

En éste apartado se definen algunos de los términos, acrónimos y conceptos asociados a los videojuegos y el desarrollo software.

Término	Definición
AAA	Videojuegos de alto presupuesto con desarrollos extensos
Indie	Videojuegos de bajo presupuesto o equipos pequeños
FPS	Ratio de fotogramas por segundo (Frames Per Second)
Serious Game	Juego con objetivos más amplios que el entretenimiento
Educational Game	Juego con orientación escolar o educativa
Social game	Juego con implicaciones sociales
F2P	Categoría referente a juegos gratuitos (Free To Play)
GUI	Interfaz Gráfica de Usuario, también (Graphic Users Inteface)
HUD	Sobreposiciones gráficas informativas en la pantalla de juego
Multiplayer	Género de juegos multijugador
Party	Equipo de jugadores
Sidescroller	Género de videojuego 2D de vista lateral con fondo parallax
Parallax Background	Fondo por capas con deslizamiento lateral
Sprite	Recurso gráficos inanimados del juego
Assets	Recursos del juego
Canvas	Componente padre de una interfaz gráfica de usuario
Frame	Fotograma de juego

FIGURA 2. CONCEPTOS

3

Planificación y diseño del videojuego

En este capítulo se describen la planificación, estructura y recursos del proyecto, así como el análisis y diseño del videojuego realizado.

3.1 Estructura del proyecto

El proyecto asociado al videojuego serio *Pykels*, el cual constituye el producto final del desarrollo, se organiza en las siguientes fases.

3.1.1 Concepto

La fase inicial del proyecto constituye la definición de la idea del juego, para ello se realiza un borrador del documento de diseño (GDD) [[Game Design Document, 2020](#)] dónde se establezcan el concepto del juego (género, mecánicas, objetivos, estilo artístico, etc.). Como ejercicio de apoyo a este proceso se utiliza la técnica de *brainstorming* [[Brainstorm, 2020](#)], seleccionando un conjunto de hashtags que se asocien a la idea del juego. Con la idea definida se realiza una recopilación de referencias en los distintos campos, como personajes, escenarios, géneros, interfaces, etc.

3.1.2 Preproducción

En la preproducción se establecen las bases del proyecto, eligiendo el entorno de desarrollo, definiendo los recursos que se utilizarán y realizando una planificación del proyecto, basada en la metodología de trabajo elegida.

3.1.3 Producción

La producción del videojuego es el eje central del proyecto, dónde se ejecuta el proyecto. En primer lugar es necesario un análisis del sistema a partir del cual se define el diseño del mismo. Posteriormente se ejecuta el desarrollo del software, desglosando el trabajo en *sprints* durante los cuales se desarrollan las funcionalidades del sistema. Al finalizar cada sprint se realizan las pruebas necesarias y se validan las funcionalidades implementadas con el *product owner*.

3.1.4 Posproducción

La última fase del proyecto se orienta al desarrollo de la documentación, manuales de usuario y el mantenimiento del software. Éste último no se tendrá en cuenta por tratarse de un proyecto de fin de grado y no de un proyecto software comercial en el que sería indispensable el mantenimiento del sistema.

3.2 Planificación

La planificación del desglose de trabajo del proyecto se representa con los siguientes de diagramas de *Gant*. Empezando por una visión general del proyecto, y después detallando las tareas que componen cada una de las fases.

Proyecto Pykels: 166 días estimados.



FIGURA 3. PLANIFICACIÓN PYKELS

Fase de Preproducción: 54 días estimados.

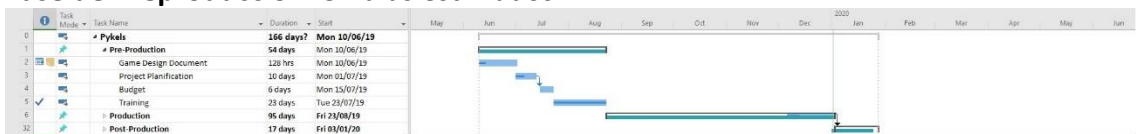


FIGURA 4. PLANIFICACIÓN PREPRODUCCIÓN

Fase de Producción: 95 días estimados.

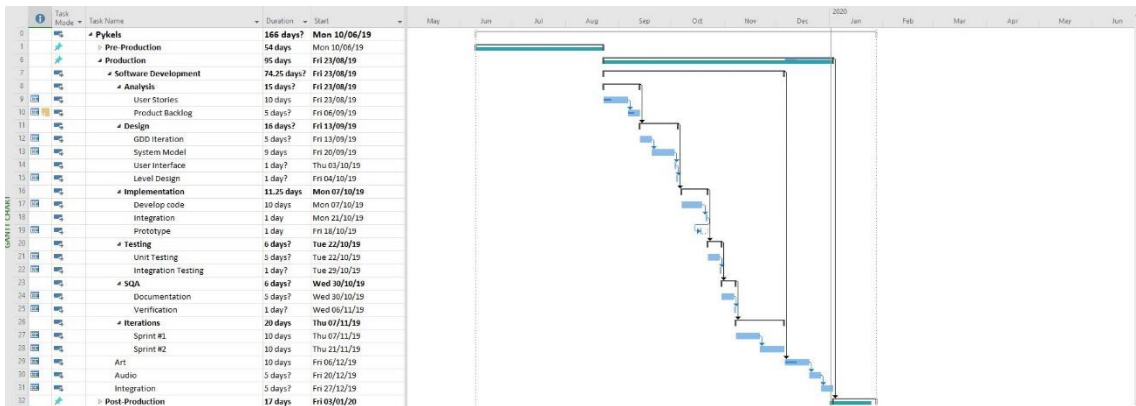


FIGURA 5. PLANIFICACIÓN PRODUCCIÓN

Fase de Posproducción: 17 días estimados.



FIGURA 6. PLANIFICACIÓN POSPRODUCCIÓN

3.3 Recursos

A continuación se listan todos los recursos empleados para desarrollar este proyecto.

3.3.1 Recursos hardware

Se utilizará un ordenador personal de sobremesa para el desarrollo, testing y documentación del proyecto.

3.3.2 Recursos software

Sistemas Operativos

- **PC desarrollo y pruebas:** Windows 10 Home Edition [[Windows 10, 2020](#)]

Entorno de desarrollo

- **Motor Gráfico:** Unity 2018.3.12f1
- **IDE:** Visual Studio Code 1.39.1 [[Visual Studio Code, 2020](#)]

Diseño de arte

Se utilizará Photoshop para el diseño y creación de Sprites, escenarios, animaciones, y en general todo el arte gráfico asociado al juego. Se dibujará digitalmente a mano, utilizando esta herramienta junto a la tableta gráfica.

- **Editor gráfico:** Adobe Photoshop CC 2019
- **Animaciones:** Unity Animations (integrada)
- **Efectos gráficos:** Unity Particles Plug-in

Desarrollo Software

- **Gestión de proyectos:** Microsoft Project (MS Office 365)
- **Diagramas y modelado:** draw.io, Magic Draw
- **Control de versiones:** Gitlab [[Gitlab, 2020](#)]
- **Pruebas:**

Ofimática

MS Office 365 [[Office, 2020](#)]

3.3.3 Recursos Humanos

- **Desarrolladores:** Programadores de C#
- **Analistas:** Especialistas en requisitos software.
- **Jefe de Proyecto:** Máximo responsable del proyecto.
- **Diseñador del juego:** Diseñador de escenas y niveles del videojuego.
- **Diseñadores gráficos:** Diseñadores de recursos y arte gráfico.
- **Animadores:** Diseñadores de animaciones.
- **Compositor:** Diseñador de la música y efectos de sonido
- **Tutor:** Guía y orientador del proyecto.
- **Testers:** Especialistas en pruebas de software.

Nótese, que todos los roles mencionados se asumen por el estudiante del trabajo de fin de carrera, excepto el de jefe de proyecto, el cual asume el tutor del trabajo, guiando y verificando el desarrollo del mismo.

También cabe mencionar, en el ámbito de recursos artísticos, que *Pykels* es un *Game from Craft* (videojuego artesanal), lo cual implica que tanto recursos sonoros/auditivos como gráficos del videojuego son creados a mano y desde cero asumiendo los roles de diseñador gráfico, compositor y animador.

3.4 Análisis

En éste capítulo se identifican, analizan y establecen los requisitos; esto implica una especificación del comportamiento y las funcionalidades del sistema. Para ello, y haciendo uso de la metodología *Scrum* elegida, se realizará un proceso análisis iterativo centrado en las necesidades del cliente.

3.4.1 Actores

El producto a crear se trata de un videojuego en el que participan 3 personas, cada una de las cuales toma un papel determinado en el juego. Desde este punto de vista podemos considerar a los jugadores como actores que asumen un rol para superar los retos que se plantean trabajando en equipo.

Aquí se identifican los actores del sistema como las personas y grupos que interactúan con el videojuego serio *Pykels*.

ID	<i>Número de identificación único (ACT-XY)</i>
Nombre	<i>Nombre del actor que ejerce la interacción</i>
Descripción	<i>Descripción detallada del actor</i>
Interacción	<i>Medio de interacción con el sistema</i>

FIGURA 7. EJEMPLO TABLA ACTOR

ID	ACT-00
Nombre	Game Party
Descripción	Equipo de jugadores que interactúan con el juego. Referencia a los 3 jugadores en conjunto.
Interacción	Ratón y teclado

FIGURA 8. ACT-00 GAME PARTY

ID	ACT-01
Nombre	Player 1
Descripción	Jugador del equipo que controla el personaje principal
Interacción	Flechas de dirección del teclado

FIGURA 9. ACT-01 PLAYER 1

ID	ACT-02
Nombre	Player 2
Descripción	jugador del equipo que controla la activación de elementos del entorno (plataformas, objetos y habilidades)
Interacción	Caracteres alfabéticos del teclado

FIGURA 10. ACT-02 PLAYER 2

ID	ACT-03
Nombre	Player 3
Descripción	Jugador del equipo que controla el ratón para descubrir elementos del entorno y permitir las activaciones del jugador 2
Interacción	Botones del ratón y posición del cursor

FIGURA 11. ACT-03 PLAYER 3

3.4.2 Objetivos

Para definir el objetivo general del producto final se dividen objetivos del desarrollo a los que se asociarán los requisitos del software. De esta manera se mantienen agrupadas las funcionalidades del juego con dependencias a los objetivos de alto nivel, se facilita la trazabilidad de los requisitos y obtiene una visión general del videojuego.

Los objetivos de alto nivel del desarrollo del videojuego *Pykels* se listan a continuación.

ID	<i>Número de identificación único (OBJ-XY)</i>
Título	<i>Título del objetivo</i>
Descripción	<i>Descripción general del objetivo</i>
Estado	Alcanzado/No alcanzado

FIGURA 12. EJEMPLO TABLA OBJETIVO

ID	OBJ-00
Título	Pykels
Descripción	Creación de una fase completa del videojuego serio Pykels
Estado	Alcanzado

FIGURA 13. OBJ-00 PYKELS

ID	OBJ-01
Título	Estructura del videojuego
Descripción	Diseño y creación de la estructura y mecánicas básicas de un videojuego de plataformas sidescroller 2D
Estado	Alcanzado

FIGURA 14. OBJ-01 ESTRUCTURA DEL VIDEOJUEGO

ID	OBJ-02
Título	Implementación de Pykels
Descripción	Creación de niveles y programación de las funcionalidades específicas del videojuego
Estado	Alcanzado

FIGURA 15. OBJ-02 IMPLEMENTACIÓN DE PYKELS

ID	OBJ-03
Título	Creación del entorno gráfico
Descripción	Diseño y creación de los recursos gráficos del juego
Estado	Alcanzado

FIGURA 16. OBJ-03 CREACIÓN DEL ENTORNO GRÁFICO

ID	OBJ-04
Título	Creación del sonido de juego
Descripción	Creación del audio asociado al juego
Estado	Alcanzado

FIGURA 17. OBJ-04 CREACIÓN DEL SONIDO DE JUEGO

ID	OBJ-05
Título	Creación de la GUI

Descripción	Diseño y creación de la interfaz gráfica de usuario, menús, HUD, etc.
Estado	Alcanzado

FIGURA 18. OBJ-05 CREACIÓN DE LA GUI

3.4.3 Requisitos del Sistema

Ramificando los objetivos se establecen dependencias con los requisitos del software, para los que se realiza un análisis de las funcionalidades y características que se espera alcanzar. Desde ésta perspectiva, se ha realizado un análisis de requisitos apoyado por la metodología *scrum* y las *User Stories*, las cuales definen de manera sencilla características del software.

Tras la obtención de las *User Stories* a partir del cliente, se analiza la información para formalizar la especificación técnica de requisitos del sistema, la cual se lista a continuación. La información que se plantea ha sido utilizada durante el desarrollo, constituyendo las tareas a realizar, y en la figura 17 se explica la utilidad que proporciona cada campo.

ID	<i>Número de identificación único (RF/RNF-XY)</i>
Título	Título descriptivo del requisito
Grupo	<i>Grupo de requisitos al que pertenece</i>
Versión	Número de versión del videojuego en la que se implementa
User Story	<i>Historia de usuario de la que se origina el requisito</i>
Prioridad	<i>Grado de importancia del requisito durante el desarrollo (Baja, Media, Alta)</i>
Objetivo	Objetivo de alto nivel que depende del requisito

FIGURA 19. RF-01 CÁMARA DE JUEGO

Requisitos Funcionales:

Los requisitos funcionales representan las funcionalidades específicas que el software debe proporcionar. Se asocian a los objetivos de alto nivel dependiendo del ámbito del requisito.

ID	RF-01
Título	Cámara principal
Grupo	Cámaras
Versión	V1
User Story	“La cámara principal debe ser ortográfica, en dos dimensiones y con perspectiva lateral del juego (<i>sidescroller</i>)”
Prioridad	Alta
Objetivo	OBJ-00, OBJ-01

FIGURA 20. RF-01 CÁMARA DE JUEGO

ID	RF-02
Título	Cámara sigue al personaje
Grupo	Cámaras
Versión	V1
User Story	“La cámara principal debe seguir al personaje, de manera suave y fluida”
Prioridad	Alta
Objetivo	OBJ-00, OBJ-01

FIGURA 11. RF-02 CÁMARA SIGUE AL PERSONAJE

ID	RF-03
Título	Límites de cámara
Grupo	Cámaras
Versión	V1
User Story	“La cámara principal debe estar siempre dentro de los límites del nivel”
Prioridad	Alta
Objetivo	OBJ-00, OBJ-01

FIGURA 22. RF-03 LÍMITES DE CÁMARA

ID	RF-04
Título	Efecto parallax
Grupo	Cámaras
Versión	V1
User Story	“El movimiento de la cámara principal debe dotar de perspectiva al juego; desplazándose rápidamente los planos principales y lentamente los planos más profundos”
Prioridad	Media
Objetivo	OBJ-00, OBJ-02

FIGURA 23. RF-04 CÁMARA EFECTO PARALLAX

ID	RF-05
Título	Físicas de Juego
Grupo	Físicas
Versión	V1
User Story	“El juego debe controlar las físicas del personaje principal y los escenarios gestionando colisiones, estados de salto, tipos de superficies, activadores, etc”
Prioridad	Alta
Objetivo	OBJ-00, OBJ-01

FIGURA 24. RF-05 FÍSICAS DE JUEGO

ID	RF-06
Título	Control del personaje
Grupo	Control del personaje
Versión	V1
User Story	“El control del personaje debe ser a través de las flechas de dirección del teclado”
Prioridad	Alta
Objetivo	OBJ-00, OBJ-02

FIGURA 25. RF-06 CONTROL DEL PERSONAJE

ID	RF-07
Título	Movimiento horizontal
Grupo	Control del personaje
Versión	V1
User Story	“El personaje principal debe poder moverse horizontalmente”
Prioridad	Alta
Objetivo	OBJ-00, OBJ-02

FIGURA 26. RF-07 MOVIMIENTO HORIZONTAL

ID	RF-08
Título	Movimiento vertical
Grupo	Control del personaje
Versión	V1
User Story	“El personaje principal debe poder saltar y agacharse”
Prioridad	Alta
Objetivo	OBJ-00, OBJ-02

FIGURA 27. RF-08 MOVIMIENTO VERTICAL

ID	RF-09
Título	Control de plataformas
Grupo	Control de plataformas
Versión	V1
User Story	“Las plataformas activables deben poder controlarse mediante el alfabeto del teclado, asociándose la tecla con la forma de la plataforma”
Prioridad	Alta
Objetivo	OBJ-00, OBJ-02

FIGURA 28. RF-09 CONTROL DE PLATAFORMAS

ID	RF-10
Título	Control de ratón
Grupo	Control de ratón
Versión	V1
User Story	“Por medio de la posición del cursor se podrán visualizar plataformas y partes ocultas del escenario”
Prioridad	Alta
Objetivo	OBJ-00, OBJ-02

FIGURA 29. RF-10 CONTROL DE RATÓN

ID	RF-11
Título	Plataforma con movimiento vertical
Grupo	Plataformas
Versión	V1
User Story	“Deben existir plataformas con desplazamiento vertical automático y en bucle”
Prioridad	Media
Objetivo	OBJ-00, OBJ-02

FIGURA 30. RF-11 PLATAFORMA CON MOVIMIENTO VERTICAL

ID	RF-12
Título	Plataforma con solidez activable
Grupo	Plataformas
Versión	V1
User Story	“Deben existir contornos de plataformas cuya activación por teclado las convierta en sólidas”
Prioridad	Alta
Objetivo	OBJ-00, OBJ-02

FIGURA 31. RF-12 PLATAFORMA CON SOLIDEZ ACTIVABLE

ID	RF-13
Título	Plataforma giratoria
Grupo	Plataformas
Versión	V1
User Story	“Deben existir plataformas giratorias”
Prioridad	Media
Objetivo	OBJ-00, OBJ-02

FIGURA 32. RF-13 PLATAFORMA GIRATORIA

ID	RF-14
Título	Hojas cayendo
Grupo	Efectos
Versión	V1
User Story	“Deben caer hojas desde el cielo en todos los escenarios”
Prioridad	Baja
Objetivo	OBJ-00, OBJ-02

FIGURA 33. RF-14 HOJAS CAYENDO

ID	RF-15
Título	Nubes
Grupo	Efectos
Versión	V1
User Story	“Deben existir nubes en el cielo desplazándose y generándose dinámicamente”
Prioridad	Baja
Objetivo	OBJ-00, OBJ-02

FIGURA 34. RF-15 NUBES

ID	RF-15
Título	Portales
Grupo	Efectos
Versión	V1
User Story	“Deben existir portales formados con partículas representando la finalización del nivel y dando acceso al siguiente si procede”
Prioridad	Alta
Objetivo	OBJ-00, OBJ-02

FIGURA 35. RF-15 PORTALES

ID	RF-16
Título	Transición entre niveles
Grupo	Flujo de juego
Versión	V1
User Story	“Deben existir portales al final de cada nivel desde los que se accederá al siguiente escenario mediante un efecto de transición”
Prioridad	Alta
Objetivo	OBJ-00, OBJ-01

FIGURA 36. RF-16 TRANSICIÓN ENTRE NIVELES

ID	RF-17
Título	Límites de nivel
Grupo	Flujo de juego
Versión	V1
User Story	“Deben existir límites en todos los niveles que impidan al personaje salir de los escenarios”
Prioridad	Alta
Objetivo	OBJ-00, OBJ-01

FIGURA 37. RF-17 LÍMITES DE NIVEL

ID	RF-18
Título	Pausar el juego
Grupo	Flujo de juego
Versión	V1
User Story	“Se debe poder pausar el juego en cualquier momento”
Prioridad	Alta
Objetivo	OBJ-00, OBJ-01

FIGURA 38. RF-18 PAUSAR EL JUEGO

ID	RF-19
Título	Reanudar el juego
Grupo	Flujo de juego
Versión	V1
User Story	“Se debe poder reanudar el juego en cualquier momento mientras esté pausado”
Prioridad	Alta
Objetivo	OBJ-00, OBJ-01

FIGURA 39. RF-19 REANUDAR EL JUEGO

ID	RF-20
Título	Menú de inicio
Grupo	Menús
Versión	V1
User Story	“La primera pantalla de juego debe ser el menú de inicio; permitiendo iniciar una partida, acceder al menú de opciones o salir de la aplicación”
Prioridad	Alta
Objetivo	OBJ-00, OBJ-05

FIGURA 40. RF-20 MENÚ DE INICIO

ID	RF-21
Título	Menú de pausa
Grupo	Menús
Versión	V1
User Story	“Se debe mostrar un menú al pausar el juego; permitiendo reanudar el juego, volver al menú de inicio o salir del juego”
Prioridad	Alta
Objetivo	OBJ-00, OBJ-05

FIGURA 41. RF-21 MENÚ DE PAUSA

ID	RF-22
Título	Menú de opciones
Grupo	Menús
Versión	V1
User Story	“Debe existir dentro del menú de inicio la posibilidad de configurar parámetros como la resolución y el sonido”
Prioridad	Alta
Objetivo	OBJ-00, OBJ-05

FIGURA 42. RF-22 MENÚ DE OPCIONES

ID	RF-23
Título	Tutorial
Grupo	Niveles
Versión	V1
User Story	“Debe diseñarse un tutorial como primer nivel para instruir e iniciar a nuevos jugadores”
Prioridad	Alta
Objetivo	OBJ-00, OBJ-02, OBJ-03, OBJ-04

FIGURA 43. RF-23 TUTORIAL

ID	RF-24
Título	Nivel 1
Grupo	Niveles
Versión	V1
User Story	“Debe diseñarse el primer nivel del juego con retos de dificultad baja”
Prioridad	Alta
Objetivo	OBJ-00, OBJ-02, OBJ-03, OBJ-04

FIGURA 44. RF-24 NIVEL 1

ID	RF-25
Título	Nivel 2
Grupo	Niveles
Versión	V1
User Story	“Debe diseñarse el segundo nivel del juego con retos de dificultad media”
Prioridad	Alta
Objetivo	OBJ-00, OBJ-02, OBJ-03, OBJ-04

FIGURA 45. RF-25 NIVEL 2

ID	RF-26
Título	Nivel 3
Grupo	Niveles
Versión	V1
User Story	“Debe diseñarse el tercer nivel que finalice la primera fase del juego”
Prioridad	Alta
Objetivo	OBJ-00, OBJ-02, OBJ-03, OBJ-04

FIGURA 46. RF-26 NIVEL 3

Requisitos No Funcionales:

Los requisitos no funcionales representan las características en forma de restricciones sobre el producto final. Los requisitos no funcionales también se asocian a objetivos generales.

ID	RNF-01
Título	Jugabilidad
Grupo	Calidad
Versión	V1
User Story	“El control del juego no debe trabarse, siendo fluida cada interacción por parte de los jugadores”
Prioridad	Alta
Objetivo	OBJ-00

FIGURA 47. NRF-01 JUGABILIDAD

ID	RNF-02
Título	Rendimiento
Grupo	Calidad
Versión	V1
User Story	“Los gráficos de juego deben verse fluidos en todo momento, como mínimo se ejecutará a 30fps”
Prioridad	Alta
Objetivo	OBJ-00

FIGURA 48. NRF-02 RENDIMIENTO

ID	RNF-03
Título	Compatibilidad
Grupo	Calidad
Versión	V1
User Story	“El juego debe funcionar en cualquier ordenador actual con Windows 7, 8 o 10”
Prioridad	Alta
Objetivo	OBJ-00

FIGURA 49. NRF-03 COMPATIBILIDAD

3.4.4 Matriz de trazabilidad de requisitos

En esta sección se adjunta la matriz de trazabilidad de requisitos, la cual permite conocer el estado del desarrollo en todo momento. La utilización de esta tabla es esencial para un desarrollo software de calidad, en el que se evite dejar funcionalidades o características sin implementar. En la figura 48 se define la matriz de trazabilidad asociada a los requisitos que el videojuego *Pykels* demanda en su primera versión.

R \ OBJ	OBJ-00	OBJ-1	OBJ-2	OBJ-3	OBJ-4	OBJ-5
RF-01						
RF-02						
RF-03						
RF-04						
RF-05						
RF-06						

RF-07						
RF-08						
RF-09						
RF-10						
RF-11						
RF-12						
RF-13						
RF-14						
RF-15						
RF-16						
RF-17						
RF-18						
RF-19						
RF-20						
RF-21						
RF-22						
RF-23						
RF-24						
RF-25						
RF-26						
NRF-01						
NRF-02						
NRF-03						

FIGURA 50. MATRIZ DE TRAZABILIDAD DE REQUISITOS

3.5 Modelo

Previo al diseño se requiere establecer la visión general del sistema, para ello se realiza una división en módulos cada uno de los cuales contará con un modelo que represente los distintos componentes y las asociaciones entre ellos.

A continuación se representan los diagramas generados durante el modelado del sistema.

Módulos del sistema:

Como se muestra en la figura 50, el sistema del videojuego serio *Pykels* se compone de tres módulos, con sus correspondientes componentes. Esta agrupación ha sido la implementada desde el inicio del desarrollo, y no se han realizado modificaciones en este sentido.

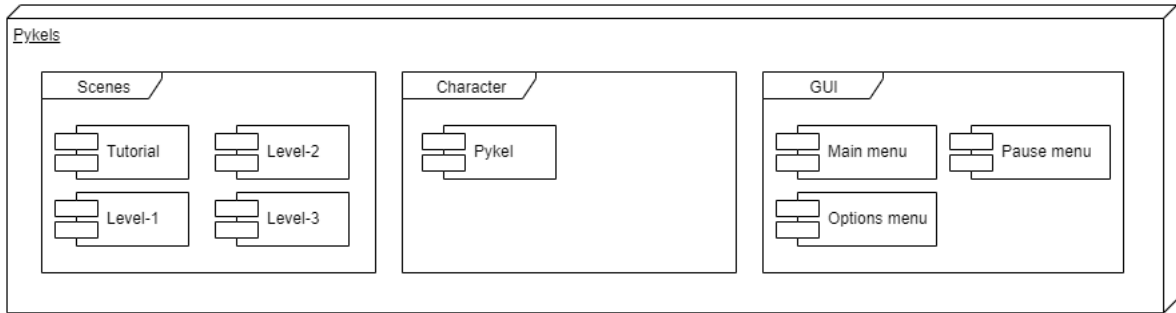


FIGURA 51. MÓDULOS DEL SISTEMA

Modelo Personaje:

El videojuego cuenta con un único personaje, el cual será controlado por uno de los tres jugadores que interaccionen con el juego. En la figura 51 se representa el modelo final de dicho personaje por medio de un diagrama de clases. Este modelo fue variando durante el desarrollo debido a la necesidad de adaptación a los cambios de género y contexto del videojuego.

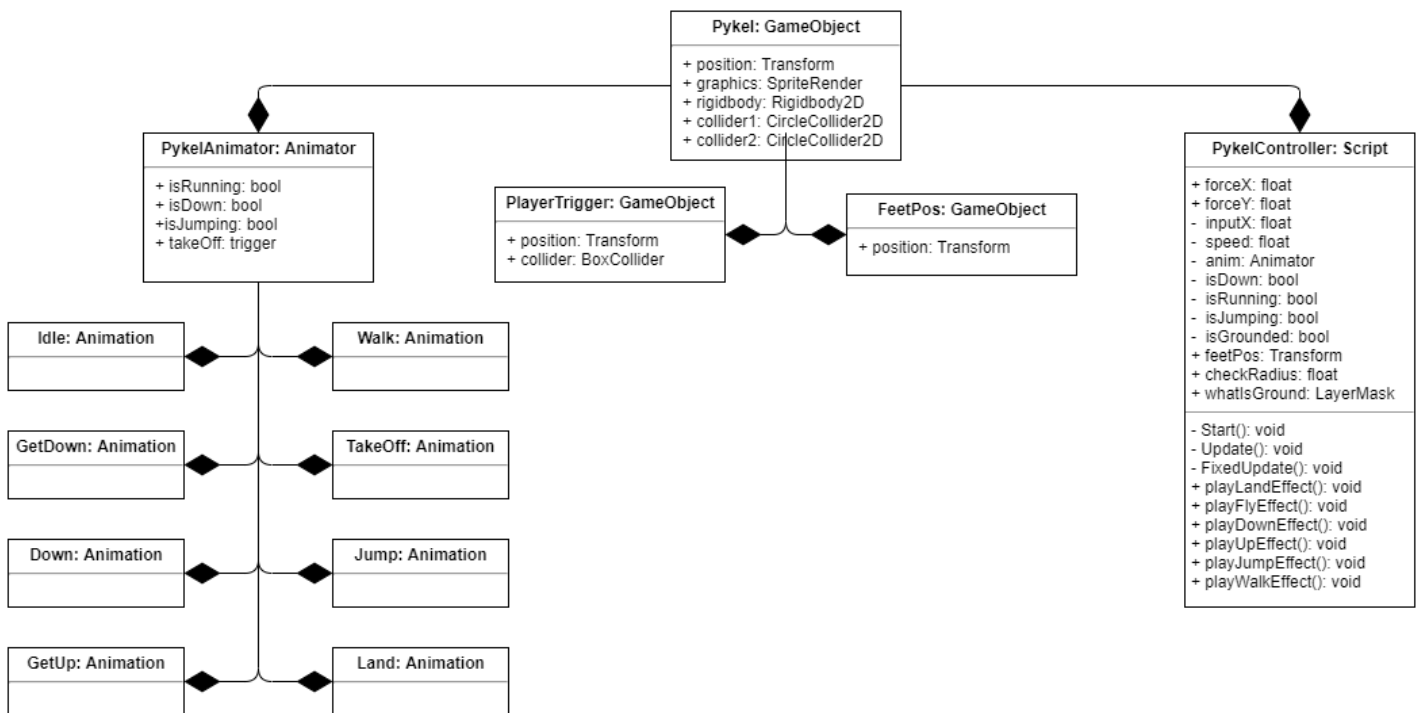


FIGURA 52. MODELO DEL PERSONAJE

Modelo Escenas:

Las escenas constituyen la agrupación de más alto nivel dentro del desarrollo en Unity. En la figura 52 se puede contemplar el modelo genérico utilizado para la construcción de las escenas del videojuego *Pykels*.

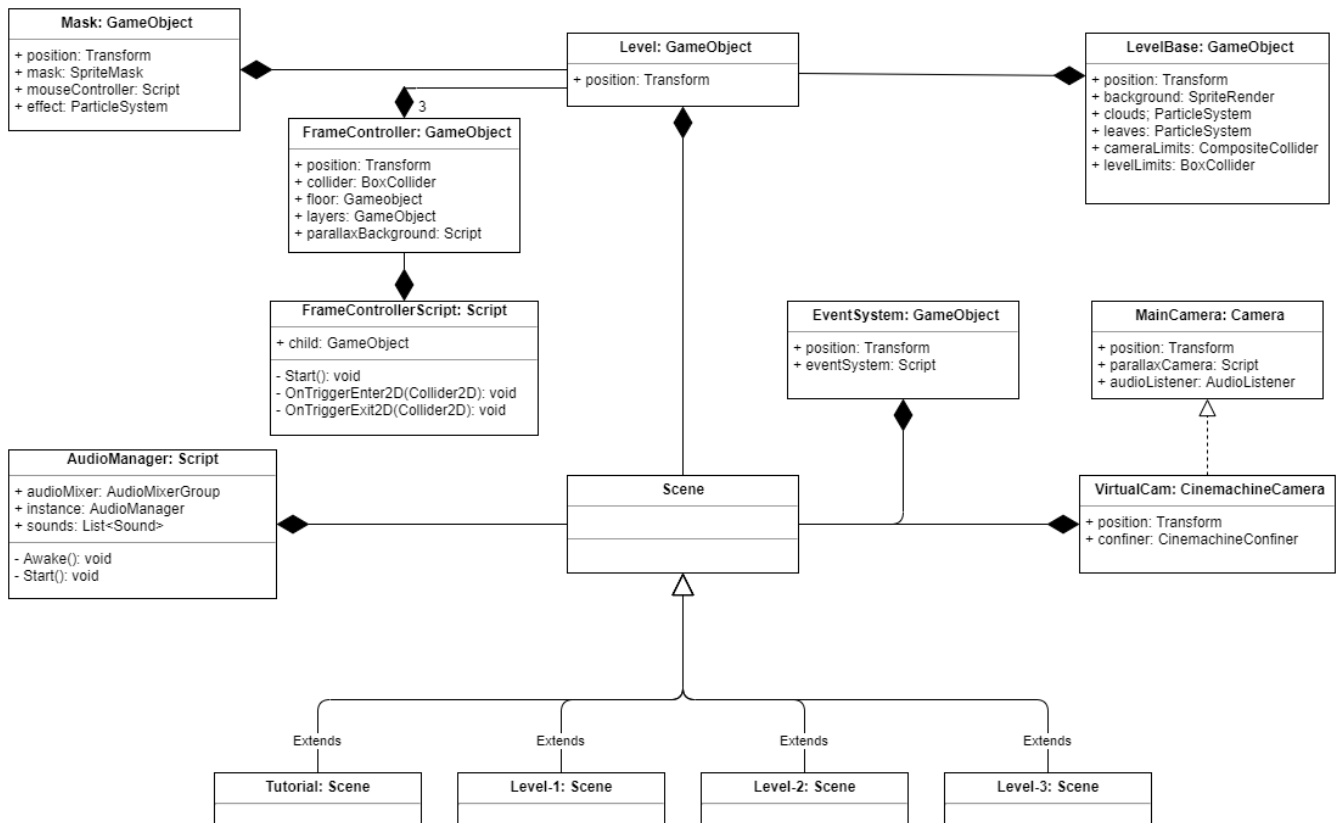


FIGURA 53. MODELO DE ESCENAS

Modelo GUI:

La interfaz gráfica de usuario es una parte esencial de cualquier producto software, es por ello que, para un desarrollo de calidad, se deba modelar la estructura de dicho módulo. Para el videojuego *Pykels* la interfaz gráfica se ha modelado de la forma representada en la figura 53.

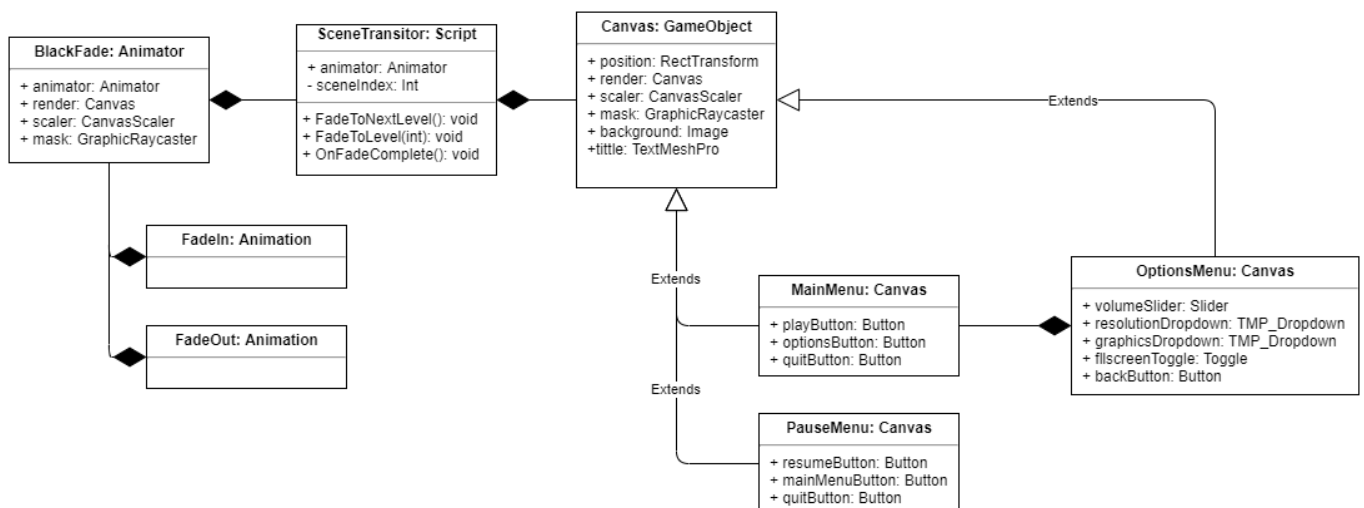


FIGURA 54. MODELO GUI

3.6 Diseño

Esta sección describe parte del trabajo de diseño realizado en el proyecto, el cual se documenta detalladamente en el documento de diseño del videojuego (GDD).

3.6.1 Patrón MVC

El motor gráfico Unity está orientado al desarrollo de videojuegos bajo el patrón de diseño Modelo Vista Controlador (MVC) [MVC, 2020]. Es por ello, la elección de esta estructura software para el desarrollo del videojuego *Pykels*.

Utilizando el patrón de diseño MVC se realiza la siguiente distribución:

Modelo: Constituye la jerarquía y relaciones entre los componentes Unity del juego.

Vista: Módulo de salida del sistema con la que interaccionan los usuarios.

Controlador: Conjunto de scripts que determinan el comportamiento del juego y la interacción de los jugadores. Utiliza el modelo y controla la vista.

3.6.2 Documento de Diseño

Se adjunta el GDD en su última versión como anexo A. No obstante se incluyen a continuación algunas partes esenciales para la comprensión del diseño y desarrollo del videojuego.

3.6.3 Diseño Técnico

En esta sección se detalla el apartado técnico definido en la fase de diseño y actualizado durante el desarrollo del juego por medio del GDD.

Motor Gráfico

El motor gráfico será Unity, utilizando el tipo de proyecto 2D. La plataforma destino será PC (Windows) y se construirán dos versiones, de x32 y de x64. El juego se adapta a las distintas resoluciones del usuario, siendo la máxima 4k. La composición de gráficos del juego se basa en 3 capas (frontal, principal y fondo), utilizando *sprites* en formato png, con 100 ppu (pixels Per Unit).

Audio

Todo el audio se ha generado y compuesto para el juego, no se utiliza música o efectos de sonido externos. La banda sonora principal deberá ser calmada y relajante, con instrumentos clásicos y melodías simples pero pegadizas. También es necesario disponer de melodías específicas y sonidos de efectos especiales para dotar de realismo e inmersión al juego.

Tal y como se muestra en la figura 55, se utilizan una clase *AudioManager* y otra *Sound* para la gestión del sonido del juego, a través del *Mixer* de Unity. *AudioManager* permanecerá siempre en escena, evitando cortes de audio entre cambios de escenas, y se compondrá de un array de la clase *Sound*, a partir de los cuales se crearán los correspondientes *AudioSources*.

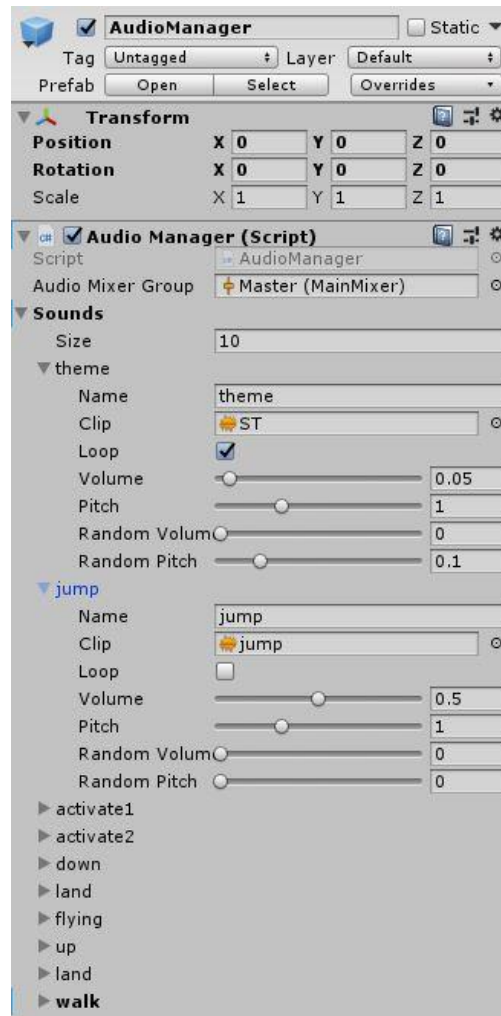


FIGURA 55. AUDIO MANAGER

Escenarios

El escenario de cada nivel se compone de 3 *frames*, donde cada *frame* equivale a dos pantallas de 4k (7680x2189), de esta manera el jugador recorrerá cada *frame* dos veces. Los tres *frames* de cada nivel están gestionados por un controlador que los desactiva cuando el jugador sale de ellos y reactiva cuando se acerca; con esto se pretende establecer un diseño de niveles eficiente que permita una posterior extensión de los mismos sin reparar en rendimiento. Dentro de cada *frame* se distribuirán jerárquicamente los elementos del escenario (plataformas, decoración, capas, partículas...)

Interfaz

Mediante el componente *Canvas* de Unity se implementan las interfaces del juego del menú principal y el menú de pausa. En el caso del menú de inicio compondrá una escena propia, mientras el de pausa se sobrepondrá oculto sobre cada nivel y se activará con la tecla escape, parando mediante dicha acción el juego (quedando en segundo plano tras un fondo transparente) y desactivando las instrucciones del nivel si las hubiera (para evitar superponer dos textos).

Transiciones de escenas

La transición entre escenarios y niveles debe ser lo más fluida posible, siendo lo ideal que todo el flujo de juego sea continuo sin parones. Por ejemplo, como en Limbo donde el juego completo se puede completar sin pantallas de carga, parones. Para ello se utilizarán las animaciones de Unity para crear efectos Fading. Mientras que debido al diseño de los niveles se podría implementar todo el juego en una sola escena de manera eficiente, ya que sólo está activo en cada momento el *frame* en el que se encuentra el jugador.

Cada nivel contará con un portal que definirá la superación del mismo, el objetivo es simple, llevar a *Pykel* al portal para pasar al siguiente nivel. Dicho portal se comunica con el *SceneManager* para ejecutar la transición al siguiente nivel. Nótese que tras el último nivel se una fase se volverá al menú inicial.

Cámara principal

La cámara principal es la única cámara del cada escena, de tipo ortográfica, con un tamaño de 10, permitiendo de esta manera un procesado gráfico limpio. Se utilizará el componente *cinemachine* para la gestión de la cámara; de manera que pueda seguir al personaje principal, pero sin salir de los límites de la escena (*CameraLimits*).

Se utilizará efecto *parallax* sobre la capa frontal (*front layer*) mediante los scripts asociados se permitirá desplazar dicha capa cuando el jugador se desplace, dotando de profundidad al juego.

3.6.4 Diseño Artístico

En esta sección se detalla el apartado artístico definido en la fase de diseño y actualizado durante el desarrollo del juego por medio del GDD. Como se menciona anteriormente, todos los recursos gráficos han sido creados por el autor del trabajo y en ningún caso se han obtenido de terceros.

Estilo Artístico

Todo el apartado artístico será diseñado de cero, no se utilizarán directamente recursos gráficos externos.

El estilo artístico general será el minimalista, pero orientado al público infantil, se utilizarán figuras geométricas básicas para el diseño de escenarios, pero con colores llamativos y gradientes. Se deberán diseñar sombras en base a la luz de cada escenario. Se tomarán como referencias artísticas juegos como *Limbo*, *Inside* pero dotando las escenas de colorido para atraer a los jugadores.

Interfaces de Usuario

En las figuras 56-57 se muestran las capturas de la interfaz gráfica del videojuego.



FIGURA 56. GUI-MENU PRINCIPAL



FIGURA 57. GUI-MENU DE PAUSA

Sprites

Para el diseño gráfico se ha empleado la herramienta de dibujo Photoshop en combinación con una tableta gráfica. El proceso de dibujo se ha realizado por partes, comenzando con bocetos evolutivos hasta la obtención de un resultado satisfactorio.

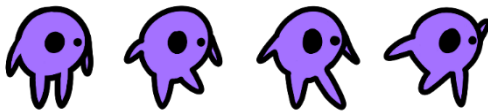


FIGURA 58. PYKEL WALK SPRITES



FIGURA 59. PYKEL IDLE SPRITES

Como se puede observar en las figuras 58 y 59, las animaciones se componen de distintos *sprites* sobre los que se transita durante el tiempo. Según el estado del personaje se reproducirá una secuencia de *sprites* u otra.



FIGURA 60. TREE-1



FIGURA 61. TREE-2

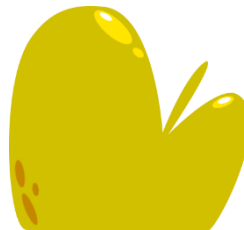


FIGURA 62. BUSH-1



FIGURA 63. ROCK-1

En las figuras 64-67, se representan los tipos de plataformas sobre los que el personaje podrá moverse. Estas plataformas cuentan con propiedades adicionales según el nivel del juego.



FIGURA 64. PLATFORM-I



FIGURA 65. PLATFORM-V



FIGURA 66. PLATFORM-O

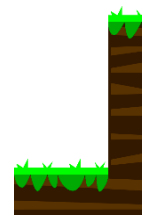


FIGURA 67. PLATFORM-L

En las figuras 68 y 69 se muestran los bloques utilizados como suelos en altura para las distintas escenas. Mientras que en la figura 70 se adjunta el suelo principal del videojuego.

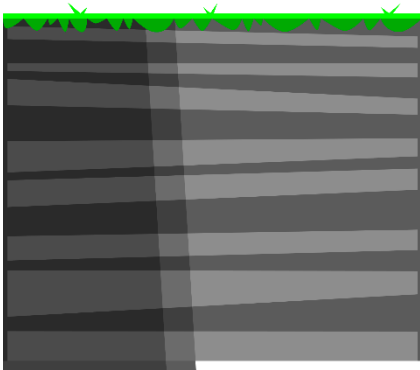


FIGURA 68. BLOCK-1

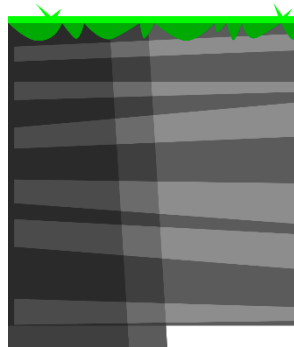


FIGURA 69. BLOCK-2

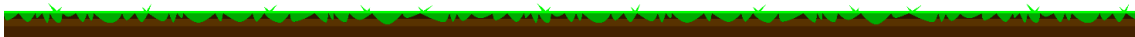


FIGURA 70. SUELO PRINCIPAL

Animaciones

Idle: animación de *Pykel* por defecto cuando está parado. Dotará de vida al personaje que pestañeará cada tres segundos.

Walk: animación de movimiento, animará a *Pykel* cuando se desplace en el eje horizontal.

Jump: animación de salto, constará de tres partes (despegue, salto y aterrizaje), se utilizará la flecha arriba del teclado y un parámetro *grounded* para definir el salto del personaje.

Down: animación de agachado del personaje, carece de funcionalidad, pero se implementó en un prototipo inicial y se decidió dejarla porque resulta divertida y dota de personalidad al personaje.

FadeIn: Transición del *canvas* negro a transparente y desactivado.

FadeOut: Transición del *canvas* transparente a negro y activado.

PauseAnimation: Transición del *canvas* de desactivado a negro semitransparente

4

Desarrollo del videojuego

En este capítulo se describe detalladamente el producto final, así como el proceso de desarrollo.

4.1 Versiones del GDD

El documento de diseño del videojuego es un documento vivo que se actualiza durante todo el desarrollo. A continuación se listan las distintas modificaciones realizadas en forma cronológica.

- 1. 19/06/19**
 - a. Creación del documento
 - b. Distribución de secciones principales
- 2. 22/06/19**
 - a. Desarrollo de la Introducción
- 3. 22/06/19**
 - a. Adición y modificación de secciones
- 4. 23/06/19**
 - a. Creación del título
 - b. Definición de la atmósfera y el contexto
 - c. Desarrollo de la historia
- 5. 24/06/19**
 - a. Especificación de retos
 - b. Finalización de la sección introducción
- 6. 25/06/19**
 - a. Finalización de la sección El Juego
 - b. Finalización de la sección Mecánicas
 - c. Finalización de la sección Ámbito educativo
- 7. 26/06/19**
 - a. Modificación de personajes
 - b. Distribución temporal de personajes
 - c. Revisión y corrección de erratas
 - d. Finalización del primer borrador
- 8. 04/07/2019**
 - a. Revisión del tutor
 - b. Inicio de la segunda versión del GDD

c. Acotación de la idea de juego

9. 08/07/2019

a. Corrección de errores y falta de detalle

10.09/07/2019

a. Reescritura de la sección Concepto

b. Edición de la sección Género

c. Edición de la sección Características

d. Edición de la sección Género

e. Edición de la sección Propósito

11.10/07/2019

a. Especificación detallada de los retos

12.15/07/2019

a. Creación del sistema de Iteraciones y Fases

b. Extensión de sección Características (más información)

13.16/07/2019

a. Inicio especificación del Apartado Artístico

b. Modificaciones menores en algunos retos

c. Corrección de incongruencias y modificaciones menores

14.17/07/2019

a. Adición de recursos gráficos y referencias

b. Extensión del apartado artístico

15.18/07/2019

a. Revisión completa y finalización del segundo borrador

16.21/08/2019

a. Inicio de la tercera versión del GDD

b. Edición de detalles del contenido, inicio de aplicación, autoguardado y carga de partidas

17.30/08/2019

a. Cambio de historia completo para facilitar el desarrollo

b. Nuevo título *Pykels*

18.02/09/2019

a. Modificación Introducción

b. Modificación Mecánicas de juego

19.03/09/2019

a. Reescritura de apartado El juego

b. Modificación Mecánicas de juego

20.24/10/2019

a. Extensión apartado artístico

b. Extensión apartado técnico

4.2 El producto final

4.2.1 Propósito

El propósito principal del videojuego es evaluar la capacidad de trabajar en equipo de un grupo, analizando cómo se desenvuelve tanto cada individuo como el grupo en general en los distintos niveles del juego, y en base a las capacidades definidas.

Por otra parte, en el ámbito escolar, se transmite la importancia y el beneficio del trabajo en equipo a la hora de afrontar retos; y se fortalecen los aspectos y capacidades asociadas como comunicación, debate, toma de decisiones, cohesión, etc. También se acerca la tecnología y el uso de ordenadores a los niños, ofreciendo un videojuego divertido y que rompe con estereotipos, diferente a los que los niños están acostumbrados a jugar (violencia, matar, sexistas, etc.).

Por la orientación del juego, éste, puede ser enfocado a otros contextos fuera del escolar, por ejemplo, en selecciones de personal donde se busquen altas prestaciones de trabajo en equipo, o en el tratamiento de autismo infantil.

4.2.2 La atmósfera

Este videojuego se ambienta en un planeta desconocido, con características similares a la tierra (gravedad parecida, vegetación, agua, etc.), y habitado por unas pequeñas criaturas con aspecto de células llamadas *Pykels*.

El planeta es un sitio colorido y bonito, lleno de formas geométricas y con un fondo espacial atractivo; pero se contrasta con una sensación de suspense en el ambiente.

4.2.3 La historia

Los *Pykels* conviven en armonía en algún lugar del universo, pero una de estas criaturas se separa del grupo quedando sola. El equipo jugador deberá controlar éste personaje y el escenario para lograr reunir la pequeña criatura con su colonia de nuevo.

Puesto que en la primera versión del juego se implementará únicamente la primera fase, se define solo la introducción de la historia, dejando el núcleo de la historia y el desenlace abiertos para un desarrollo futuro.

4.2.4 La perspectiva

Pykel es realmente un juego de plataformas, con la distinción de que es cooperativo, es decir, cada nivel del plataformas está pensado para ser resuelto en equipo. El juego se basa en la perspectiva 2D pero con sensación de profundidad utilizando una cámara ortográfica y la sobreposición de las capas del escenario. Las cinemáticas serán animaciones predefinidas dentro del mismo juego y se utilizará un fondo estilo *parallax* para la sensación de movimiento de la perspectiva.

4.2.5 Personaje

Los *Pykels* son una raza ficticia, con aspecto similar a una célula; con influencias de *Flubber*, y *Zac* de *League of Legends*. Estas inofensivas criaturas de aspecto viscoso y elástico nacen a partir de la fusión de elementos en una galaxia desconocida.

El personaje principal será *Pykel*, una de estas criaturas a la que los jugadores del equipo deberán controlar para resolver los retos. *Pykel* contará inicialmente con un conjunto de habilidades básicas que se irá extendiendo con cada fase completa; siendo el set de habilidades para la versión inicial: movimiento lateral, y salto, complementado con la interacción del entorno con ratón y teclado.

Físicas de personaje

Como se puede observar en la figura 71, se controlan las colisiones del personaje mediante dos *CircleCollider2D* dispuestos verticalmente. El componente *FeetPos* es un objeto vacío que hereda del personaje y se utiliza para controlar cuando se encuentra sobre una superficie asociada a la capa de suelo (*layer floor*).

Por otro lado en la figura 72 se muestran los parámetros del componente *Rigidbody2D*. Nótese que el movimiento del personaje y parte de las físicas se gestionan mediante el script del controlador del personaje, el cual se detalla en el apartado siguiente.

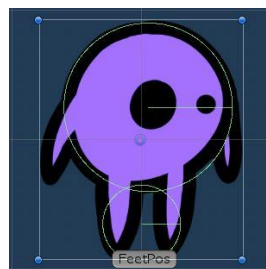


FIGURA 71. PYKEL COLLIDERS

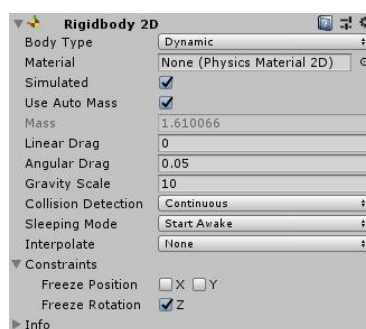


FIGURA 72. PYKEL RIGIDBODY

Controlador de personaje

El script *PykelController* ejerce de controlador, gestionando los movimientos, estados, animaciones y sonidos del personaje.

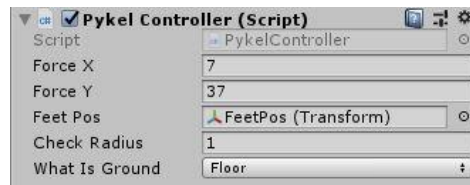


FIGURA 73. PYKELCONTROLLER SCRIPT

Diagrama de estados

En la figura 74 se adjunta el diagrama de animaciones, que representa perfectamente los distintos estados en los que puede estar el personaje durante el tiempo de juego, así como las transiciones entre ellos. Dicho diagrama se asocia igualmente al contexto artístico de las animaciones ya que en el diseño de *Pykels* cada estado se corresponde con una animación; se detallan más extensamente las animaciones en el apartado correspondiente de diseño artístico.

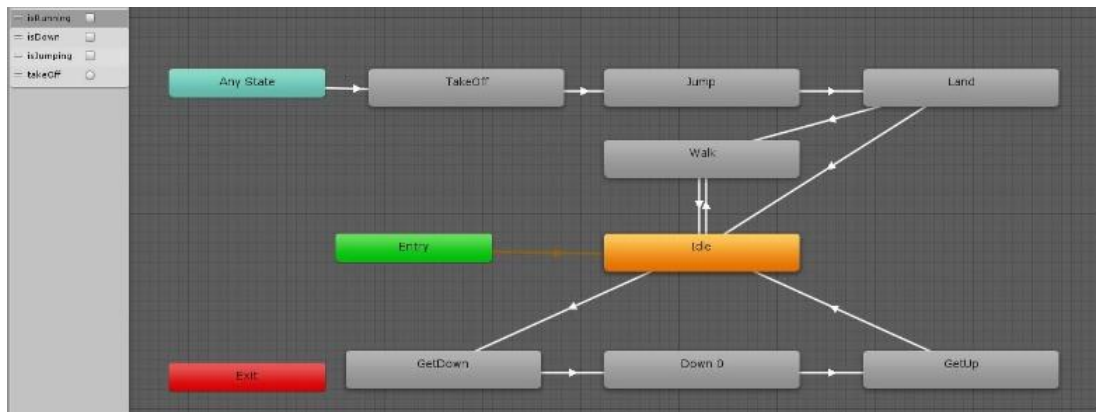


FIGURA 74. PYKEL STATE MACHINE

4.2.6 El flujo de juego

se define el flujo completo del juego, desde el inicio de la aplicación hasta la finalización de la primera fase que constituye la versión inicial.

Inicio aplicación

Se mostrará el logo del videojuego junto al título 3 segundos sobre un fondo negro, y luego mediante una transición de transparencia aparecerá el menú de inicio sobreimpreso mientras el fondo negro se convierte en el planeta.

Menú de inicio

Jugar: Se iniciará una nueva partida. Desaparece el menú de la pantalla y comienza el juego con una transición *Fading*.

Opciones: Se permite la configuración del juego como el control de sonido, resolución, calidad gráfica, pantalla completa, etc.

Salir: Salir del juego, guardando el progreso de la partida en curso. Se mostrará una ventana de confirmación previo al cierre de la aplicación.

Fase 1 - Tutorial

Este es el primer escenario al empezar una nueva partida, donde aparecerán inscritas las instrucciones de control básicas. El nivel consta de cuatro partes.

Movimiento básico: Con *Pykel* en el centro del escenario y un grabado con las instrucciones de movimiento (flechas de dirección) simplemente se debe avanzar hacia la derecha.

Salto: Aparecerá un montículo al que se deberá saltar para continuar el avance. Sobre el montículo estará grabada la instrucción de salto (flecha arriba).

Cursor: Llegados a este punto aparecerá la primera plataforma oculta, que habrá que descubrir con el desplazamiento del ratón sobre ella (*sprite mask*) y activar mediante las teclas correspondientes.

Finalización: Con una última pantalla con diversas plataformas para enseñar a los jugadores las mecánicas de juego, al final de esta última parte del tutorial habrá un vórtice temporal a través del cual se accederá al siguiente nivel. En las figuras 75-86 se representan las partes del escenario de dicho nivel.

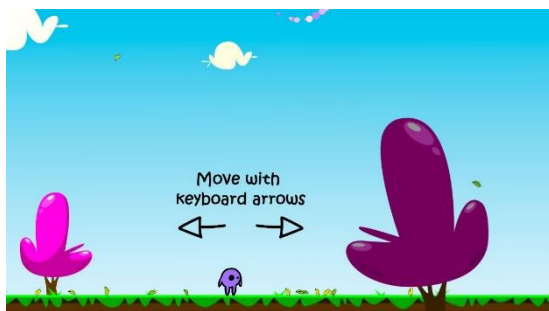


FIGURA 75. TUTORIAL-1



FIGURA 76. TUTORIAL-2

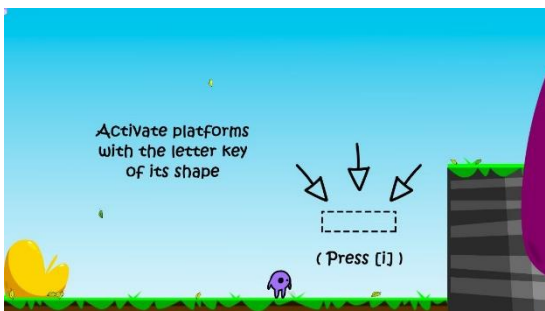


FIGURA 77. TUTORIAL-3

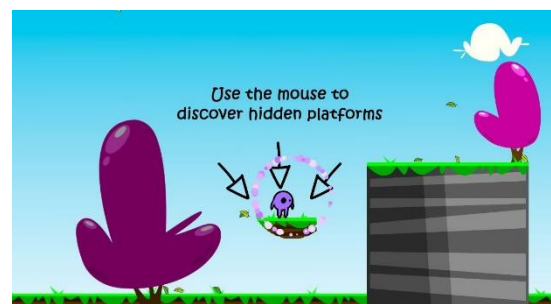


FIGURA 78. TUTORIAL-4



FIGURA 79. TUTORIAL-4

Fase 1 - Nivel 1

En este primer nivel se introducen plataformas móviles, combinadas con las funciones ya presentadas en el tutorial (normales, ocultas y activables). Se pretende que el equipo coordine los controles del juego para llevar a *Pykel* al portal para acceder al siguiente nivel. En las figuras 80-86 se representan las partes del escenario de dicho nivel.

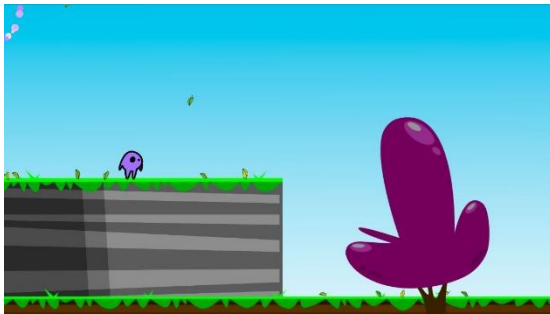


FIGURA 80. NIVEL1-1



FIGURA 81. NIVEL1-2



FIGURA 82. NIVEL1-3

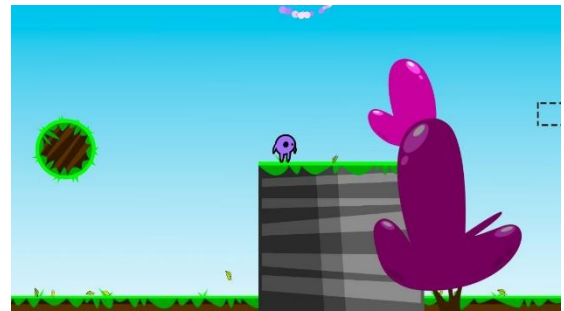


FIGURA 83. NIVEL1-4



FIGURA 84. NIVEL1-5

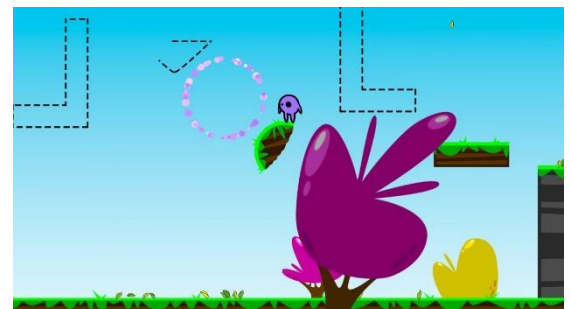


FIGURA 85. NIVEL1-6

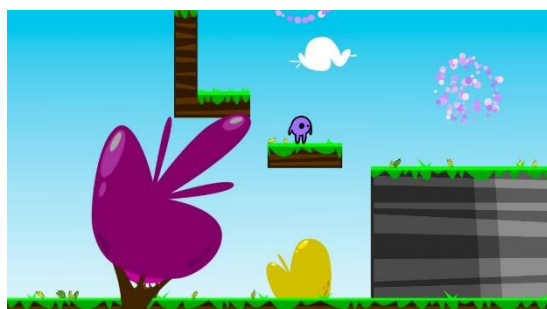


FIGURA 86. NIVEL1-7

Fase 1 - Nivel 2

En el segundo nivel de la fase inicial se introducen plataformas falsas, las cuales pueden parecer normales, pero bajo el *sprite mask* asociado al ratón se podrá comprobar que

realmente son solo imágenes (sin *collider*) por lo que éstas plataformas deberán evitarse. También se incluyen plataformas temporales en las que tras un segundo desaparecerán un breve período de tiempo, para volver a aparecer mediante un sistema de *respawn*.

Fase 1 - Nivel 3

El último nivel finaliza la primera fase del juego, en él se introduce un camino oculto por encima del escenario visible, al cual se deberá acceder mediante una plataforma oculta al inicio del nivel. En el recorrido de este camino existirán plataformas dinámicas de niveles anteriores, si el personaje cae del camino oculto deberá volver al inicio para reintentarlo. Al final del camino un portal oculto permitirá finalizar la fase, volviendo al menú principal del juego.

4.3 Manual de usuario

4.3.1 Concepto

Pykels es un juego serio cooperativo, que se ofrece como un recurso pedagógico orientado a la enseñanza y el análisis de la capacidad de trabajo en equipo de escolares.

El juego está compuesto por una serie de niveles, que el equipo deberá superar para completar los objetivos de la historia. Cada nivel presenta distintos retos, que pretenden poner a prueba las capacidades de trabajo en equipo (coordinación, comunicación, cohesión, creatividad, organización, etc.). Una persona externa (profesor, tutor, padres, etc.) deberá ejercer de evaluador, observando el comportamiento del equipo durante el transcurso del juego, para posteriormente realizar un breve cuestionario que evalúe el rendimiento del equipo. A partir de los datos de este cuestionario se generarán unos informes del rendimiento del equipo y de las capacidades mostradas. En conclusión, un videojuego de dinámicas de trabajo en equipo para niños.

4.3.2 Características

Un equipo (de tres jugadores como mínimo) debe trabajar en conjunto para superar los niveles que el videojuego les plantea. El juego constará de varias fases compuesta cada una por tres niveles, donde cada nivel supone un reto diseñado para destacar en los participantes las capacidades del trabajo en equipo (Coordinación, Cohesión, Investigación, Creatividad, Especialización, Previsión, Ejecución, Iniciativa, Planificación).

Durante todo el juego un evaluador externo al equipo (tutor, padres, profesores, etc.) debe observar el comportamiento tanto individual como colectivo del equipo, y al finalizar cada fase (tres niveles), completar un breve cuestionario con el que se evalúa el rendimiento y las capacidades de trabajo en equipo demostradas.

De esta manera al completar el juego se habrán evaluado, en varias fases, todas las capacidades del equipo, y se podrá generar, a partir de los cuestionarios realizados, un informe final con el análisis de la capacidad para trabajar en equipo que presenta el

grupo. Sin embargo, en la primera versión de este proyecto sólo se implementará la primera fase con los tres retos y cuestionarios correspondientes.

Inicialmente el idioma será únicamente inglés, y no se contará con diálogos de juego; no existirá abundancia de textos más lejos de las interfaces de usuario y algún elemento de referencia dentro del juego. Para futuras versiones se espera poder implementar una configuración de internacionalización con diversas opciones, además de extender el uso de textos y diálogos en el juego para permitir el aprendizaje de nuevos idiomas.

4.3.3 Las reglas de juego

Se informará a los equipos con estas normas durante la introducción.

Participación: Todos los integrantes del equipo deben participar activamente en la resolución de cada nivel.

Planificación: Para cada nivel hay que dedicar unos minutos para analizar el escenario y elaborar un plan o idea, de manera que cada miembro del equipo tenga una tarea.

Comunicación: La comunicación entre el equipo es clave, se debe hablar para planificar cada nivel y durante el juego para informar de la situación de cada momento del juego.

4.3.4 Interacción

El equipo interactuará mediante los periféricos de entrada (Ratón y teclado), siendo recomendable que cada jugador se asocie con una función específica y se vayan rotando dichas funciones de manera que todos participen al menos una vez en cada posición. Las funciones de interacción con el juego son:

Jugador	Interacción	Función
Jugador 1	Flechas dirección teclado	Movimiento del personaje
Jugador 2	Teclas alfabeto teclado	Activación de elementos
Jugador 3	Posición del cursor	Plataformas ocultas

FIGURA 87. TABLA INTERACCIÓN

Si el equipo consta de más de tres jugadores estas funciones deben asignarse de diferente manera en cada reto; el resto de jugadores deberán influir en la resolución de los retos planificando y coordinando las actividades del equipo.

4.3.5 La mecánica de juego

Aquí se definen detalladamente las mecánicas de *Pykels*. Es recomendable que cada mecánica se ejecute por un solo jugador en cada nivel, debiendo el resto participar en su resolución sin interactuar con el juego.

Movimiento básico: Por medio de las flechas de dirección del teclado se controlará el movimiento básico del personaje en plataformas; izquierda, derecha, y salto.

Letras teclado – control del escenario: Ciertas plataformas y elementos de los escenarios podrán tener asociada una tecla para su activación. El equipo deberá

investigar y analizar los escenarios para identificar estos aspectos y planificar una solución para resolver los retos. Por ejemplo, plataformas con forma de O cuyo colider se activa mediante el teclado.

Ratón – Revelador: durante todo el juego, el manejo del ratón será clave, teniendo este asociado un *spritemask* en su posición revelará plataformas, caminos, y elementos ocultos. El cursor tendrá aspecto de nube de partículas que revela partes del escenario, esto implica una constante comunicación de los jugadores para la resolución de los retos.

Plataformas: Se contará inicialmente con 4 tipos básicos de plataformas (I, V, O, L), derivando cada una en distintas variaciones. Se utilizará la siguiente nomenclatura: *Platform-(forma)-(variaciones)-(identificador)*. Formas: L, O, V, I (forma base de la plataforma). Variaciones: Activable (A), oculta (H), rotatoria (R), desplazable (M), falsa (F), temporal (T). Por ejemplo Platform-V-1 sería la primera plataforma del nivel con forma de V y básica; Platform-L-HAT-3 sería la tercera plataforma del nivel con forma de L oculta, activable y temporal.

4.3.6 Los retos

Los retos están diseñados para poner a prueba y enseñar las capacidades específicas de trabajo en equipo (Coordinación, Cohesión, Investigación, Creatividad, Especialización, Previsión, Ejecución, Iniciativa, Planificación). En cada nivel se introducen nuevas mecánicas, buscando añadir el análisis de nuevas capacidades además de las de los retos anteriores.

Reto nivel 1 (Comunicación y coordinación): Se requiere una coordinación y timing a la hora de realizar ciertos movimientos, implicando al equipo en la comunicación y ejecución sincronizada. Se pretenden evaluar la coordinación y la comunicación.

Reto nivel 2 (Estimación y previsión): Se pretende que el equipo planifique y prevea las amenazas (plataformas falsas) para lograr superar el nivel. Esto implica al equipo en la estimación y previsión de la toma de decisiones

Reto nivel 3 (Planificación y exploración): El tercer nivel se orienta a la investigación del escenario por parte del equipo, y a la planificación de la solución para superar los retos del camino oculto una vez se encuentre. Al igual que en niveles anteriores se deberán coordinar las interacciones de los jugadores para lograr finalizar el recorrido y acceder al último portal de la fase.

4.4 Pruebas

El proceso de pruebas o testing es esencial en el desarrollo software, ya que garantiza que el sistema funciona correctamente. En esta sección se detallan las pruebas realizadas sobre la implementación del sistema *Pykels*; éstas se clasifican por su orientación en pruebas de componentes, de integración y de interfaz.

ID	<i>Número de identificación único (TC-XY)</i>
Título	<i>Título del prueba</i>
Módulo	<i>Módulo del sistema donde se realiza la prueba</i>

Versión	<i>Número de versión del videojuego donde se realiza la prueba</i>
Definición	<i>Definición detallada de la prueba realizada</i>
Prioridad	<i>Grado de importancia de la prueba (Baja, Media, Alta)</i>

FIGURA 88. EJEMPLO TABLA CASO DE PRUEBA

4.4.1 Pruebas del Personaje

Las pruebas de componentes se dirigen a los elementos de alto nivel dentro de cada módulo del sistema.

ID	TC-01
Título	Pruebas de físicas del personaje
Módulo	Pykel
Versión	1
Definición	Testing en tiempo de juego, se realizan múltiples colisiones sobre el abanico completo de elementos de escenas del juego.
Prioridad	Alta

FIGURA 89. TC-01 PRUEBAS DE FÍSICAS DEL PERSONAJE

ID	TC-02
Título	Pruebas de control del personaje
Módulo	Pykel
Versión	1
Definición	Testing en tiempo de juego, se comprueban las interacciones sobre el personaje y las respuestas del sistema.
Prioridad	Alta

FIGURA 90. TC-02 PRUEBAS DE CONTROL DEL PERSONAJE

ID	TC-03
Título	Pruebas de animaciones del personaje
Módulo	Pykel
Versión	1
Definición	Testing en tiempo de juego, se comprueban las distintas animaciones y transiciones entre ellas.
Prioridad	Alta

FIGURA 91. TC-03 PRUEBAS DE ANIMACIONES DEL PERSONAJE

ID	TC-04
Título	Pruebas de triggers del personaje
Módulo	Pykel
Versión	1
Definición	Testing en tiempo de juego, se comprueban los activadores asociados al personaje para controlar estados, posición y interacción con otros módulos como escenarios.
Prioridad	Alta

FIGURA 92. TC-04 PRUEBAS DE TRIGGERS DEL PERSONAJE

4.4.2 Pruebas de Escenas

Las pruebas de escenas se dirigen a la construcción y los componentes de los distintos niveles que forman el juego.

ID	TC-05
Título	Pruebas de control de plataformas
Módulo	Scenes
Versión	1
Definición	Testing en tiempo de juego, se comprueba el control de todas las plataformas que componen el conjunto de niveles.
Prioridad	Alta

FIGURA 93. TC-05 PRUEBAS DE CONTROL DE PLATAFORMAS

ID	TC-06
Título	Pruebas de control del ratón
Módulo	Scenes
Versión	1
Definición	Testing en tiempo de juego, se comprueba el control del spritemask por ratón, asociado a la visibilidad de elementos ocultos del escenario..
Prioridad	Alta

FIGURA 94. TC-06 PRUEBAS DE CONTROL DEL RATÓN

ID	TC-07
Título	Pruebas de controlador de escenarios
Módulo	Scenes
Versión	1
Definición	Testing en tiempo de juego, se comprueba el script controlador de escenarios y las funciones asociadas.
Prioridad	Alta

FIGURA 95. TC-07 PRUEBAS DE CONTROLADOR DE ESCENARIOS

ID	TC-08
Título	Pruebas de transición de escenas
Módulo	Scenes
Versión	1
Definición	Testing en tiempo de juego, se comprueban las correctas transiciones entre escenas del juego.
Prioridad	Alta

FIGURA 96. TC-08 PRUEBAS DE TRANSICIÓN DE ESCENAS

ID	TC-09
Título	Pruebas de cámaras de juego
Módulo	Scenes
Versión	1
Definición	Testing en tiempo de juego, se comprueba la visualización del juego a través de las distintas cámaras de cada escena y su comportamiento durante el flujo de juego.
Prioridad	Alta

FIGURA 97. TC-09 PRUEBAS DE CÁMARAS DE JUEGO

4.4.3 Pruebas de Interfaz

Las pruebas de interfaz se dirigen a los menús y pantallas que forman parte de la interfaz gráfica de usuario.

ID	TC-10
Título	Pruebas del menú de inicio
Módulo	GUI
Versión	1
Definición	Comprobación del funcionamiento y diseño de la interfaz del menú de inicio.
Prioridad	Alta

FIGURA 98. TC-10 PRUEBAS DEL MENÚ DE INICIO

ID	TC-11
Título	Pruebas del menú de opciones
Módulo	GUI
Versión	1
Definición	Comprobación del funcionamiento y diseño de la interfaz del menú de opciones.
Prioridad	Alta

FIGURA 99. TC-11 PRUEBAS DEL MENÚ DE OPCIONES

ID	TC-12
Título	Pruebas del menú de pausa
Módulo	GUI
Versión	1
Definición	Comprobación del funcionamiento y diseño de la interfaz del menú de pausa.
Prioridad	Alta

FIGURA 100. TC-12 PRUEBAS DEL MENÚ DE PAUSA

4.5.4 Pruebas de Integración

Las pruebas de integración se dirigen a evaluar y corregir la interacción entre los distintos módulos que componen el juego.

ID	TC-13
Título	Pruebas de controlador de audio
Módulo	Scenes
Versión	1
Definición	Testing en tiempo de juego, se comprueba la integración del audio de juego a través del script controlador AudioManager, así como la reproducción de clips y efectos de sonido asociados a cada escena del juego.
Prioridad	Alta

FIGURA 101. TC-13 PRUEBAS DE CONTROLADOR DE AUDIO

ID	TC-14
Título	Pruebas de flujo del juego
Módulo	Scenes
Versión	1
Definición	Se comprueba la integración de la interfaz gráfica en tiempo de juego.
Prioridad	Alta

FIGURA 102. TC-14 PRUEBAS DE FLUJO DEL JUEGO

ID	TC-15
Título	Pruebas de configuración del juego
Módulo	GUI
Versión	1
Definición	Testing en tiempo de juego, se comprueba la configuración del juego a través del menú de opciones.
Prioridad	Alta

FIGURA 103. TC-15 PRUEBAS DE CONFIGURACIÓN DEL JUEGO

4.6 Calidad de Software

En el inicio del proyecto se definieron algunas técnicas asociadas a la calidad de software [SQA-1, 2020]. A continuación se describen dichas técnicas implementadas.

4.6.1 Trazabilidad de requisitos

El seguimiento de requisitos un sistema durante el desarrollo software puede llegar a ser caótico. Es por ello que se utilizan matrices de trazabilidad para representar las dependencias entre requisitos, permitiendo planificar de manera eficiente el desarrollo del software.

4.6.2 Validación

Al finalizar cada implementación de una característica del sistema es necesario que el cliente la valide, de esta manera se enfoca el desarrollo a las necesidades del cliente; y permite afrontar correcciones y cambios con menor impacto.

5

Conclusiones

En el capítulo que aquí se introduce, se realiza una evaluación del producto final desarrollado, desde un punto de vista analítico. Se presta especial atención al alcance de las posibles extensiones o mejoras que se podrían implementar en el futuro, y a los problemas de mayor relevancia afrontados durante todo el transcurso del proyecto.

5.1 Problemas encontrados

Aquí se detallan las dificultades y adversidades encontradas durante el desarrollo del proyecto.

5.1.1 Cambios en el concepto del juego

La fase de concepto supone la estructura del desarrollo del proyecto, después de la cual se deben tener clara la orientación del juego. En éste sentido, durante el desarrollo del proyecto *Pykels* se encontró con un problema en la definición del concepto, repercutiendo en la duración de dicha fase. En la primera iteración de la fase se definió la misma orientación de juego serio que ha terminado desarrollándose, pero en la parte de arte se realizaron hasta tres cambios implicando la realización de varias iteraciones sobre el documento de diseño.

Ésta adversidad fruto de la indecisión inicial provoca una extensión adicional del proyecto de tres meses ya que la dependencia de fases es secuencial. La solución que se decidida fue de posponer la fecha de entrega del proyecto de diciembre a febrero con la aceptación de todas las partes interesadas.

5.1.2 Recursos educativos

La decisión de orientar el juego al entorno educativo dio como fruto la interesante idea de enfocarse al análisis del trabajo en equipo, sin embargo, el no contar en el proyecto con un especialista en psicología ha dificultado el desarrollo de los recursos pedagógicos del juego. Por otra parte hay que mencionar que este proyecto se centra especialmente en el desarrollo del software, que constituye la parte esencial del videojuego. No obstante, el proyecto está desarrollado a falta de dicha parte, pudiendo diseñarse los cuestionarios e informes de equipo por un experto, y ajustándolos a las necesidades de cada entorno.

5.2 Ampliaciones

Realmente el concepto inicial de *Pykels* era mucho más ambicioso, pero se fueron realizando acotaciones durante el análisis por motivos de recursos disponibles. En esta

sección se detallan algunas de las características y posibles ampliaciones que no se han implementado por falta de recursos; pero que pueden ser considerados para posibles extensiones futuras del juego.

5.2.1 Desarrollo de fases

Como es lógico el videojuego *Pykels* requiere el desarrollo de más fases y niveles para poder comercializarse; la idea inicial era implementar nueve niveles distribuidos en tres fases, donde cada nivel correspondiera con una competencia de trabajo en equipo que evaluar. De esta manera se evaluarían todas las capacidades con retos enfocados a cada una de manera específica. Sin embargo, de nuevo por motivos de recursos, se decidió implementar la primera fase de esta estructura dejando el resto para ampliaciones futuras.

Cabe mencionar en este contexto que la historia del juego se ha dejado brevemente definida por motivo de la carencia de fases implementadas, esta decisión se orientó a ligar el ámbito psicológico de trabajo en equipo con la historia. De esta manera se incluyen expertos en psicología en el proceso de definición de la historia, dando matices que beneficien la inmersión del equipo en el juego.

5.2.2 Desarrollo de NPCs y agentes autónomos

En el modelo de la idea del juego inicial se pretendía incluir personajes no controlados por jugadores. Estos *npcs* (non player characters) apoyarían a la historia de juego y los retos, dotando de una atmósfera mucho más inmersiva. De igual manera se consideró la posibilidad de introducir agentes autónomos controlados por inteligencia artificial a modo de enemigos que se debían evitar, pero se consideró posponer dichos módulos para incluirse en fases posteriores.

En este sentido, la primera fase de *Pykels* transmite carencia de dinamismo, pero con la extensión de fases y las ampliaciones aquí mencionadas supondría una fase inicial misteriosa y flexible a cualquier trama.

5.2.3 Dinámicas de personaje

El módulo del personaje *Pykel* es esencial en el juego, suponiendo uno de los módulos con mayor carga de horas de desarrollo. Por este motivo se recortaron algunas características demasiado optimistas en la definición inicial. Uno de estos recortes ha sido el sistema de dinámicas de *Pykel*, en el cual el personaje iba adquiriendo nuevas habilidades según rescatase a otros *Pykels* habitantes del planeta en los distintos niveles. En este sentido, los niveles posteriores requerían de la habilidad obtenida en el nivel anterior; por ejemplo si en un nivel se obtenía la habilidad de doble salto, los niveles de las fases posteriores podían requerir de esta habilidad para su superación, constituyendo una dinámica secuencial de habilidades del personaje.

El sistema desarrollado cuenta con una habilidad que se definió con este objetivo pero que por la acotación realizada ha quedado sin utilidad en la primera fase, pero es totalmente compatible con la ampliación de este módulo en el futuro.

Bibliografía y referencias

[AEVI, 2020], La industria del videojuego en España.

[Asociación Española de Videojuegos](#), (Fecha de acceso 01/01/2020)

[Answers Unity, 2020], Foro oficial de Unity 3D.

[Answers Unity](#), (Fecha de acceso 01/01/2020)

[Aprendizaje cooperativo, 2020], Aprendizaje cooperativo.

[Aprendizaje cooperativo](#), (Fecha de acceso 01/01/2020)

[Binding of Isaac, 2020], Videojuego The Binding of Isaac (Edmund McMillen, 2011).

[The Binding of Isaac](#), (Fecha de acceso 01/01/2020).

[Brainstorm, 2020], Descripción de la técnica brainstorm.

[Brainstorm](#), (Fecha de acceso 01/01/2020)

[Clinton, 2010], Desarrollo de videojuegos con scrum.

Agile game development with Scrum, Keith Clinton (2010)

[Cpp, 2020], Lenguaje de programación C++.

[cplusplus](#), (Fecha de acceso 01/01/2020)

[Creative Cloud, 2020], Página oficial del entorno de desarrollo artístico de Adobe.

[Adobe Creative Cloud](#), (Fecha de acceso 01/01/2020)

[Csharp, 2020], Guía de referencia del lenguaje de programación C#.

[csharp reference](#), (Fecha de acceso 01/01/2020)

[draw.io, 2020], Herramienta de dibujo y diagramas online.

[draw.io](#), (Fecha de acceso 01/01/2020)

[Game Design Document, 2020], Descripción del documento de diseño de videojuegos.

[Gamasutra](#), (Fecha de acceso 01/01/2020)

[Gitlab, 2020], Página oficial de la herramienta de control de versiones GitLab.

[GitLab](#), (Fecha de acceso 01/01/2020)

[Indie Game, 2020], Documental sobre la industria de videojuegos indie.

[Indie Game](#), (Fecha de acceso 01/01/2020)

[Inside, 2020], Página oficial del videojuego Inside (Playdead, 2018).

[Inside](#), (Fecha de acceso 01/01/2020)

[Itch.io, 2020], Plataforma de alojamiento de videojuegos.

[itch.io](#), (Fecha de acceso 01/01/2020)

[Kingdom Hearts, 2020], Página oficial de Kingdom Hearts (Square Enix, 2002).

[Kingdom Hearts](#), (Fecha de acceso 01/01/2020)

[Krita, 2020], Página oficial de la herramienta de dibujo digital Krita.
[Krita](#), (Fecha de acceso 01/01/2020)

[Limbo, 2020], Página oficial del videojuego Limbo (Playdead, 2010).
[Limbo](#), (Fecha de acceso 01/01/2020)

[Magic Draw, 2020], Página oficial de la herramienta de modelado Magic Draw.
[Magic Draw](#), (Fecha de acceso 01/01/2020)

[Mastermind, 2020], Videojuego mastermind (1972).
[Mastermind](#), (Fecha de acceso 01/01/2020)

[Metodología agile, 2020], Introducción a las metodologías agile.
[Agile Methodologies](#), (Fecha de acceso 01/01/2020)

[Metodología en cascada, 2020], Comparativa de metodologías en cascada y ágiles.
[Waterfall VS Agile](#), (Fecha de acceso 01/01/2020)

[MIT, 2020], Licencia Opensource MIT.
[MIT](#), (Fecha de acceso 01/01/2020)

[MVC, 2020], Patrón de diseño Modelo Vista Controlador.
[Modelo Vista Controlador](#), (Fecha de acceso 01/01/2020)

[Office, 2020], Página oficial de la suite de ofimática Office de Microsoft.
[Microsoft Office](#), (Fecha de acceso 01/01/2020)

[Product Backlog, 2020], ¿Qué es el product backlog?
[Working with Product Backlog](#), (Fecha de acceso 01/01/2020)

[Resident Evil 4, 2020], Página oficial del videojuego Resident Evil 4 (Capcom, 2005).
[Resident Evil 4](#), (Fecha de acceso 01/01/2020)

[Scholar, 2020], Buscador académico de google.
[Google scholar](#), (Fecha de acceso 01/01/2020)

[Scrum, 2020], Guía de la metodología ágil Scrum.
[Software development with scrum](#), (Fecha de acceso 01/01/2020)

[SQA-1, 2020], Procesos de calidad para el desarrollo de software.
[QA in SDLC](#), (Fecha de acceso 01/01/2020)

[SQA-2, 2020], Tutorial rápido en de calidad de software.
[SQA Quick Tutorial](#), (Fecha de acceso 01/01/2020)

[Super Meat Boy, 2020], Página oficial de Super Meat Boy (Team Meat, 2010).

[Super Meat Boy](#), (Fecha de acceso 01/01/2020)

[Tibia, 2020], Página oficial del videojuego Tibia (Cipsoft, 1997).

[Tibia](#), (Fecha de acceso 01/01/2020)

[Tomb Raider I, 2020], Videojuego Tomb Raider (Core Designs, 1996).

[Tomb Raider I](#), (Fecha de acceso 01/01/2020)

[Trello, 2020], Página oficial de la herramienta de planificación Trello.

[Trello](#), (Fecha de acceso 01/01/2020)

[Unity, 2020], Desarrollo de videojuegos. Game Engine.

[Unity 3D](#), (Fecha de acceso 01/01/2020)

[Unity guide, 2020], Guía para el desarrollo de videojuegos en Unity.

[Game development in Unity](#), (Fecha de acceso 01/01/2020)

[User Stories, 2020], Historias de Usuario y sus beneficios durante el análisis.

[Requeriments and User Stories](#), (Fecha de acceso 01/01/2020)

[Videojuegos Indie, 2020], ¿Qué es un videojuego indie?

[Videojuegos Indie](#), (Fecha de acceso 01/01/2020)

[Videojuego serio, 2020], ¿Qué es un videojuego serio?

[Juego serio](#), (Fecha de acceso 01/01/2020)

[Visual Studio Code, 2020], Página oficial de la herramienta de desarrollo VS Code.

[Visual Studio Code](#), (Fecha de acceso 01/01/2020)

[Windows 10, 2020], Página oficial del sistema operativo Windows 10.

[Microsoft Windows 10](#), (Fecha de acceso 01/01/2020)

[Witcher 3, 2020], Página oficial del videojuego The Witcher 3 (CD Projekt, 2015).

[Witcher 3](#), (Fecha de acceso 01/01/2020)

Apéndices

Apéndice A

Manual de Instalación

Guía de instalación

Al finalizar el desarrollo en Unity del videojuego no es necesario ningún método específico de instalación, ya que el motor gráfico nos genera un ejecutable acorde a las plataformas disponibles. El videojuego *Pykels* estará disponible en varios formatos dependiendo del uso que se le quiera dar:

- **Ejecutable para plataforma Windows:** Formato genérico del videojuego con el cual se realiza una instalación común en sistema operativo del computador.
- **Itch.io:** Plataforma online para el alojamiento de videojuegos. Se introduce mediante un archivo de juego HTML generado desde Unity. Esta versión del juego es la más accesible y no requiere de instalación, únicamente de un ordenador con navegador con el que acceder.
- **GitLab:** Plataforma de control de versiones para el desarrollo software. Esta versión del videojuego ofrece el código fuente del videojuego para posibles ampliaciones, investigación, etc.

Requisitos mínimos

Sistema Operativo: Windows XP, Vista, 7

Procesador: 2.5 GHz

Memoria RAM: 1GB

Espacio en disco: 500MB

Gráficos: tarjeta compatible con Direct X9.0c

Apéndice B

Recursos Producidos

Paquete Prefabs

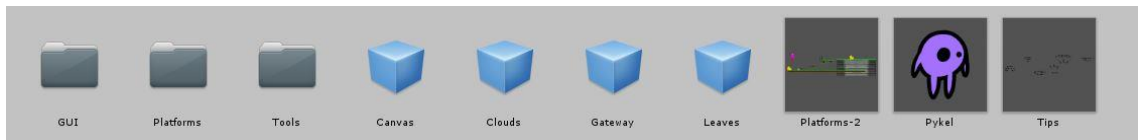


FIGURA 104. PAQUETE PREFABS-1

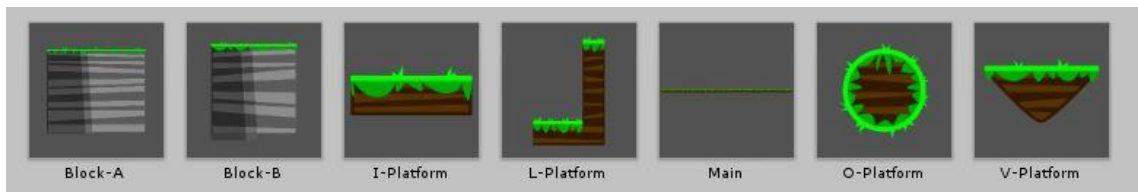


FIGURA 105. PAQUETE PREFABS-2



FIGURA 106. PAQUETE PREFABS-3

Paquete Scenes



FIGURA 107. PAQUETE SCENES

Paquete Sprites



FIGURA 108. PAQUETE SPRITES-1

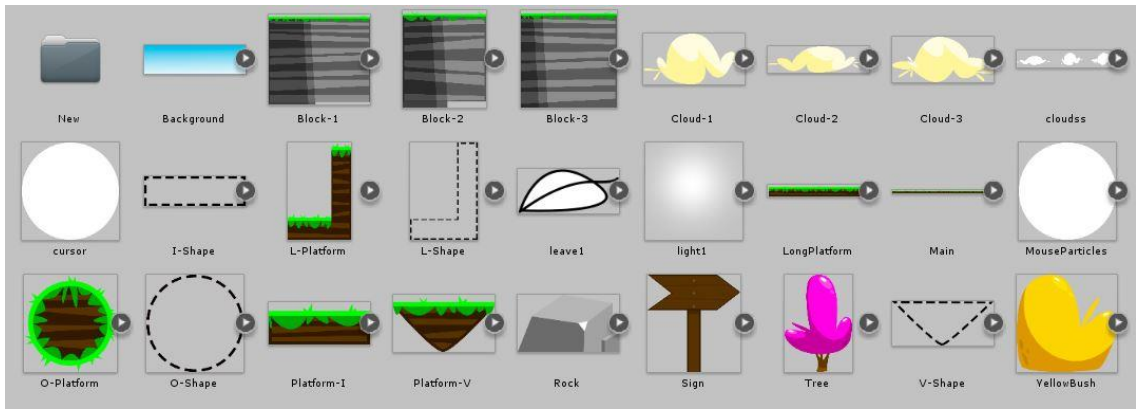


FIGURA 109. PAQUETE SPRITES-2

Package Animations

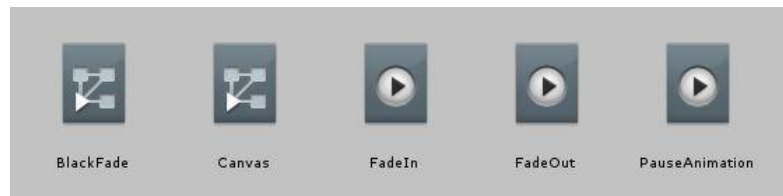


FIGURA 110. PAQUETE ANIMATIONS-1



FIGURA 111. PAQUETE ANIMATIONS-2

Package Audio

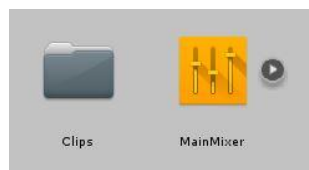


FIGURA 112. PAQUETE AUDIO-1

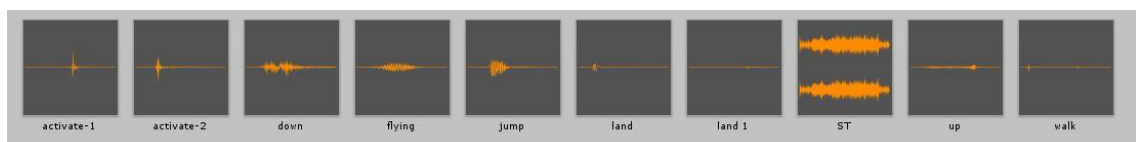


FIGURA 113. PAQUETE AUDIO-2

Paquete GUI



FIGURA 114. PAQUETE GUI-1



FIGURA 115. PAQUETE GUI-2

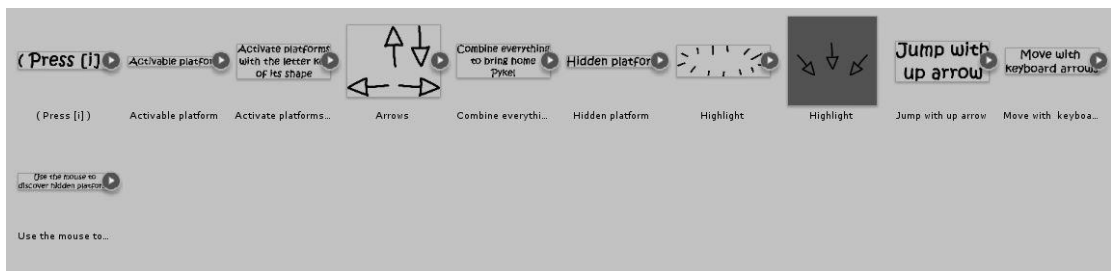


FIGURA 116. PAQUETE GUI-3

Paquete Materials

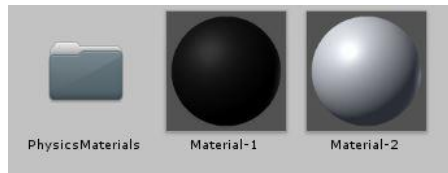


FIGURA 117. PAQUETE MATERIALS

Paquete PhysicMaterials

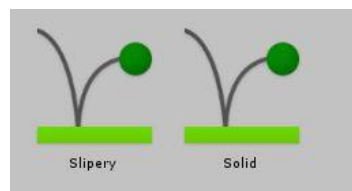


FIGURA 118. PAQUETE PHYSICMATERIALS

Paquete Particles



FIGURA 119. PAQUETE PARTICLES

Paquete Scripts

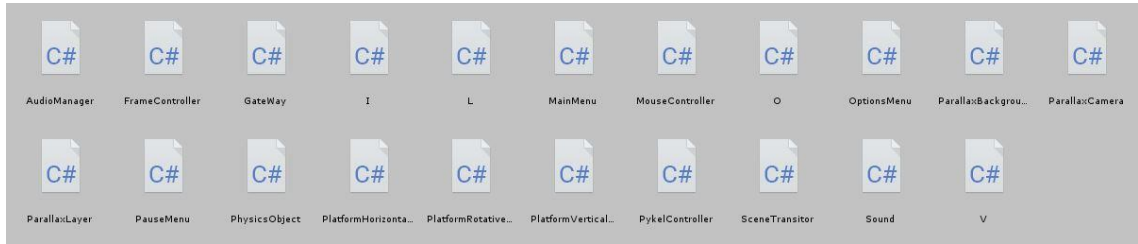


FIGURA 120. PAQUETE SCRIPTS

Apéndice C

Documento de Diseño del Videojuego

PYKELS

GAME DESIGN DOCUMENT

Introducción

Concepto

Pykels es un juego serio cooperativo, que se ofrece como un recurso pedagógico orientado a la enseñanza y el análisis de la capacidad de trabajo en equipo de escolares.

El juego está compuesto por una serie de niveles, que el equipo deberá superar para completar los objetivos de la historia. Cada nivel presenta distintos retos, que pretenden poner a prueba las capacidades de trabajo en equipo (coordinación, comunicación, cohesión, creatividad, organización, etc.). Una persona externa (profesor, tutor, padres, etc.) deberá ejercer de evaluador, observando el comportamiento del equipo durante el transcurso del juego, para posteriormente realizar un breve cuestionario que evalúe el rendimiento del equipo. A partir de los datos de este cuestionario se generarán unos informes del rendimiento del equipo y de las capacidades mostradas.

En conclusión, un videojuego de dinámicas de trabajo en equipo para niños.

Género

Pykels es principalmente un **juego serio**, ya que su propósito principal no es el de entretener, sino ejercer como herramienta para analizar y enseñar el trabajo en equipo en escolares. Por otro lado, las mecánicas del juego, su diseño de niveles y su modo de interacción le ubican en el género de **plataformas cooperativo** (en local), como son los videojuegos de la saga *Trine*, o *Worms*. El apartado artístico y la atmósfera del videojuego lo asocian al género de **aventura gráfica y ciencia ficción**, con toques divertidos y cómicos; como *Binding of Isaac* o *Hollow Knight*, etc.

Por último, debido a los informes que el juego genera y el contexto de uso al que esencialmente va destinado, se considera como un **recurso docente**, que, utilizado en aulas docentes puede aportar información pedagógica sobre la capacidad de trabajo en equipo tienen los alumnos, además de fortalecer ciertos aspectos en dicho ámbito.

Características

Un equipo (de tres jugadores como mínimo) debe trabajar en conjunto para superar los niveles que el videojuego les plantea. El juego constará de varias fases compuesta cada una por tres niveles, donde cada nivel supone un reto diseñado para destacar en los participantes las capacidades del trabajo en equipo (Coordinación, Cohesión, Investigación, Creatividad, Especialización, Previsión, Ejecución, Iniciativa, Planificación).

Durante todo el juego un evaluador externo al equipo (tutor, padres, profesores, etc.) debe observar el comportamiento tanto individual como colectivo del equipo, y al finalizar cada fase (tres niveles), completar un breve cuestionario con el que se evalúa el rendimiento y las capacidades de trabajo en equipo demostradas.

De esta manera al completar el juego se habrán evaluado, en varias fases, todas las capacidades del equipo, y se podrá generar, a partir de los cuestionarios realizados, un informe final con el análisis de la capacidad para trabajar en equipo que presenta el grupo. Sin embargo, en la primera versión de este proyecto sólo se implementará la primera fase con los tres retos y cuestionarios correspondientes.

Audiencia objetivo

El público objetivo son estudiantes de entre 8 y 16 años, a los que van enfocados los recursos y retos que ofrece el juego. El juego está orientado a aulas de estudiantes, que jugarán en múltiples partidas, divididos por equipos y supervisados por un tutor; pero este contexto es flexible, pudiéndose realizar en otros entornos y situaciones, siendo lo realmente esencial que la interacción con el juego sea en equipo de al menos tres jugadores, y que un evaluador supervise el comportamiento de los equipos para posteriormente realizar los cuestionarios de análisis de las capacidades demostradas.

Propósito

El propósito principal del videojuego es evaluar la capacidad de trabajar en equipo de un grupo, analizando cómo se desenvuelve tanto cada individuo como el grupo en general en los distintos niveles del juego, y en base a las capacidades definidas.

Por otra parte, en el ámbito escolar, se transmite la importancia y el beneficio del trabajo en equipo a la hora de afrontar retos; y se

fortalecen los aspectos y capacidades asociadas como comunicación, debate, toma de decisiones, cohesión, etc. También se acerca la tecnología y el uso de ordenadores a los niños, ofreciendo un videojuego divertido y que rompe con estereotipos, diferente a los que los niños están acostumbrados a jugar (violencia, matar, sexistas, etc.).

Por la orientación del juego, éste, puede ser enfocado a otros contextos fuera del escolar, por ejemplo, en selecciones de personal donde se busquen altas prestaciones de trabajo en equipo, o en el tratamiento de autismo infantil.

El juego

La atmósfera

Este videojuego se ambienta en un planeta desconocido, con características similares a la tierra (gravedad parecida, vegetación, agua, etc.), y habitado por unas pequeñas criaturas con aspecto de células llamadas *Pykels*.

El planeta es un sitio colorido y bonito, lleno de formas geométricas y con un fondo espacial atractivo; pero se contrasta con una sensación de suspense en el ambiente.



GDD-GAME DESIGN

La historia

Los *Pykels* conviven en armonía en algún lugar del universo, pero una de estas criaturas se separa del grupo quedando sola. El equipo jugador deberá controlar éste personaje y el escenario para lograr reunir la pequeña criatura con su colonia de nuevo.

Puesto que en la primera versión del juego se implementará únicamente la primera fase, se define solo la introducción de la historia, dejando el núcleo de la historia y el desenlace abiertos para un desarrollo futuro.

Los personajes

Los *Pykels* son una raza ficticia, con aspecto similar a una célula; con influencias de *Flubber*, y *Zac* de *League of Legends*. Estas inofensivas criaturas de aspecto viscoso y elástico nacen a partir de la fusión de elementos en una galaxia desconocida.

El personaje principal será *Pykel*, una de estas criaturas a la que los jugadores del equipo deberán controlar para resolver los retos. *Pykel* contará inicialmente con un conjunto de habilidades básicas que se irá extendiendo con cada fase completa; siendo el set de habilidades para la versión inicial: movimiento lateral, y salto, complementado con la interacción del entorno con ratón y teclado.



GDD-ZAC REFERENCE



GDD-FLUBBER REFERENCE



GDD-PYKEL DESIGN

La perspectiva

Pykel es realmente un juego de plataformas, con la distinción de que es cooperativo, es decir, cada nivel del plataformas está pensado para ser resuelto en equipo. Esta combinación es similar a la de la saga *Trine*, con la diferencia de que en *Pykels* solo hay un personaje controlable; pero aún así se puede utilizar como referencia para la perspectiva al igual que *Limbo*.

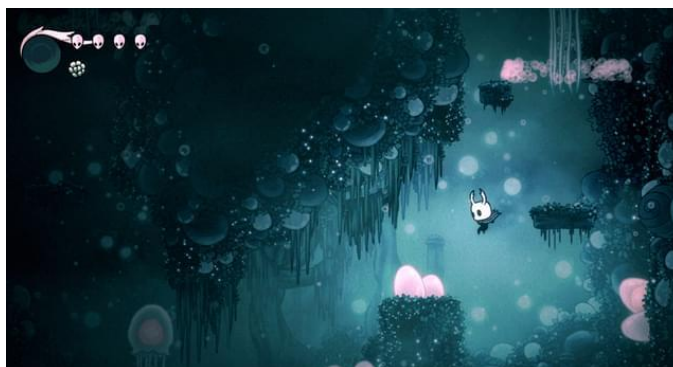
El juego se basa en la perspectiva 2D pero con sensación de profundidad utilizando una cámara ortográfica y la sobreposición de las capas del escenario. Las cinemáticas serán animaciones predefinidas dentro del mismo juego y se utilizará un fondo estilo parallax para la sensación de movimiento de la perspectiva.

Influencias

Como ya se ha comentado el videojuego tiene influencias específicamente definidas. El estilo de plataformas se referencia altamente en juegos indies como *Little Nightmares*, *Limbo* y *Inside*, y el diseño de puzzles y retos se basa en la saga de videojuegos *Trine*. A pesar de estas influencias se intenta consolidar, en el mismo estilo de plataformas que *Trine*, un videojuego mucho más orientado al trabajo en equipo, hasta el punto de que se generen informes con el análisis del equipo y su rendimiento.



GDD-LIMBO INGAME SCREENSHOT



GDD-HOLLOW KNIGHT INGAME SCREENSHOT

Mecánicas

La forma de interacción

El equipo interactuará mediante los periféricos de entrada (Ratón y teclado), siendo recomendable que cada jugador se asocie con una función específica y se vayan rotando dichas funciones de manera que todos participen al menos una vez en cada posición. Las funciones de interacción con el juego son:

Jugador	Interacción	Función
Jugador 1	Flechas dirección teclado	Movimiento del personaje
Jugador 2	Teclas alfabeto teclado	Activación de elementos
Jugador 3	Posición del cursor	Descubrimiento del escenario

GDD-INTERACCIÓN DE JUGADORES

Si el equipo consta de más de tres jugadores estas funciones deben asignarse de diferente manera en cada reto; el resto de jugadores deberán influir en la resolución de los retos planificando y coordinando las actividades del equipo.

El flujo de juego

Aquí se define el flujo completo del juego, desde el inicio de la aplicación hasta la finalización de la primera fase que constituye la versión inicial.

Inicio aplicación:

Se mostrará el logo del videojuego junto al título 3 segundos sobre un fondo negro, y luego mediante una transición de transparencia aparecerá el menú de inicio sobreimpreso mientras el fondo negro se convierte en el planeta.

Menú de inicio:

- **Jugar**

Se iniciará una nueva partida. Desaparece el menú de la pantalla y comienza el juego con una transición *Fading*.

- **Opciones**

Se permite la configuración del juego como el control de sonido, resolución, calidad gráfica, pantalla completa, etc.

- **Salir**

Salir del juego, guardando el progreso de la partida en curso. Se mostrará una ventana de confirmación previo al cierre de la aplicación.

Fase 1

- **Tutorial**

Éste es el primer escenario al empezar una nueva partida, donde aparecerán inscritas las instrucciones de control básicas. El nivel consta de cuatro partes.

- **Movimiento básico:** Con *Pykel* en el centro del escenario y un grabado con las instrucciones de movimiento (flechas de dirección) simplemente se debe avanzar hacia la derecha.
- **Salto:** Aparecerá un montículo al que se deberá saltar para continuar el avance. Sobre el montículo estará grabada la instrucción de salto (flecha arriba).
- **Cursor:** Llegados a este punto aparecerá la primera plataforma oculta, que habrá que descubrir con el desplazamiento del ratón sobre ella (sprite mask) y activar mediante las teclas correspondientes.
- **Finalización:** Con una última pantalla con diversas plataformas para enseñar a los jugadores las mecánicas de juego, al final de esta última parte del tutorial habrá un vórtice temporal a través del cual se accederá al siguiente nivel.

- **Nivel 1:**

En este primer nivel se introducen plataformas móviles, combinadas con las funciones ya presentadas en el tutorial (normales, ocultas y activables). Se pretende que el equipo coordine los controles del juego para llevar a *Pykel* al portal para acceder al siguiente nivel.

- **Nivel 2:**

En el segundo nivel de la fase inicial se introducen plataformas falsas, las cuales pueden parecer normales, pero bajo el sprite mask asociado al ratón se podrá comprobar que realmente son solo imágenes (sin collider) por lo que éstas plataformas deberán evitarse. También se incluyen plataformas temporales en las que tras un segundo desaparecerán un breve período de tiempo, para volver a aparecer mediante un sistema de respawn.

- **Nivel 3:**

El último nivel finaliza la primera fase del juego, en él se introduce un camino oculto por encima del escenario visible, al cual se deberá acceder mediante una plataforma oculta al inicio del nivel. En el recorrido de este camino existirán plataformas dinámicas de niveles anteriores, si el personaje cae del camino

oculto deberá volver al inicio para reintentarlo. Al final del camino un portal oculto permitirá finalizar la fase, volviendo al menú principal del juego.

- **Feedback:**

Al finalizar el último nivel de la fase se mostrará la interfaz del feedback con tres iconos para que el equipo transmita su opinión sobre el juego (malo, normal, bueno) seleccionando uno de ellos.

- **Evaluación:**

Al completarse el feedback del equipo la interfaz cambiará al cuestionario de evaluación, que deberá realizar el evaluador rellenando los campos con la mayor afinidad a lo observado sobre el equipo durante el juego. Este cuestionario podrá ser realizado en cualquier momento, quedando guardado en la partida asociada.

- **Informes:**

Una vez realizadas las evaluaciones se mostrarán los informes generados en base a la información del cuestionario. Dichos informes también estarán asociados a la partida e irán modificando los valores de las capacidades de cada jugador con la realización de nuevas fases.

La mecánica de juego

Aquí se definen detalladamente las mecánicas de *Pykels*. Es recomendable que cada mecánica se ejecute por un solo jugador en cada nivel, debiendo el resto participar en su resolución sin interactuar con el juego.

Movimiento básico: Por medio de las flechas de dirección del teclado se controlará el movimiento básico del personaje en plataformas; izquierda, derecha, y salto.

Letras teclado – control del escenario: Ciertas plataformas y elementos de los escenarios podrán tener asociada una tecla para su activación. El equipo deberá investigar y analizar los escenarios para identificar estos aspectos y planificar una solución para resolver los retos. Por ejemplo, plataformas con forma de O cuyo colíder se activa mediante el teclado.

Ratón – Revelador: durante todo el juego, el manejo del ratón será clave, teniendo este asociado un spritemask en su posición revelará

plataformas, caminos, y elementos ocultos. El cursor tendrá aspecto de nube de partículas que revela partes del escenario, esto implica una constante comunicación de los jugadores para la resolución de los retos.

Las Reglas de Juego

Se informará a los equipos con estas normas durante la introducción.

1. **Participación:** Todos los integrantes del equipo deben participar activamente en la resolución de cada nivel.
2. **Planificación:** Para cada nivel hay que dedicar unos minutos para analizar el escenario y elaborar un plan o idea, de manera que cada miembro del equipo tenga una tarea.
3. **Comunicación:** La comunicación entre el equipo es clave, se debe hablar para planificar cada nivel y durante el juego para informar de la situación de cada momento del juego.

Los retos

Los retos están diseñados para poner a prueba y enseñar las capacidades específicas de trabajo en equipo (Coordinación, Cohesión, Investigación, Creatividad, Especialización, Previsión, Ejecución, Iniciativa, Planificación).

Cada nivel se introducen nuevas mecánicas, buscando añadir el análisis de nuevas capacidades además de las de los retos anteriores.

Reto nivel 1 (Comunicación y coordinación): Se requiere una coordinación y timing a la hora de realizar ciertos movimientos, implicando al equipo en la comunicación y ejecución sincronizada. Se pretenden evaluar la coordinación y la comunicación.

Reto nivel 2 (Estimación y previsión): Se pretende que el equipo planifique y prevea las amenazas (plataformas falsas) para lograr superar el nivel. Esto implica al equipo en la estimación y previsión en la toma de decisiones

Reto nivel 3 (Planificación y exploración): El tercer nivel se orienta a la investigación del escenario por parte del equipo, y a la planificación de la solución para superar los retos del camino oculto una vez se encuentre. Al igual que en niveles anteriores se deberán coordinar las interacciones de los jugadores para lograr finalizar el recorrido y acceder al último portal de la fase.

Ámbito Educativo

Competencias de Trabajo en Equipo

A continuación, se describen las capacidades de trabajo en equipo a las que el juego de orienta, y los perfiles que se asocian a dichas capacidades. Nótese que dichas capacidades son evaluadas individualmente, pero en conjunto, pueden ser utilizadas para obtener datos sobre el rendimiento de un equipo, en base al equilibrio entre capacidades.

- **Coordinación:** capacidad de motivar y organizar al equipo, distribuyendo tareas de manera organizada; presenta madurez y confianza para asumir responsabilidades. Tiene siempre claros los objetivos y delega tareas eficazmente. Puede transmitir la imagen de controlador al resto del equipo, y puede delegar tareas que le corresponden para quitarse trabajo
Capacidad importante para perfiles de líderes, jefes de equipo, jefes de proyecto, coordinadores, directores, etc.
- **Cohesión:** capacidad para mantener el buen clima en el equipo, conversando y actuando de manera razonable. Perfil empático y mediador con buen humor y templanza. Cooperador y diplomático, escucha y evita enfrentamientos. Puede ser indeciso en ciertas situaciones o guardarse opiniones por no discutir. Asociado a perfiles de psicólogos, entrevistadores, negociadores, etc.
- **Investigación:** capacidad para obtener los recursos externos que el grupo necesita (información, alianzas, materiales, herramientas...). Perfil extrovertido y entusiasta con buena comunicación y relación con el entorno.
Puede perder el entusiasmo inicial si se encuentra con dificultades.
- **Creatividad:** capacidad para deducir soluciones ante adversidades del equipo. Perfil creativo e ingenioso, comparte ideas para solucionar adversidades o problemas difíciles.
Puede tener dificultades para comunicarse eficazmente, distraerse con facilidad o ser olvidadizo.
- **Especialización:** capacidad para aportar conocimientos específicos sobre una tarea particular. Perfil intelectual sobre la materia que centra la tarea del equipo, entregado e independiente.
Puede contribuir en áreas muy limitadas, o saturar al equipo con información menos relevante.

- **Previsión:** capacidad de valoración en la toma de decisiones, y estimación de riesgos. Permite recapacitar antes que el equipo se lance en una dirección. Perfil estratega, observador, visión objetiva, precavido.
Puede carecer de iniciativa, tener dificultades a la hora de tomar decisiones o ser excesivamente crítico.

- **Ejecución:** capacidad para transformar ideas en acciones. Perfil disciplinado, organizado y eficiente. Organiza al equipo para cumplir con el trabajo planeado.
Puede costarle adaptarse a nuevas ideas o ser poco flexible considerar posibilidades diferentes.

- **Iniciativa** capacidad para animar y ayudar a superar obstáculos en situaciones adversas. Perfil trabajador, con iniciativa, se crece bajo presión y ante retos difíciles.
Puede, en ocasiones, ser demasiado explícito o directo, y ofender a integrantes del equipo cuando pretende que las tareas se hagan.

- **Planificación:** capacidad de cumplir con los plazos y la calidad esperada. Perfil detallista y perfeccionista, revisa todo antes de darlo por finalizado buscando y puliendo errores.
Puede preocuparse excesivamente en ocasiones o ser demasiado perfeccionista no permitiendo que el equipo avance.

Repercusión

Este enfoque puede convertirse en el eje central de la educación en el futuro, donde el desarrollo de competencias y capacidades individuales en el contexto del trabajo en equipo sean las claves del aprendizaje.

Evaluaciones

Evaluación de resultados grupales e individuales, Evaluación de compañeros, Evaluación del software. De esta manera se genera un informe grupal del equipo y uno individual para cada miembro de este, destacando las capacidades fuertes y débiles, así como cuestiones a mejorar.

Las evaluaciones serán individuales en base a las capacidades mostradas por cada integrante; a partir de éstas se realizará una media de las capacidades del equipo y el rendimiento demostrado.

Por último, al finalizar la primera fase, que compone la versión inicial de este videojuego, aparecerá un feedback con iconos para recibir la opinión directa de los usuarios sobre que les ha parecido el juego, de manera general con 5 iconos correspondientes a: muy malo, malo, normal, bueno, muy bueno.

Cuestionario de evaluación por agente externo:

Cada capacidad definida contará con un rango de puntuación de 0 a 10, ampliable con puntos extra mediante cuestiones específicas. Este cuestionario se realizará sobre cada integrante del equipo.

Puntuación base de capacidades

Coordinación [0-10]

Cohesión [0-10]

Investigación [0-10]

Creatividad [0-10]

Especialización [0-10]

Previsión [0-10]

Ejecución [0-10]

Iniciativa [0-10]

Planificación [0-10]

Puntuación extra

Tareas asignadas/delegadas (Coordinación +2)

Conflictos resueltos (Cohesión +2)

Exploraciones externas (investigación +2)

Ideas propuestas (Creatividad +2, iniciativa +1)

Información útil aportada (Especialización +1)

Informes e interpretación de resultados

Feedback de Equipo:

Cada componente del grupo se representará por medio de contenedor, y aparecerá una nube de burbujas donde cada una definirá una frase o palabra asociada al juego o al trabajo en equipo; dichas burbujas deberán distribuirse en los distintos contenedores para asociarlas a cada jugador, de manera que al final cada uno tenga una lista de términos asociados que definan su experiencia de juego.

Esta característica junto con la evaluación e informes, se han dejado de lado durante el desarrollo de la primera versión del videojuego, ya que se consideró necesaria la colaboración con un especialista en el campo de la psicología para tratar de obtener los mejores resultados. También se tuvo en cuenta la audiencia objetivo, que al tratarse de alumnos en edad escolar, demandan un trato delicado en cuanto a evaluaciones psicológicas. Por todo ello se decidió finalmente posponer esta funcionalidad para versiones posteriores de *Pykels*.

Apartado Técnico

Motor Gráfico:

El motor gráfico será Unity, utilizando el tipo de proyecto 2D. La plataforma destino sera PC (Windows) y se construirán dos versiones, de x32 y de x64.

Gráficos: El juego se adapta a las distintas resoluciones del usuario, siendo la máxima 4k. La composición de gráficos del juego se basa en 3 capas (front, main, back), utilizando Sprites en formato png, con 100 ppu (Pixels Per Unit).

Audio: Se utilizan una clase AudioManajer y otra Sound para la gestión del sonido del juego, a través del Mixer de Unity. AudioManager permanecerá siempre en escena, evitando cortes de audio entre cambios de escenas, y se compondrá de un array de la clase Sound, a partir de los cuales se crearán los correspondientes AudioSources.

Plataformas: Se contará inicialmente con 4 tipos básicos de plataformas (I, V, O, L), derivando cada una en distintas variaciones. Se utilizará la siguiente nomenclatura:

Platform-(forma)-(variaciones)-(identificador)

Platform: Nombre base

(forma): L, O, V, I (forma base de la plataforma)

(variaciones): Activable (A), oculta (H), rotatoria (R), desplazable (M), falsa (F), temporal (T)

Por ejemplo: Platform-V-1 sería la primera plataforma del nivel con forma de V y básica; Platform-L-HAT-3 sería la tercera plataforma del nivel con forma de L oculta, activable y temporal.

Personaje: El personaje principal se formará por un RigidBody2D dinámico y dos colliders circulares 2D, uno para el cuerpo y otro para la base. Un script controlador se encargará de gestionar la interacción del jugador, los efectos gráficos, de sonido y la activación de plataformas mediante teclado. Por otro lado, se implementarán dos componentes hijos del personaje; un BoxCollider externo a modo de trigger para controlar las partes activas del escenario del nivel; y un objeto que defina cuándo el personaje se encuentra sobre el suelo. Para la gestión de animaciones se utilizará un componente Animator, con la siguiente máquina de estados.

Ratón: Se utilizará un objeto con spritemask y un script de seguimiento para el descubrimiento del escenario por medio del cursor. La activación de plataformas se implementa dentro del controlador del personaje

Escenario: El escenario de cada nivel se compone de 3 frames, donde cada frame equivale a dos pantallas de 4k (7680x2189), de esta manera el jugador recorrerá cada frame dos veces. Los tres frames de cada nivel están gestionados por un controlador que los desactiva cuando el jugador sale de ellos y reactiva cuando se acerca; con esto se pretende establecer un diseño de niveles eficiente que permita una posterior extensión de los mismos sin reparar en rendimiento.

Dentro de cada frame se distribuirán jerárquicamente los elementos del escenario (plataformas, decoración, capas, partículas...)

- Level
 - o Player
 - o MouseController
 - o LevelBase
 - Background
 - Particles (nubes y hojas)
 - o Frame-1-Controller
 - Frame-1
 - Platforms
 - Environment

- Frame-2-Controller
 - ...

Interfaz: Mediante el componente Canvas de Unity se implementan las interfaces del juego del menú principal y el menú de pausa. En el caso del menú de inicio compondrá una escena propia, mientras el de pausa se sobrepondrá oculto sobre cada nivel y se activará con la tecla escape, parando mediante dicha acción el juego (quedando en segundo plano tras un fondo transparente) y desactivando las instrucciones del nivel si las hubiera (para evitar superponer dos textos).

Gateway:

Cada nivel contará con un portal que definirá la superación del mismo, el objetivo es simple, llevar a *Pykel* al portal para pasar al siguiente nivel. Dicho portal se comunica con el *SceneManager* para ejecutar la transición al siguiente nivel. Nótese que tras el último nivel se una fase se volverá al menú inicial.

Transiciones:

La transición entre escenarios y niveles debe ser lo más fluida posible, siendo lo ideal que todo el flujo de juego sea continuo sin parones. Por ejemplo, como en *Limbo* donde el juego completo se puede completar sin pantallas de carga, parones. Para ello se utilizarán las animaciones de unity para crear efectos *Fading*. Mientras que debido al diseño de los niveles se podría implementar todo el juego en una sola escena de manera eficiente, ya que sólo está activo en cada momento el frame en el que se encuentra el jugador.

Cámara:

La cámara principal será ortográfica con un tamaño de 10, permitiendo de esta manera un renderizado limpio de los gráficos. Se utilizará el componente *cinemachine* para la gestión de la cámara; de manera que pueda seguir al personaje principal, pero sin salir de los límites de la escena (*CameraLimits*).

Se utilizará efecto *parallax* sobre la capa frontal (*front layer*) mediante los scripts asociados se permitirá desplazar dicha capa cuando el jugador se desplace, dotando de profundidad al juego.

Idioma:

Inicialmente el idioma será únicamente inglés, y no se contará con diálogos de juego. El idioma de desarrollo elegido también ha sido inglés debido a la nomenclatura de componentes de unity.

Recursos disponibles

Recursos hardware

Equipo de desarrollo

Torre: Aerocool Strike-X Air Gaming Series

Placa Base: MSI B250 Gaming M3

Procesador: Intel Core i5-7400 3.0GHz

Memoria: 8GB RAM

Nvidia GTX 1050-TI

SSD 250GB – HDD 1TB – HDD 2TB

Recursos software

Unity (Motor gráfico)

Cinemachine

TextMeshPro

Visual Studio Code (Scripting)

Photoshop (Game Art)

Word Office (Documentación)

Draw.io (Diagramas)

Apartado Artístico

Estilo Artístico

Todo el apartado artístico será diseñado de cero, no se utilizarán directamente recursos gráficos externos.

El estilo artístico general será el minimalista, pero orientado al público infantil, se utilizarán figuras geométricas básicas para el diseño de escenarios, pero con colores llamativos y gradientes. Se deberán diseñar sombras en base a la luz de cada escenario. Se tomarán como referencias artísticas juegos como *Limbo*, *Inside* pero dotando las escenas de colorido para atraer a los jugadores.

Interfaces de Usuario



GDD-MENÚ PRINCIPAL



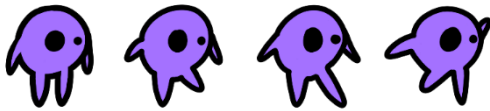
GDD-MENÚ DE PAUSA

Sprites

Pykel



GDD-IDLE SPRITES

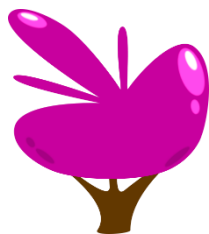


GDD-WALK SPRITES

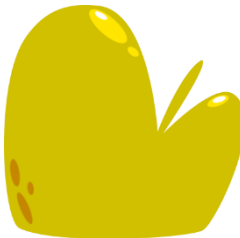
Entorno



GDD-TREE-1



GDD-TREE-2



GDD-BUSH-1



GDD-ROCK-1

Plataformas



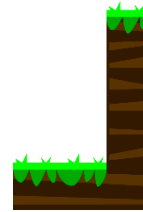
GDD-PLATFORM-I



GDD-PLATFORM-V

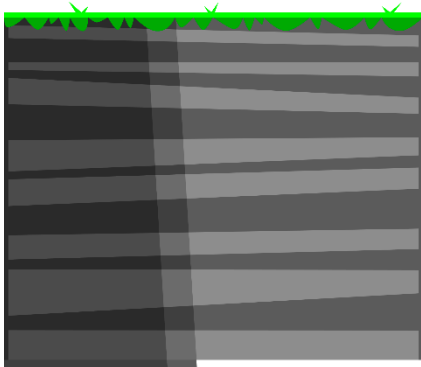


GDD-PLATFORM-O

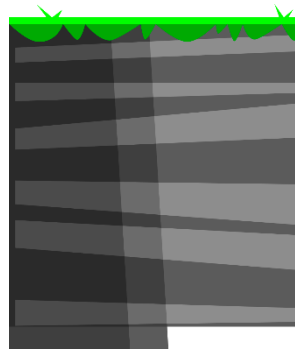


GDD-PLATFORM-L

Bloques



GDD-BLOCK-1



GDD-BLOCK-2

Suelo principal



GDD-Main Floor

Animaciones

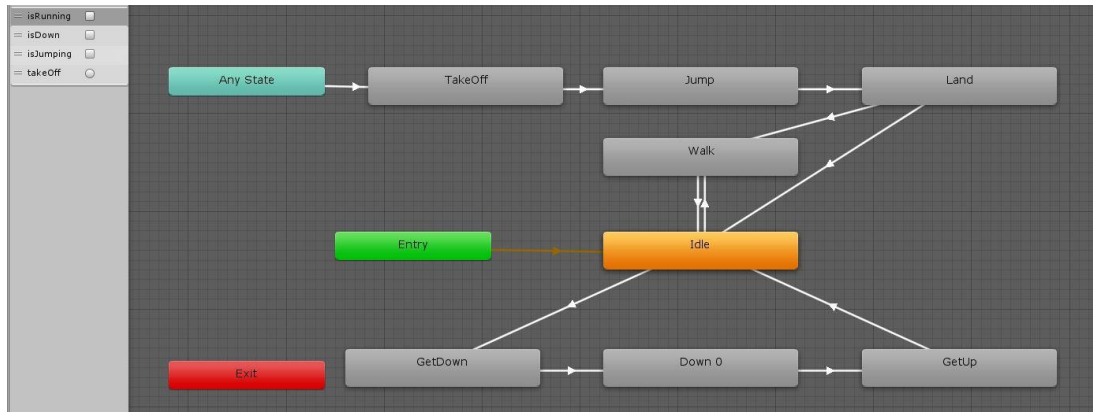
Pykel

Idle: animación de *Pykel* por defecto cuando está parado. Dotará de vida al personaje que pestañeará cada tres segundos.

Walk: animación de movimiento, animará a *Pykel* cuando se desplace en el eje horizontal.

Jump: animación de salto, constará de tres partes (despegue, salto y aterrizaje), se utilizará la flecha arriba del teclado y un parámetro *grounded* para definir el salto del personaje.

Down: animación de agachado del personaje, carece de funcionalidad, pero se implementó en un prototipo inicial y se decidió dejarla porque resulta divertida y dota de personalidad al personaje.



GDD-PYKEL ANIMATOR

GUI

FadeIn: Transición del canvas negro a transparente y desactivado.

FadeOut: Transición del canvas transparente a negro y activado.

PauseAnimation: Transición del canvas de desactivado a negro semitransparente

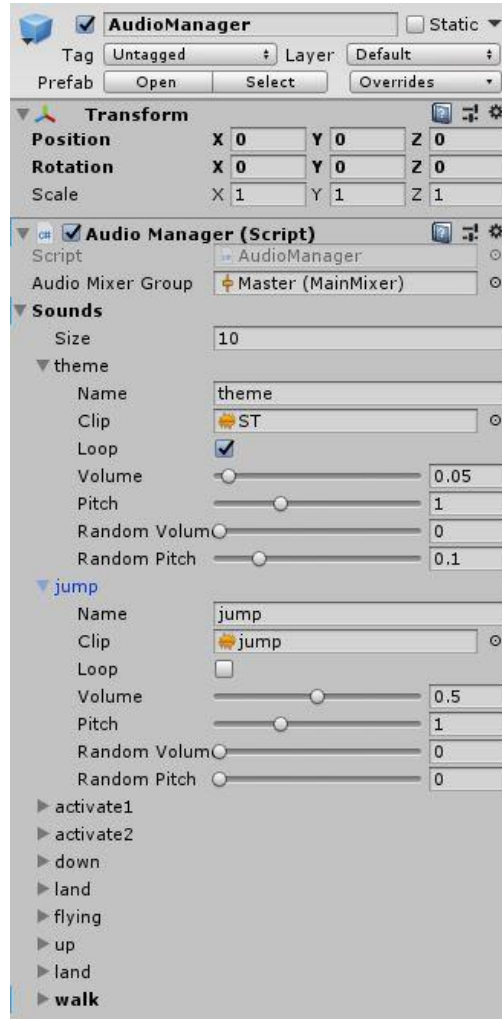
Audio

Todo el audio se ha generado y compuesto para el juego, no se utiliza música o efectos de sonido externos.

La banda sonora principal deberá ser calmada y relajante, con instrumentos clásicos y melodías simples pero pegadizas. También es necesario disponer de melodías específicas y sonidos de efectos especiales para dotar de realismo e inmersión al juego.

Audio ambiente: Soundtrack principal (theme).

Efectos de sonido: Activate, jump, flying, land, up, down, walk.



GDD-AUDIO MANAGER

Referencias

[Blackthornprod \(Youtube channel\)](#)

[Brackeys \(Youtube channel\)](#)

[TomasBush \(Youtube channel\)](#)

[Photoshop tutorials](#)

[Unity documentation](#)

[Documento de diseño info-1](#)

[Documento de diseño info-2](#)

[Documento de diseño info-3](#)

[Documento de diseño info-4](#)

[Documento de diseño ejemplo-1](#)

[Documento de diseño ejemplo-2](#)

[Documento de diseño videotutorial](#)

[juegos de lógica](#)



UNIVERSIDAD
DE MÁLAGA

| **uma.es**

E.T.S de Ingeniería Informática
Bulevar Louis Pasteur, 35
Campus de Teatinos
29071 Málaga

E.T.S. DE INGENIERÍA INFORMÁTICA