TFG

INFINITE ANDROMEDA.

VIDEOJUEGO DE PLATAFORMAS 2D: GUNS AND FIGHTS

Presentado por Sergio Espino Palma
Tutor: Francisco José Abad Cerdá

Facultat de Belles Arts de Sant Carles
Grado en Diseño y Tecnologías Creativas
Curso 2019-2020

RESUMEN Y PALABRAS CLAVE

Proyecto que abarca el desarrollo multidisciplinar de un videojuego 2D, incluido en los géneros de plataformas, peleas y mazmorras. Este documento reúne todo el proceso de trabajo, desde la concepción de la idea hasta la creación de una versión de prueba, pasando por el diseño e implementación de todos los elementos necesarios para su funcionamiento. No como una versión final en sí misma, sino como una demostración de lo que podría llegar a ser del producto acabado. En cada partida el jugador se adentra en una nave espacial generada de forma aleatoria, con el objetivo de sobrevivir enfrentándose a los enemigos que vayan apareciendo hasta que encuentre la salida.

Videojuegos, Plataformas, Roguelike, Peleas, “Píxel Art”, 2D, Acción

ABSTRACT AND KEY WORDS

The project covers the multidisciplinary development of a 2D videogame included in the platform, fight and rogue-like genre. This document holds the whole working process, from the conception of the idea to the creation of a fully operational trial version, as well as the design and implementation of all the required elements for its functioning. It shouldn’t be taken as a final version but as a demonstration of what the finished product could be. In each game the player enters a randomly generated spaceship and his goal is to beat all the enemies that appear until he finds the exit.

Videogames, Platforms, Rogue-like, Fight, “Pixel Art”, 2D, Action

CONTRATO DE ORIGINALIDAD

Este Trabajo de Fin de Grado ha sido realizado íntegramente por el alumno Sergio Espino Palma. Este es el último trámite para la obtención del título de la promoción 2016/2020 del Grado en Diseño y Tecnologías Creativas de la Universidad Politécnica de Valencia.

El presente documento es original y no ha sido entregado como trabajo académico previo, y todo el material tomado de otras fuentes ha sido citado correctamente.

Firmado a día: **16** de **Noviembre** de **2020**

ÍNDICE

1. INTRODUCCIÓN

1.1. PRESENTACIÓN Y JUSTIFICACIÓN DEL TRABAJO

Infinite Andromeda es el nombre de la propuesta de trabajo realizada y descrita en esta memoria, de carácter individual, desarrollada desde un punto de vista puramente profesional, en este caso el de un desarrollador independiente.

Esta propuesta, de hecho, está pensada para traspasar el ámbito de lo académico, afianzando su condición profesional. Ya que el videojuego fue diseñado para continuar con su desarrollo y expansión en un futuro próximo, para poder finalmente ser lanzado al mercado. Gracias a este diseño "expansible", durante los próximos años se podría ampliar con gran facilidad el producto aquí presentado, añadiendo nuevos personajes, u otros escenarios como templos o refugios subterráneos.

La construcción de esta "prueba jugable" (a la que a partir de ahora llamaremos simplemente, 'demo'), es un proyecto ideal como demostración de los conocimientos obtenidos durante la formación. No solo es una muestra, sino que este ejercicio también servirá para consolidar todo ese aprendizaje.

Esta memoria recoge, de manera de forma ordenada y secuencial, el conjunto de procesos de trabajo, técnicas y materiales utilizados, referentes visuales e inspiraciones, que han formado parte o tenido una influencia en el flujo de trabajo, y por tanto tienen un impacto o han dejado huella en el producto final.

1.2. OBJETIVOS

1.2.1. Clasificación de los Objetivos

A la hora de plantear y establecer los objetivos que persiguen el diseño del videojuego, cabe destacar dos aspectos: uno es que todas las decisiones de diseño, se han tomado buscando lo mejor para el juego y lo más práctico (*más práctico* se traduce en *más fácil de implementar,* y por tanto, *más tiempo de diseño/desarrollo*); y el segundo aspecto es que no todos los objetivos son de la misma naturaleza, por lo que se han subdividido en dos tipos, dependiendo de si pueden ser fácilmente comprobados o no.

Objetivos Materiales: Estos son los objetivos más fáciles de probar, ya que los podemos asociar directamente con un número o comprobar si se han implementado. Cumplen funciones concretas: permitir al jugador moverse;

elaborar todos los fondos para las habitaciones de la nave; animar la lista de acciones del personaje; etc.

Los objetivos establecidos en esta categoría son los siguientes:

-*Elaborar un* ***producto*** *que sea* ***completamente funcional****.*
-***Construir*** *un* ***modelo de juego*** *fácilmente* ***expansible****.*
-*Diseñar e implementar un* ***sistema generador de niveles****.*
-***Permitir al jugador moverse*** *por el entorno de una forma satisfactoria*.

Objetivos Conceptuales: Este segundo tipo pertenece a una categoría más abstracta, y por ello su revisión o control requiere un estudio más exhaustivo. Los objetivos conceptuales se refieren a cuestiones como sentimientos o nivel de interés. Dentro de este grupo se encuentran los siguientes objetivos:

-*Definir un* ***estilo atractivo*** *y eficaz.*
-*Ofrecer una* ***experiencia adaptable****.*
-***Estudiar e imitar*** *elementos de otros videojuegos.*
-***Aplicar*** *las* ***técnicas y metodologías*** *aprendidas durante la formación.*

Ambos objetivos materiales y conceptuales son ampliamente detallados y explicados en el apartado 3.4.2. Definición del diseño de acuerdo con los objetivos.

1.3. METODOLOGÍA

Los conocimientos previos, fundamentales para el diseño y producción de la 'demo', incluyen diversas áreas como: un amplio dominio y comprensión de los principios del diseño y la teoría del color, que permitirán definir un estilo cohesionado y atractivo a la hora de diseñar escenarios y personajes. Dicho personaje posteriormente tendrá que ser animado, por lo que el diseñador también desempeñará la función de animador; todo esto se añade además a la destreza y experiencia necesaria como programador, que será indispensable para la construcción del prototipo que reúna y haga uso de los materiales producidos.

En este apartado, nos centraremos en el cómo se ha estructurado y ordenado todo el proceso de trabajo, así como las técnicas y materiales empleados, organización de las tareas, y una descripción de las principales etapas del desarrollo.

1.3.1. *Organización del proyecto*

Cuando se plantea la elaboración de un proyecto como éste, en el que intervienen múltiples disciplinas, es de vital importancia que las distintas partes involucradas actúen siguiendo un plan estudiado antes de comenzar a trabajar, lo que permitirá realizar un trabajo en cadena perfectamente sincronizado. Por

exponerlo de una forma simple, el animador no puede empezar a trabajar hasta que no haya una narrativa o un personaje definido, los programadores tampoco sabrán qué programar si no hay un juego diseñado.

Esto explica por qué algunos procesos deben realizarse antes que otros, pero también hay fases de producción que tienen que ser paralelas y se deben realizar de forma simultánea. Para que el producto sea coherente todas sus partes deben de estar cohesionadas, por lo que es imprescindible que etapas como la animación y programación del personaje, tengan lugar de manera coordinada en un proceso de intercambio constante de información, buscando la sintonía entre todos los elementos.

Una vez aclarado esto, procedemos a enumerar las tres etapas diferenciadas del proyecto.

1.3.2. *Etapas del proyecto*

1.3.2.1. Diseño del videojuego

Fase donde se termina de definir la idea y se establecen los límites del proyecto. Posteriormente se realiza una esquematización de la idea o **subdivisión** que resultó en **dos elementos principales**.

Uno de esos elementos extraídos del diseño, engloba todas las mecánicas del **escenario** y la **generación del mapa**, *Diseñar e implementar un sistema generador de niveles.*

El segundo componente se encarga de cubrir el otro aspecto fundamental de la 'demo', que es el **protagonista** y las **mecánicas de combate**, *Permitir al jugador moverse por el entorno de una forma satisfactoria*.

Por lo tanto, el diseño se centrará en un estudio de referentes visuales que permitirá definir un estilo propio, deberá concretar el funcionamiento del generador de niveles, todas sus partes y normas, además de determinar la lista de acciones o capacidades del jugador.

Hasta este punto del trabajo, el estudio y los primeros bocetos del diseño se llevaron a cabo haciendo uso de medios exclusivamente analógicos, utilizando láminas Din A4 para abocetar, dibujar o escribir, y un portaminas de grosor 0.7mm, dureza 2H.

Podríamos identificar esta etapa con la tradicionalmente conocida como 'Preproducción'.

1.3.2.2. Desarrollo de la 'demo'

Concluida la fase de planteamiento y siguiendo los resultados obtenidos al esquematizar nuestro diseño, se establecieron dos líneas de trabajo que se desarrollarían paralelamente, de forma simultánea y en proyectos independientes:

-El **personaje**, será el encargado de recopilar y **trasladar al juego la información** que el jugador introduzca en la máquina al pulsar los botones. Dependiendo del botón que se haya pulsado, y siguiendo unas normas que tendremos que

implementar, el personaje tendrá que reaccionar dentro del juego, lo cual también implica que tendremos que **definir las reacciones posibles y asociarles una animación**.

De esta manera podemos dividir todo el desarrollo de este componente en dos áreas fundamentales: la **programación**, que se ha llevado a cabo usando *Unity 2019*[1], del código donde se establecen las acciones que puede tomar el jugador y sus reacciones; y la **animación**, que ha sido producto de la combinación de dibujos en papel, posteriormente escaneados y animados de manera digital. Los programas empleados en este último proceso han sido: *Photoshop CS6*[2], para modificar los dibujos y corregirlos con mayor facilidad; y *Adobe Animate 2019*[3] para realizar todas las animaciones.

-El **escenario**, que cumplirá con funciones fundamentales para el videojuego, será el **gestor de la partida** y **proporciona al protagonista un espacio** para moverse. Estará compuesto por dos partes que nuevamente pertenecen a dos ámbitos diferentes: una parte "física" y visual, compuesta por el diseño de las habitaciones en sí mismas; y otra segunda parte, conformada por la programación del conjunto de normas que gestionan la partida o colocan las habitaciones.

Decimos que las habitaciones son la parte física del mapa, porque brindan todos los elementos con los que el personaje puede interactuar físicamente. Estos componentes también se encargan de la **parte estética** del mapa, ya que incorporan la imagen correspondiente a cada habitación y renderiza[4] los objetos de su interior en el orden correcto. Para elaborar el píxel-art[5] para los fondos de las diferentes habitaciones, se ha utilizado de nuevo la herramienta ***Photoshop CS6***.

Por el otro lado tenemos toda la parte que no se puede ver en el juego, el **sistema generador de niveles**. Este sistema proporciona un espacio para moverse porque coloca todas las habitaciones de manera que estén conectadas y todas sean accesibles. Además gestiona la partida porque debe ser capaz de crear un nuevo mapa y comenzar la partida automáticamente. El sistema se ha construido en ***Unity*** también.

---

[1] *Unity* es uno de los mejores motores de videojuegos, además de que su uso es gratuito mientras que los desarrolladores no tengan ingresos anuales más de 100.000 euros. https://unity.com/es

[2] *Photoshop CS6* es un editor de fotografía profesional desarrollado por *Adobe Systems Incorporated* utilizado principalmente en nuestro caso para el retoque y edición de los bocetos en papel. https://www.adobe.com/es/products/photoshop.html

[3] *Adobe Animate 2019* es otro producto de la casa *Adobe Systems Incorporated. S*e utiliza para manipular y crear gráficos vectoriales a través del uso de frames. https://www.adobe.com/es/products/animate.html

[4] El término renderización es un anglicismo para representación gráfica, usado en la jerga informática para referirse al proceso de generar una imagen a partir de un modelo 2D o 3D por medio de programas informáticos.

[5] Se denomina Píxel Art al diseño artístico digital basado en píxeles.

Esta etapa contiene todos los procesos conocidos como la 'Producción', por lo que con su implementación sólo quedaría una fase de ajustes y correcciones.

1.3.2.3. Refinado del prototipo

Finalmente, las dos líneas de trabajo convergen en un único proyecto, la primera versión "completa" del videojuego. A partir de esta versión se afinan todos los aspectos menos trabajados en las etapas anteriores, las transiciones de la cámara, una prueba en vivo de los usuarios, y demás cuestiones estéticas que puedan mejorar el producto con la realización de cambios relativamente pequeños.

1.3.3. *Organización de las etapas del proyecto*

Fig. 1. Línea de tiempo, muestra la organización de las etapas del proyecto.

Cabe destacar algunas áreas del trabajo transversales, por tanto, no están incluidas en ninguna etapa en concreto. Son los casos de tareas como el desarrollo de la memoria escrita, gestada durante todo el proceso de trabajo siguiendo fielmente toda la producción. También es el caso del examen continuo del código generado, así como de todos los materiales producidos en el diseño del escenario y las animaciones.

2. EL GÉNERO DE MAZMORRAS O 'ROGUELIKE'

2.1 POR QUÉ HABLAMOS DE 'ROGUELIKE' [6]

A pesar de que Infinite Andrómeda es un juego que podría ser clasificado dentro de muchos subgéneros, a la hora de valorar cuál de ellos tiene una mayor

[6] Los videojuegos 'Roguelike' o de mazmorras son un subgénero de los videojuegos de rol, que se caracterizan por que el personaje tendrá que entrar en una mazmorra repleta de enemigos que ha sido generada siguiendo procedimientos aleatorios.

influencia en el proyecto, sin duda sería el género de los 'Roguelike', además de ser el más interesante a la hora de generar una discusión.

Dentro de los anexos se incluye un documento que establece una introducción y repasa los orígenes y evolución del género (**Historia, Origen y Evolución de los Roguelike**).

2.2 CARACTERÍSTICAS ÚNICAS DEL GÉNERO

El género 'roguelike' en su subgénero derivado de los videojuegos de rol, engloba a una categoría de juegos que comparten una serie de características comunes, como la muerte permanente del personaje, o que cada nivel contiene un número de habitaciones determinadas, que pueden variar en forma o tamaño, pero siempre contendrán enemigos o tesoros en su interior. Aunque existe una definición mucho más rigurosa y estricta, que determina si un juego pertenece realmente a esta categoría, llamada *'The Berlin Interpretation 2008'* [7] , constituida por los asistentes a la *"Conferencia Internacional de Desarrolladores de Roguelikes"*[8]. A continuación, se nombran sólo las características que nos interesa incorporar a nuestro diseño:

-**Muerte permanente del jugador**. Cada partida el jugador comenzará desde el mismo punto, sin ningún tipo de progreso o ventaja. El jugador en la 'demo' no tiene la capacidad de morir porque no puede ser atacado, pero mantiene la filosofía de "nueva vida, nueva partida".

-**Mapa generado aleatoriamente**. La generación procedural del mapa es un aspecto clave que acompaña a los 'Roguelike' desde sus orígenes y será un elemento principal del desarrollo de nuestro producto.

-**Mapa compuesto por habitaciones**. La forma y tamaño del mapa deberán variar de partida a partida, pero siempre estará formado por habitaciones de contenido variable cerradas y conectadas entre sí.

-**Centrado en el "combate"**. La acción principal del videojuego se centra en los encuentros del personaje con distintos enemigos cuando éste entra a una nueva sala. Las características de este combate varían dependiendo del género, pero suele ser por turnos[9]. Esa atención especial al combate formará una gran parte de nuestra propia versión.

---

[7] *"La interpretación de Berlín"* es un convenio que establece una serie de características o atributos que debe tener un juego para ser incluido dentro del género 'Roguelike'. http://www.roguebasin.com/index.php?title=Berlin_Interpretation

[8] Esta conferencia tiene lugar en un evento anual que dura dos días, donde todas las personas interesadas pueden debatir y discutir sobre el género. http://www.roguebasin.com/index.php?title=IRDC

[9] Los videojuegos basados por turnos paran el tiempo entre las distintas acciones que se toman en el desarrollo de una partida. Generalmente cada turno o ronda de juego suele tener una serie de partes o fases diferenciadas.

2.3 INNOVANDO DENTRO DEL GÉNERO

A la hora de desarrollar un producto, cuyo objetivo último es la publicación y venta en el mercado actual, se debe tener muy en cuenta durante su concepción, cuáles son las características principales o posibilidades que ofrecen sus competidores más directos.

En el caso concreto que está siendo objeto de estudio, los 'Roguelike', ofrecen en general un conjunto de opciones estándar o que son relativamente comunes entre todos ellos.

No obstante, los juegos de este tipo más destacados de las últimas décadas han optado por fusionarse con otros de diferente naturaleza, como es el caso de los shooters[10], las plataformas y los juegos de acción. De la misma manera se han seleccionado los siguientes elementos de otros ámbitos para ser incluidos en la propuesta:

-**Combate basado en un sistema de combos**[11]. El jugador tendrá que efectuar secuencias de ataques ordenados de manera coordinada para eliminar a los objetivos.

-**Uso de armas de fuego**. Además del sistema de ataques cuerpo a cuerpo, el jugador podrá cambiar de modo de combate por completo, activando un arma de fuego.

-**Gestión de recursos limitados**. Es el jugador el que tiene que elegir en cada combate cómo hacer uso de esos recursos. Estos recursos serían la munición de las armas o el combustible.

3. DESARROLLO

3.1. BRIEFING

La construcción de esta 'demo', es un encargo de naturaleza autoimpuesta que se desarrollará de manera individual, con el objetivo de crear, desde un punto de vista estrictamente profesional, un producto que sea capaz de demostrar el potencial del videojuego presentado. No como una versión final, sino como un "adelanto jugable".

La jugabilidad y la acción principal del videojuego se centrará en los combates del personaje a medida que explora el interior de una nave abandonada. Este producto ha sido desarrollado con el objetivo de apelar a un

[10] Género de acción, normalmente el principal objetivo del videojuego consiste en disparar y matar enemigos.

[11] Un combo es una combinación de acciones que se tienen que realizar siguiendo un orden, casi siempre hay un tiempo límite entre acción y acción para añadirle dificultad.

público objetivo [12]acostumbrado a jugar a videojuegos, juvenil o adolescente, que sean usuarios habituales de géneros del tipo de plataformas, acción, peleas, dificultad y que ofrezcan un reto.

A la hora de comercializar el título, éste será evaluado por las compañías distribuidoras [13]y se establecerá una edad recomendada (véase fig. 3.), aunque se propone una clasificación PEGI 12[14] debido a un cierto nivel de violencia.

Fig. 2. Logotipo de la organización.

Fig. 3. Ejemplos de tipos de clasificación.

3.1.1. Límites del proyecto y plazos establecidos

El proyecto contempla todo el proceso requerido para construir una demostración completamente funcional del videojuego, desde el diseño de toda la estructura del videojuego, a la implementación y elaboración del código, animaciones y fondos necesarios. Se dejan fuera aspectos como sonido, vfx[15] o inteligencia artificial, que se escapan de las posibilidades que ofrece un único desarrollador para todo el trabajo.

Pero la propuesta no sólo se limita a recoger la producción del producto presentado, sino que en numerosas ocasiones también se pone de manifiesto la intención de continuar con el desarrollo del videojuego, proponiendo posibles líneas de actuación futuras para su expansión.

El plazo escogido para la realización de esta 'demo', empieza prácticamente con el curso académico 20-21, el día 2 de septiembre, y se llevará a cabo en un periodo de 14 semanas, hasta la fecha de vencimiento de los plazos establecidos por la facultad, 9 diciembre. El resultado será expuesto en forma de demostración en vivo desde un archivo ejecutable.

---

[12] El público objetivo es un concepto utilizado generalmente en la publicidad que se refiere a la representación del cliente ideal para el producto en cuestión.

[13] Las compañías distribuidoras son empresas encargadas distribuir videojuegos que ya han sido desarrollados. Aunque suele ser normal que la misma compañía realice también sus propios productos.

[14] La clasificación PEGI, es un mecanismo que utiliza la Unión Europea para regular y orientar a los consumidores sobre si un producto es adecuado o no para niños, no hablando desde un punto de vista de dificultad sino de sensibilidad. https://pegi.info/es/node/19

[15] VFX son las siglas empleadas para referirnos al apartado de los efectos especiales, como el uso de filtros o añadir partículas.

3.2. ANÁLISIS DE REFERENTES

3.2.1. Estudio de los referentes

En esta sección estudiaremos tres videojuegos que pertenecen a categorías muy distintas entre sí, pero sin duda todos ellos han destacado dentro de su género y sabido aprovechar al máximo las oportunidades que cada uno ofrece, convirtiéndose así en referentes indiscutibles.

El extenso proceso de estudio de los referentes, acompañado por una introducción a cada uno de estos juegos, está recogido dentro de los anexos (**Estudio en profundidad de los referentes históricos y visuales**). Debido a los límites del trabajo, en ese documento se explican con todo lujo de detalles los elementos más interesantes de los títulos seleccionados y la implicación que todo esto tiene en las decisiones de diseño tomadas.

A continuación, se presenta la síntesis de todas esas decisiones de diseño obtenidas a través de los referentes, aunque como ya se ha indicado, éstas se encuentran mucho mejor explicadas y más detalladas en el anexo.

Fig. 4. Crawl (2014).
Fig. 5. Jetpack Joyride (2011).
Fig. 6. Enter the Gungeon (2016).

3.2.2. Decisiones de diseño tomadas a partir del estudio de referentes

A nivel de mecánicas de juego, se estableció que el escenario estará cerrado y ocupa el espacio de la pantalla, lo cual nos permitirá profundizar en las mecánicas de combate. El protagonista tendrá una serie de ataques únicos con diferentes beneficios, y también podrá hacer uso de unos recursos limitados durante el combate para añadir mayor profundidad.

En cuanto al apartado estético de los personajes, éstos estarán muy simplificados, casi reducidos a formas básicas (véase fig. 4. 5. y 6.), por lo que deliberadamente no son muy fieles a la anatomía humana. Incluso se pueden llegar a romper algunas normas de la animación, para hacer énfasis en algún punto concreto de la acción mostrada.

Por último, en relación a la estética de los escenarios, se tomó la decisión de adoptar un diseño modular que resulte sencillo pero efectivo. Los fondos no tendrán mucho detalle para no desviar la atención de la acción principal, además de mejorar la producción. El diseño de las habitaciones será modular, ya que todas las combinaciones deben ser visualmente coherentes y estar igual de trabajadas.

3.3. ESTUDIO DE DISEÑO

Para obtener información fiable, que nos permita tomar unas buenas decisiones en el diseño del videojuego, que sean acertadas y coherentes con sus potenciales clientes, su público objetivo, se ha realizado una encuesta en la que han participado 18 personas. El cuestionario realizado ha sido descrito y explicado ampliamente junto a sus partes y objetivos en un documento adjunto a esta memoria en los anexos (**Estudio del Público Objetivo a través de la Encuesta**).

Analizando de forma general las respuestas de los usuarios (mostradas en el documento referenciado), apuntan con toda seguridad al potencial de la propuesta presentada, lo cual es un aspecto muy positivo que animó a seguir adelante con el desarrollo, reforzando la confianza en el proyecto. El principal aspecto que ha sido destacado es el interés de los jugadores por conocer esta fusión de géneros, que deberá adoptar por un estilo propio tanto a nivel artístico como de jugabilidad.

3.3.1. Decisiones de diseño tomadas a partir de la encuesta

Los resultados obtenidos indican cómo los aspectos más importantes y que deberían tener un mayor peso, serían: la jugabilidad, que es el pilar fundamental de nuestro juego; el apartado gráfico, que tendrá que competir con productos desarrollados por profesionales en el sector; la dificultad, que sin llegar a ser injusta o abrumadora tiene que plantear un reto para los jugadores más experimentados; y finalmente la re-jugabilidad[16], que se consigue al ofrecer alternativas de personajes, ataques y escenarios al jugador, y hace que la experiencia resulte mucho más variada y entretenida, mientras aumenta el grado de interés de los compradores.

Los géneros que han sido seleccionados como los más interesantes para formar parte del juego, son los de estrategia, disparos y peleas. Seguidos por los puzzles[17], las plataformas y los juegos de estilo arcade. De todos estos sólo descartamos los puzzles, ya que no se encontró una buena forma de implementar esta característica en el producto de forma natural, aportándole algo que sea bueno o interesante para la experiencia.

Para responder a la demanda que han realizado los usuarios, de centrar el 'gameplay'[18] en una acción muy acelerada, con un cierto grado de descontrol que te obligue a improvisar, se encuentra también la necesidad de encontrar pequeños momentos de tranquilidad, que le permitan al jugador establecer un plan. Por lo tanto, el diseño se encargará de proporcionar ambas situaciones: cuando el jugador entre a una sala donde nunca antes ha estado comenzará un combate a muerte de forma inevitable, obligándole a reaccionar en ese momento. Una vez el jugador haya eliminado a todos los enemigos, las puertas de la sala se desbloquearán y le permitirán avanzar a la siguiente habitación. Desde que termine el combate hasta que el jugador decida avanzar a la siguiente sala, puede tomarse todo el tiempo que necesite para recuperarse de su último combate, administrar sus recursos o cambiar de rumbo.

---

[16] Re-jugabilidad se refiere al término empleado en el mundo de los videojuegos, que representa el potencial de un título para seguir siendo jugado una y otra vez.

[17] Los videojuegos denominados como, "videojuegos de puzzles" plantean una serie de retos o problemas al jugador que éste deberá resolver siguiendo una lógica que puede estar basada o no en la realidad.

[18] El 'gameplay' es el conjunto de acciones y mecánicas que interactúan en el desarrollo de una partida.

El elemento más solicitado ha sido el de la estrategia, con un 83 '3% de los usuarios, que pedían esa "necesidad" de hacer pensar al jugador y obligarlo a organizar o planificar la partida. Dentro de nuestra propuesta, como ya se ha comentado en varias ocasiones, se traducirá en el sistema de combos y un sistema de administración de recursos. Estos recursos se situarán al máximo al comienzo de cada nuevo combate, lo que significa que el jugador puede hacer un uso total de esos recursos cada vez que tenga que pelear. Si el límite de recursos fuera mucho mayor, pero éstos no pudieran ser repuestos y por tanto, fuesen los restantes hasta el final de la partida, se darían muchas situaciones donde los jugadores no hacen uso de todas sus capacidades y mueren por almacenar estos valiosos recursos, en caso de que los necesiten en el futuro, generando así frustración. De este modo, se considera que es mucho mejor asegurar que el jugador pueda hacer un uso más limitado de los recursos que dispone de manera regular, antes que permitir la gran mayoría de su uso en un pequeño espacio de tiempo.

3.4. DISEÑO

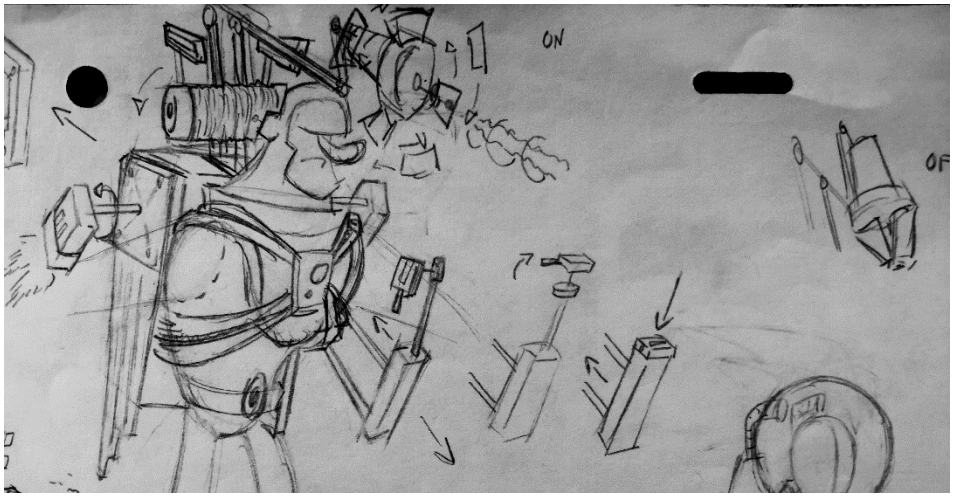


Fig. 7. Primeros diseños del personaje.

3.4.1. Concepción/Origen de la idea

La creación de Infinite Andromeda y los primeros bocetos (véase fig. 7.), nacen como una respuesta ante una necesidad en el mercado, que se considera que actualmente no está cubierta.

Por un lado, nos encontramos con los videojuegos que se centran en las peleas, hablamos de sagas como "Street Fighter"[19], Tekken [20]o "Smash Bros"[21],

[19] Street Fighter es una saga de juegos desarrollada y distribuida por la compañía *Capcom*. https://streetfighter.com/street-fighter-30th-anniversary-collection/

[20] *Tekken* es una saga de juegos desarrollada por *Namco* y distribuida por *Namco Bandai Games.* https://es.bandainamcoent.eu/tekken

[21] *Smash Bros* es la saga de peleas más famosa de la distribuidora *Nintendo*, sus desarrolladores han ido cambiando con el paso de los años. https://www.smashbros.com/es_ES/

todas ellas disponen sistemas de combate de los mejores, alabados por miles y miles de jugadores desde hace décadas. Por otro lado, nos encontramos con videojuegos del tipo 'roguelike', algunos de ellos desarrollados por dos o tres personas, diseñados de una forma tan inteligente y acompañados por algunas estéticas tan únicas y exquisitas, que son capaces de trasladar al jugador a mundos inimaginables.

Planteando los hechos de esta forma, la respuesta es tan clara que parece imposible que ningún videojuego haya optado por esta opción. De hecho, es verdad que muchos juegos han recorrido este camino, pero siempre de la forma incorrecta. Los videojuegos de mazmorras que utilizan sistemas de combate en tiempo real utilizan mecánicas bastante sencillas, que normalmente se resumen en pulsar un botón por ataque y no ofrecen ninguna profundidad. Los videojuegos que realmente están enfocados al combate y ofrecen la capacidad de explorar, lo hacen en general de una forma pobremente implementada o que resulta poco natural para la experiencia.

Es por esto por lo que la propuesta presentada en este documento serviría para cubrir esa área, que ha sido considerada como no cubierta en el mercado actualmente. El objetivo del videojuego es recoger los mejores elementos de los videojuegos de peleas, e "injertarlos" dentro de la infalible fórmula de los 'roguelikes', juegos capaces de generar niveles y contenido de forma infinita, ¿acaso no suena a la receta del videojuego perfecto?

A continuación, se describe por orden cronológico cómo transcurrirá el desarrollo de una partida: en primer lugar, el personaje aparece en una habitación que se cierra y se llena de enemigos, cuando el jugador haya eliminado a todos los objetivos podrá avanzar a la siguiente sala, hasta que finalmente encuentre la salida y pueda escapar, terminando así la partida.

3.4.2. Definición del diseño de acuerdo con los objetivos

3.4.2.1. Objetivos Materiales

-*Elaborar un producto que sea completamente funcional.* La 'demo' debe permitir al jugador empezar una nueva partida; recorrer todas las habitaciones de la nave, mientras va eliminando los objetivos; y finalmente encontrar la salida para completar el nivel.

-*Construir un modelo de juego fácilmente expansible.* Como ya se ha comentado previamente, la idea es que el videojuego siga desarrollándose hasta que pueda ser comercializado. Por eso es fundamental que la arquitectura del juego esté diseñada para crear variaciones del producto actual con relativa facilidad; lo que se traducirá en re-jugabilidad y aumentará su valor. A nivel de empresa esto dispara considerablemente la viabilidad del producto.

-*Diseñar e implementar un sistema generador de niveles.* Para que una partida pueda tener lugar, es necesario que, de manera automática e independiente, el videojuego sea capaz de generar un nuevo mapa, a partir de una serie de piezas y siguiendo unas normas. Este apartado tiene como objetivo establecer ese

conjunto de normas, y proporcionarle al juego las piezas que utilizará para generar el mapa.

-*Permitir al jugador moverse por el entorno de una forma satisfactoria*. El principal aspecto de la jugabilidad viene del personaje y las acciones que realiza dentro de una partida. Para ello crearemos un sistema, que se encargue de recoger las "órdenes del jugador", y las traduzca al juego, en forma de acciones, compuestas por animaciones y eventos, como "saltar" o "caminar".

3.4.2.2. Objetivos Conceptuales

-*Definir un estilo atractivo y eficaz.* A pesar de que gran parte de la experiencia de un videojuego sólo se puede obtener jugándolo directamente, otra gran parte de lo que transmite se produce de manera visual, a través de su estilo. El estilo define qué información recibe una persona cuando mira la pantalla, hablamos de aspectos como: la paleta de colores, una jerarquía visual que permita identificar con facilidad al protagonista, qué elementos forman parte de la escenografía y cuáles son enemigos. En resumen, ayuda a situar al jugador en el contexto y el mundo en el que viven los personajes.

-*Ofrecer una experiencia adaptable.* Una de las principales características, que pretende separar a este juego frente a sus competidores, es la posibilidad de que el jugador pueda gestionar sus recursos para enfrentarse a cada situación de la manera que él decida, así los jugadores podrán adaptar cada partida a su estilo de juego.

-*Estudiar e imitar elementos de otros videojuegos.* La primera etapa de todo diseño comienza con una búsqueda de referentes. Se buscarán juegos con características que queramos incorporar a nuestro producto. Para ello debemos estudiar qué elementos del juego producen este efecto y cómo lo hacen para diseñar un elemento en nuestro videojuego que cumpla con la misma función.

-*Aplicar las técnicas y metodologías aprendidas durante la formación.* La demostración y correcta aplicación de los conocimientos adquiridos permitirán que el proyecto se desarrolle de la manera más ágil y efectiva posible. Sólo manifestando dichos conocimientos, podremos resolver satisfactoriamente todos los problemas de diseño que se planteen en las diferentes fases de la producción.

3.4.3. Definición de las mecánicas del videojuego

Partiendo de todo el estudio realizado, siguiendo con los objetivos establecidos (3.4.2. Definición del diseño de acuerdo con los objetivos), teniendo en cuenta todas las decisiones de diseño tomadas gracias a la encuesta (3.3.1. Decisiones de diseño tomadas a partir de la encuesta). o el extenso proceso de referenciación del que obtuvimos varios elementos claves (**Estudio en profundidad de referentes históricos y visuales**), se definieron finalmente

todas las partes del videojuego y sus funciones, separadas siguiendo con las dos líneas paralelas de trabajo fijadas en el apartado *1.3.2.1* Diseño del videojuego.

3.4.3.1. Generador de Niveles

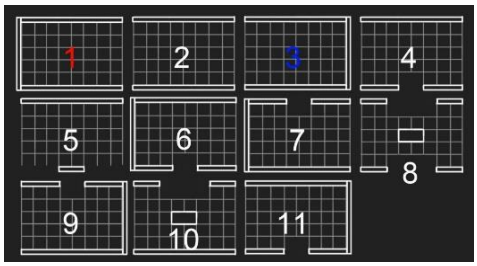


Fig. 8. Primeros bocetos de las habitaciones.

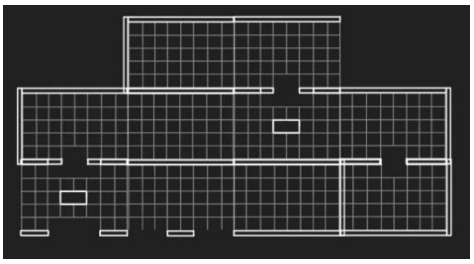


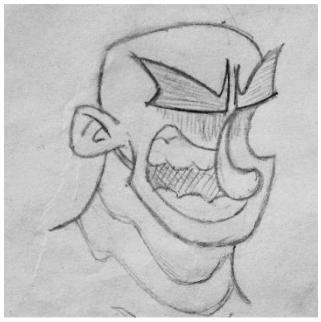
Fig. 9. Prueba de encajado.

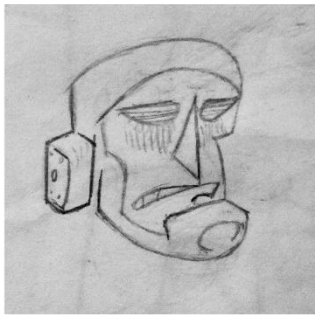
El generador de niveles representa el conjunto de elementos que cumplen con todas las funciones de generación del mapa, estética, reglas de la partida, además de que proporciona y limita el espacio por el que se mueve el personaje. Los tres componentes en los que se ha subdividido este sistema generador son los siguientes:

-El 'Grid'[22]. - Es una cuadrícula de 6x6, capaz de almacenar en cada casilla 4 valores independientes. Haciendo uso de un código que permita modificar o leer los valores de estas casillas, podremos generar "los planos" que utilizará el sistema para crear el nivel. La serie de normas y explicación con pleno detalle de todos los pasos que se siguen para rellenar la información de estas casillas se encuentra en el anexo (**Estudio extendido del sistema generador de niveles**), debido a la acotación de este trabajo.

-Las normas de juego. - El sistema generador, que ya se encarga de organizar y colocar todo el mapa, una vez haya comenzado la partida será el administrador de ésta. Por lo que es el responsable de empezar la partida y cuando el jugador haya encontrado la salida, debe ser el que la concluya de forma independiente.

-Las habitaciones. – Desde el momento que se termina de escribir en el 'Grid', el sistema se encarga de generar un código para cada casilla con esos cuatro números. Ese código generado para cada casilla le dice a Unity qué habitación debe colocar en cada lugar, a través de un código implementado en el sistema generador.





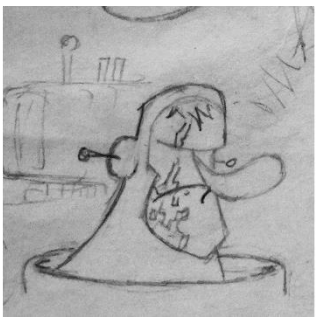


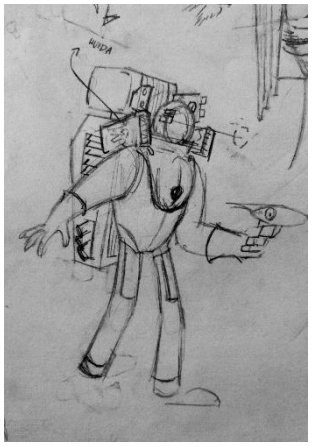
Fig. 10. 11. Y 12. Bocetos del protagonista realizados siguiendo el estilo visual de los referentes.

3.4.3.2. Personaje

Partiendo de los referentes visuales escogidos para el diseño del personaje, se realizan los primeros bocetos (véase fig. 10. 11. Y 12.) y dibujos en una fase de exploración que busca la definición de esa línea estética de la que hablábamos en los Objetivos Conceptuales (3.4.2.2.). A partir de que estos

[22] 'Grid' es la palabra que se utiliza en inglés para denominar a una cuadricula, aunque normalmente este término se usa en programación para referirnos a una cuadrícula que contiene una serie de celdas que a su vez contienen información en su interior.

dibujos hayan alcanzado un grado de similitud al original que se considere satisfactorio, pero añadiendo su propio toque personal, se considerará que el estilo ha sido "dominado" y por tanto, se dará comienzo a la producción.



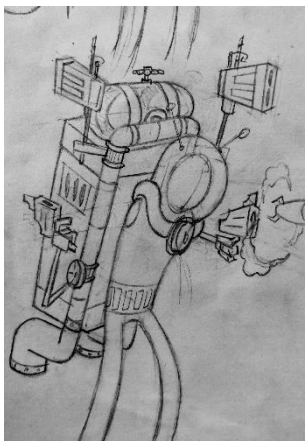


Fig. 13. 14. Y 15. Exploración del personaje y distintas versiones del traje de astronauta.

Pero antes de comenzar a trabajar con una versión más precisa de nuestro protagonista (véase fig. 13. 14. 15.), es necesario recordar la finalidad de dicho diseño, que resulta ser la animación del personaje. Por lo tanto, debemos optimizar su diseño al máximo para facilitar las labores de producción. Teniendo claros estos conceptos se establece la lista de acciones que puede realizar el personaje y por tanto deberemos animar:

-Posición Estática
-Caminar/Correr
-Saltar
-Combo de ataques
-Disparar

Después de haber decidido qué puede hacer el personaje y qué no, nos centramos en la funcionalidad que tendremos que implementar en el personaje dentro de Unity, para que éste sea capaz de reaccionar según las órdenes del jugador, siguiendo una lógica que se tendrá que definir. Para cumplir con esta función se han diseñado tres elementos con funciones conjuntas:

-Los parámetros o variables. – Los parámetros son valores independientes que pueden almacenar distintos tipos de información.

Podemos crear una variable cuyo valor sea negativo cuando saltemos y vuelva a ser positivo solamente al tocar el suelo, de este modo la variable representa si estamos tocando el suelo o no.

-El Animator Controller. – Haciendo uso de variables como la velocidad horizontal y vertical de un objeto, si está apoyado en el suelo o si se ha pulsado una tecla en concreto, se puede crear un grafo que contenga la lógica encargada de reproducir en orden las distintas animaciones.

Por ejemplo, cuando el jugador pulse el botón de atacar, el personaje debe mostrar cierta animación, pero cuando ésta acabe el personaje no debe quedarse estático al final del ataque, sino que deberá volver automáticamente a mostrar la animación de pose estática.

-El código del personaje. – El personaje será el encargado de transformar la información que introduzca el jugador al sistema y debe completar la acción.

Para completar cada una de las acciones disponibles, es necesario definir todas las reacciones (moverse, atacar, saltar) y asociarles un parámetro a cada una. Cuando el jugador pulse una tecla se activará dicho parámetro y el Animator Controller reproducirá la animación indicada.

3.5. DESARROLLO DEL PRODUCTO

3.5.1. Escenario

3.5.1.1. El generador de niveles

El sistema debe seguir una serie de reglas para evitar errores durante la generación del mapa, como habitaciones que no estén conectadas al resto o el caso de una habitación que tenga una puerta y al otro lado no haya ninguna sala construida.

De este modo se estableció la siguiente función, llamada CreateMap(), encargada de escribir dentro de las casillas del 'Grid':

```
public void CreateMap()1.
{
    while (OCUPADO <= 20)2.
    {
        int x, y;                3.
        Vec2 newRoom;
        newRoom = selectRoom();
        if (newRoom.x != 0       4.
            && newRoom.y != 0
            && newRoom != null)
        {
            x = newRoom.x;       5.
            y = newRoom.y;
            Interconnect(x, y);
        }
    }
}
```

Fig. 16. Función encargada de la generación de las salas.

La función Createmap (1) se encarga de generar aleatoriamente las salas del mapa. En este caso, se ha limitado el número de habitaciones a 20 (2).

En primer lugar, la función SelectRoom se encarga de seleccionar una casilla del Grid, devolviendo sus coordenadas (3). A continuación, se comprueba si en la casilla seleccionada se puede construir una sala (4). Por último, se llama

a la función Interconnect (5), que es la encargada de conectar la nueva sala con sus vecinas.

Con el objetivo de evitar que al concluir la generación haya habitaciones o partes del mapa que no se encuentren conectadas al resto, se ha decidido que sólo se podrán construir nuevas salas en las casillas que estén directamente pegadas a las habitaciones ya construidas, a través de la función SelectRoom (véase fig. 17.).

```
private Vec2 selectRoom()
{
    int x, y;
    Vec2 res;

    if (OCUPADO == 0) 1.
    {
        res = RandomSelect();
        x = res.x;
        y = res.y;                    2.
    3.  rooms[x, y].setStartRoom();
        startRoomIndex = new Vec2(x, y);
        return res;
    }
    else 4.
    {
        Vec2 newVec2 = getUsedRoom();
        x = newVec2.x;
        y = newVec2.y;                5.

            if ((rooms[x + 1, y].isEmpty() != true && x + 1 != rooms.GetLength(0)) &&
            (rooms[x - 1, y].isEmpty() != true && x - 1 != 0) &&
            (rooms[x, y + 1].isEmpty() != true && y + 1 != rooms.GetLength(1)) &&
            (rooms[x, y - 1].isEmpty() != true && y - 1 != 0))                        6.
            {
                blockRoom(x, y);
                return new Vec2(0, 0); 7.
            }
            else 8.
            {
                res = getSalida(newVec2);
                x = res.x;
                y = res.y;
                return res;           9.
            }
    }
}
```

Fig. 17. La función SelectRoom se utiliza dentro de la función CreateMap (véase fig. 16.) e indica en qué casilla se construirá la siguiente sala para que todas estén conectadas.

Al escoger una nueva casilla para su construcción, la función comprueba cuántas habitaciones hay ya colocadas en el Grid. Si el número de habitaciones es igual a cero (1), significa que no hay ninguna sala y ésta será la primera, de manera que esa primera casilla es escogida de forma aleatoria a través de la función RandomSelect (2) y se marca como la “Primera Sala Construida” gracias al método setStartRoom (3).

Si por el contrario la función detecta que hay habitaciones ya construidas (4), llama a la función getUsedRoom (5) que se encarga de escoger aleatoriamente una de las habitaciones ya colocadas. En caso de que todas las casillas alrededor de la habitación seleccionada se encuentren ocupadas o fuera de los límites del Grid (6) no podremos construir a partir de dicha sala, por lo que la función se encargará de bloquear esa casilla (7) para que no vuelva a ser escogida por el método getUsedRoom. En el caso de que alguna de las casillas colindantes con la habitación escogida esté disponible (8), se escogerá una de

estas casillas a través del método getSalida (9), y dicha casilla será el resultado de la función SelectRoom (fig. 17.), que a su vez devolverá la posición de ésta a la función CreateMap (fig.16.) para que finalmente construya la habitación.

Como ya se ha comentado, necesitamos que las salas no sólo se coloquen en el lugar correcto, sino que también es importante que las entradas y salidas de cada habitación se configuren en relación con las distintas conexiones que puedan existir a su alrededor. La función Interconnect (fig. 18.) es la encargada de cumplir con esta misión de "alinear" todas las distintas conexiones entre salas.

```
private void Interconnect(int x, int y)
{
    useRoom(x, y); 1.

    if (x + 1 <= rooms.GetLength(0)-1) 2.
    {
        if (rooms[x + 1, y].isEmpty() != true) 3.
        {
4.          horizontalConnect(x, y, HorizontalDirection.GoRIGTH);
        }
    }
    if (x - 1 >= 0)
    {                                                          5.
        if (rooms[x - 1, y].isEmpty() != true)
        {
            horizontalConnect(x, y, HorizontalDirection.GoLEFT);
        }
    }
    if (y + 1 <= rooms.GetLength(1)-1)
    {                                                          6.
        if (rooms[x, y + 1].isEmpty() != true)
        {
            verticalConnect(x, y, VerticalDIrection.GoUP);
        }
    }
    if (y - 1 >= 0)
    {                                                          7.
        if (rooms[x, y - 1].isEmpty() != true)
        {
            verticalConnect(x, y, VerticalDIrection.GoDOWN);
        }
    }
}
```

Fig. 18. Interconnect es llamada desde el último paso de CreateMap (fig. 16.), para después de haber seleccionado una nueva casilla escribir en su interior dependiendo de la información que contengan las casillas de su alrededor.

Cuando se escribe dentro de cualquiera de las casillas del 'Grid' lo primero que se hace es indicar que ésta ha sido utilizada, a través de la función useRoom (1). Gracias a que hemos marcado la nueva casilla como construida, ésta podrá ser escogida por la función SelectRoom (Fig. 17.) al invocar el método getUsedRoom.

Una vez marcada, se comprueba si la casilla de su derecha se encuentra dentro de los límites del 'Grid' (2), y si dicha casilla contiene una habitación (3). En el caso de que estas condiciones se cumplan, la habitación que está siendo construida hará uso de la función HorizontalConnect con la habitación de su derecha (4), escribiendo dentro de ambas casillas en el 'Grid'. De nuevo este proceso de conexión entre salas se encuentra detallado en los documentos anexos (**Estudio extendido del Sistema Generador de Niveles).**

Hecha la comprobación y posible conexión con la casilla derecha, se procede a realizar la misma acción con las tres casillas restantes, haciendo uso tres funciones idénticas a la estudiada (5. 6. Y 7.), cambiando por supuesto la dirección de comprobación y conexión de cada una.

3.5.1.2. Las habitaciones

Las habitaciones se colocarán dentro de un escenario vacío en Unity, siguiendo los resultados obtenidos por el sistema generador de niveles. Los cuatro números almacenados por el 'Grid' se juntan por orden y señalan el número de habitación que se debe colocar en cada casilla. Cada habitación está compuesta por: un conjunto de imágenes renderizadas por capas, creando profundidad; unos colisionadores o paredes, que limitan el espacio de movimiento; y su propio Animator Controller, encargado de gestionar las animaciones de las puertas.

```
public class Room
{
    public enum Direction { UP, LEFT, RIGHT, DOWN };
    private int left, right, up, down;
    private bool in_use;
    private bool blocked;
    private bool startRoom;

    public Room()
    {
        in_use = false;
        blocked = false;
        startRoom = false;
    }

    public void SetValue(Direction d, int val)
    {
        switch (d)
        {
            case Direction.UP:
                up = val;
                break;
            case Direction.DOWN:
                down = val;
                break;
            case Direction.LEFT:
                left = val;
                break;
            case Direction.RIGHT:
                right = val;
                break;
        }
    }
```

Fig. 19. Fragmento del código generado. Definición de las variables que contienen la información de las casillas.

El fragmento de código mostrado en la Figura 19 muestra la clase Room, que representa el estado de una habitación.

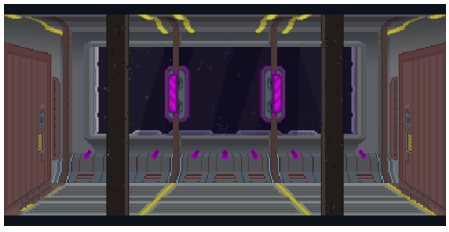
1- Estas variables representan diferentes atributos de la casilla como si ya se ha construido en su interior, o si ha sido bloqueada porque todas las casillas de su alrededor están ocupadas. Variables que son muy útiles para su uso dentro de otras de las funciones estudiadas.

2- Cuando se activa por primera vez el código de la clase Room, sus variables se declaran como falsas ya que todavía no se le ha atribuido ninguna característica.

3- Función para escribir en las direcciones – La función SetValue recibe un valor (val) y una dirección (d), entonces accede a una casilla del 'Grid' y sustituye el valor de la dirección seleccionada por un número especificado. Este código nos permite definir las conexiones de los vecinos de cada habitación.

3.5.1.3. Fondos y escenario

Todas las imágenes fueron realizadas en Photoshop Cs6, utilizando la herramienta lápiz para dibujar píxel por píxel, en una cuadrícula de un tamaño de 384x192 píxeles. Las imágenes generadas fueron posteriormente separadas en tres capas para exportarlas por separado a Unity y crear un efecto de profundidad.



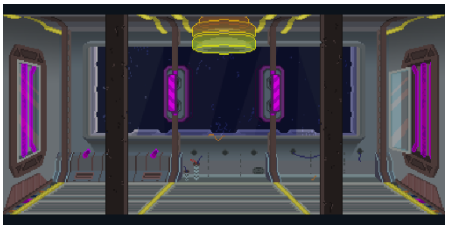
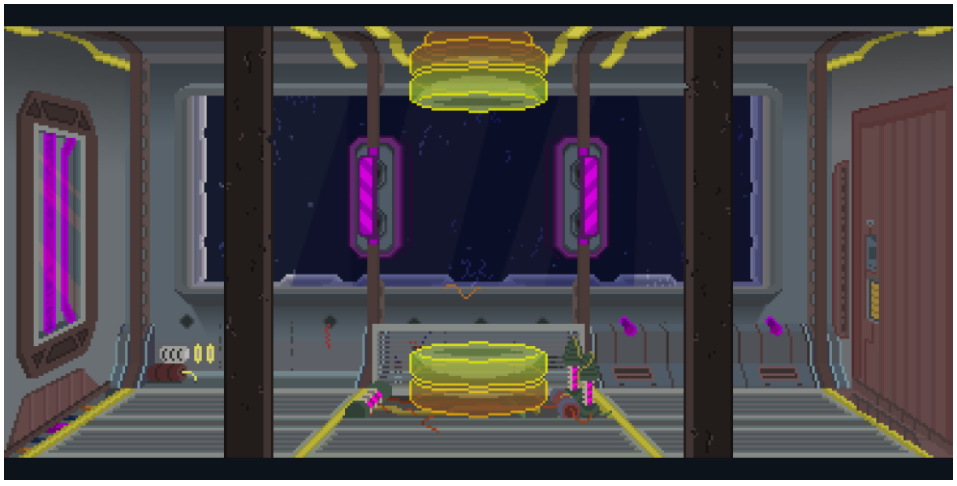


Fig. 20. 21. y 22. Ejemplos de habitaciones con diferentes conexiones.



El escenario está formado principalmente por la habitación y el fondo que a su vez contiene distintas imágenes más pequeñas en su interior dependiendo de las conexiones de la sala. Esto significa que, si la habitación tiene una salida a la derecha, deberá colocar en ese lugar una puerta capaz de abrirse mostrando una animación cuando el jugador cumpla con los requisitos establecidos en su código.

3.5.1.4. Futuros planes de expansión del sistema generador de niveles

La arquitectura de este sistema se ha diseñado con la intención de su expansión futura, de manera que permita generar escenarios con distintas apariencias estéticas utilizando esta misma fórmula.

De hecho, esto es posible con la realización de unos cambios realmente sencillos de implementar, simplemente haría falta el tiempo necesario para dibujar todos los nuevos fondos. El sistema generador podría aplicar esa segunda estética con el simple uso de una nueva variable aleatoria, si su valor es positivo elegirá la nave espacial y si por el contrario es negativo escogerá el segundo tipo de habitación.

3.5.2. Personaje

3.5.2.1. Animación

El modelo del personaje inicialmente se hizo a mano en papel, por un motivo de soltura, partiendo de formas muy básicas como un triángulo para para las extremidades. En esta primera fase se buscaba la mejor forma de representar cada acción, cuidando aspectos como la silueta del dibujo, que debe ser lo más limpia y reconocible posible, además de intentar marcar todo lo posible la línea de acción [23]del movimiento en cuestión.

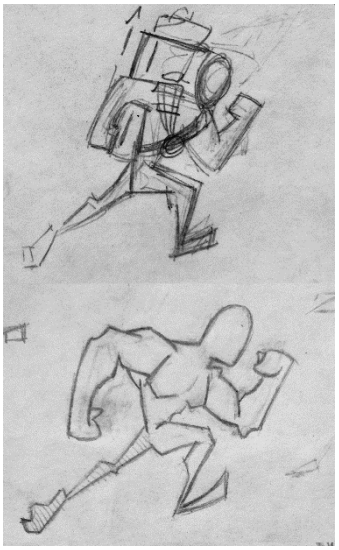


Fig. 23. Estudio de la posición que mejor representase la acción de correr.
Fig. 24. Limpiado del dibujo y definición de la anatomía.

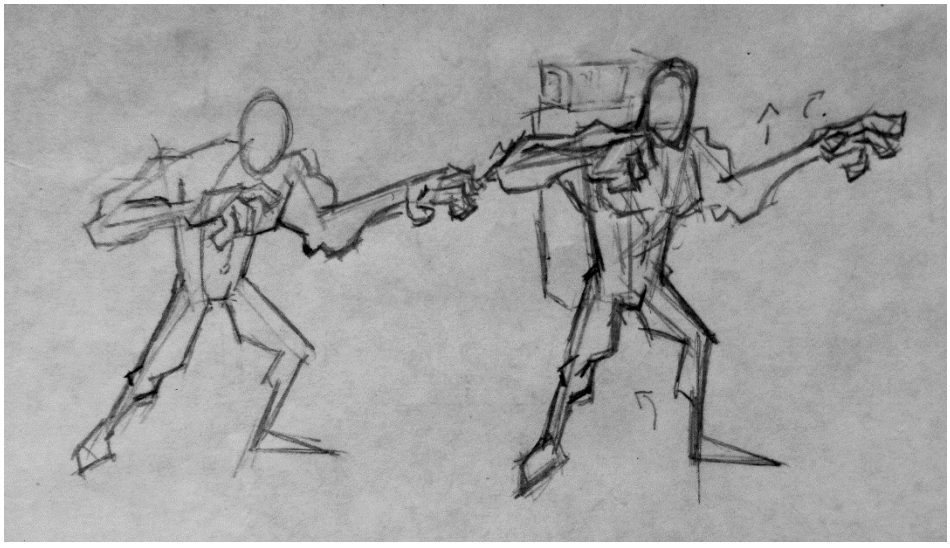


Fig. 25. Prueba de la animación del personaje cuando está quieto.

Con estos primeros bocetos como base (véase fig. 23.), se añadieron digitalmente los volúmenes de los diferentes músculos para dar fuerza al personaje (fig. 24 y 25). Cuando la animación ya mostraba claramente la acción representada, se procedió a limpiar la línea de todos los frames y a definir con más precisión los músculos.

[23] La línea de acción es una línea imaginaria que denota la fuerza y actitud del movimiento, representa lo más esencial de la acción.

Finalmente se dibujaron la ropa y demás detalles por encima del personaje, y por debajo de la línea se creó una capa donde iría el color.

Es interesante recalcar la importancia de los conocimientos adquiridos previamente para la elaboración de esta parte del proyecto en concreto, ya que la capacidad de aplicar correctamente los principios de la animación[24] tiene un efecto muy notable en el resultado. Pero si debemos de hablar de un principio que ha marcado y tenido una mayor influencia en el movimiento, ha sido el uso de la anticipación, imprescindible en un videojuego de peleas.

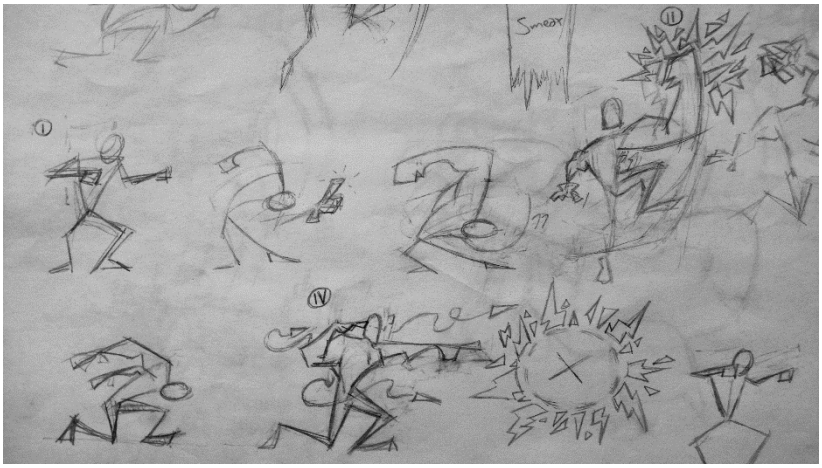


Fig. 26. Primeros bocetos del combo del personaje y estudio del efecto de la anticipación.

La anticipación representa de manera visual la preparación para realizar una acción, cuanto mayor sea la anticipación más fuerza tendrá la acción. Esta anticipación se puede ver por ejemplo cuando un personaje va a dar un gran salto, antes de estirarse y dejar el suelo tiene que contraerse al agacharse y doblar las piernas para coger impulso. De la misma manera utilizamos la anticipación para hacer que los golpes del personaje tengan un mayor impacto (véase fig. 26.), o al menos lo tendrán en el subconsciente de los jugadores.

Después de haber realizado todas las animaciones del personaje, se juntaron todos los frames en varias hojas divididas en 12 rectángulos iguales. Este tipo de hoja se llaman "Spritesheet" y son utilizadas por los diseñadores para ahorrar espacio de almacenamiento en los videojuegos, ya que de una sola imagen podemos obtener una o varias animaciones completas. A continuación, se muestra de ejemplo de una de las 'Spritesheet' generadas (véase fig. 27.) para su uso dentro de Unity.

[24] Los 12 principios de la animación fueron presentados en la obra**: "Disney Animation: The Illusion of Life"** (1981). Thomas, Frank; Ollie Johnston.

Fig. 27. Ejemplo de una de las hojas de Sprites generados. Los seis primeros frames corresponden a la animación estática, y los cinco últimos forman el primer ataque del combo del personaje.

3.5.2.2. Programación

Obtenidas las hojas de sprites que incluyen las animaciones del personaje, fueron introducidas en Unity para construir su Animator Controller. Cada animación se separa por frames dentro del editor de Unity[25] y se guarda como un elemento independiente dentro del Animator, en la siguiente imagen podemos observar cómo se organizaron las transiciones entre todos los diferentes estados [26]del protagonista, incluyendo múltiples estados que no se contemplan dentro de este trabajo:

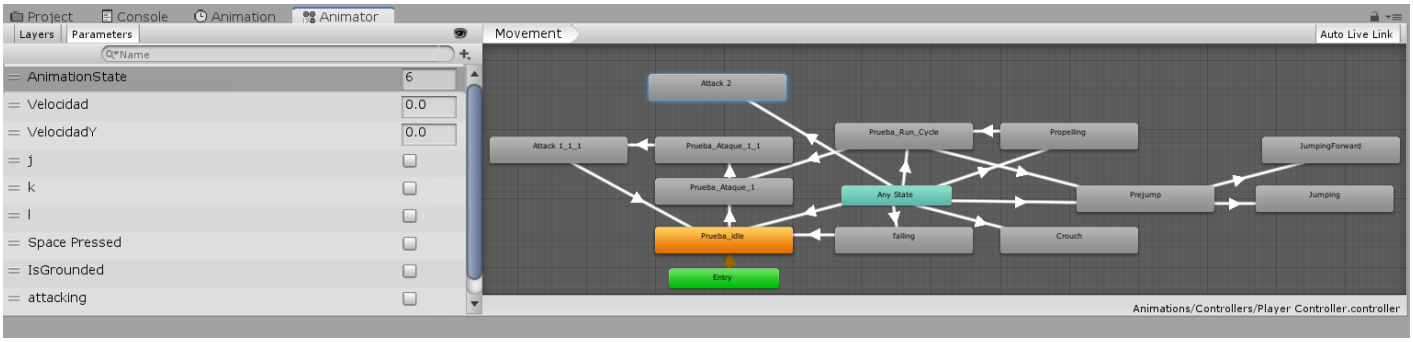


Fig. 28. Animator del personaje y lista de los parámetros.

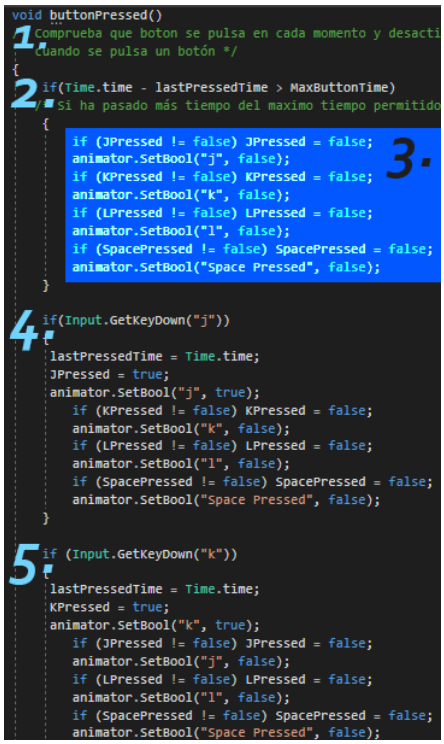


Fig. 29. Fragmento del código encargado de recibir las órdenes del usuario y traducirlas al juego a través de los parámetros del Animator.

También podremos ver a la izquierda de la misma imagen, todos los parámetros de los que hace uso el Animator. Estas variables están directamente conectadas a algunas teclas designadas a través de una función en el código del personaje, función que recibe y traduce la información que introduce el jugador, mediante el código mostrado en la Figura 26.

1- Definición del método- buttonPressed es el nombre de la función.
2- Condición – Si ha pasado más de 0,5 segundos (MaxButtonTime) desde la última vez que se ha pulsado un botón (LastPressedTime), se considera que ningún botón ha sido presionado y en consecuencia lo retransmite a través de los parámetros del Animator con la función SetBool, al indicar que el valor de todos los botones es falso (3).
4- Condición - Si por el contrario la función detecta que el jugador ha puslado una tecla, en este caso la letra “j”, entonces le dirá al Animator que establezca los valores de todas las demás teclas menos la presionada a falso, ya que no puede haber dos teclas pulsadas a la vez.
5 en adelante - Se repite exactamente el mismo proceso, pero con la “k”, la “l” y la tecla de “espacio”, cada vez que una de éstas es pulsada la función se encarga de negar el resto.

[25] El editor de Unity es la interfaz donde los diseñadores trabajan y pueden realizar los cambios en un proyecto.
[26] Cada estado representa una acción del protagonista, uno seria caminar, otro sería atacar, etc.

3.5.2.3. Futura expansión del modelo presentado

A partir del modelo o personaje fabricado, el cual podría ser considerado como una plantilla, podemos crear todas las variantes deseadas en forma de otros personajes, con el simple hecho de copiar el modelo original y sustituir sus animaciones por otras nuevas.

Claro que esto deberá ir acompañado de una modificación del código original, para que cada uno de los personajes tenga ataques y habilidades únicas ofreciendo así una experiencia diferente, pero esta aclaración sirve para poner de manifiesto la versatilidad del producto presentado.

3.6. REFINADO

3.6.1. Testeo

La revisión de todo el material generado fue una constante en el proceso de trabajo seguido. Desde un punto de vista estético, los primeros bocetos del personaje y los distintos escenarios deben buscar una homogeneidad para obtener un producto cohesionado.

Pero desde el punto de vista de la programación, la labor de testeo cumple con una función imprescindible, ya que el código generado suele presentar errores inesperados que requieren de un estudio exhaustivo de cada caso. A través de estas comprobaciones continuas, se han descubierto una serie de errores menores o carencias en la estructura del proyecto presentado, debilidades de las cuales podemos aprender y que se han anotado para su corrección en un futuro próximo:

Saturación del Animator Controller. - A pesar de ser una herramienta realmente útil para realizar cambios entre las animaciones, siguiendo una lógica muy sencilla, el espacio del Animator se llena de estados enseguida y dificulta en gran medida el trabajo con la herramienta. Por lo que en el futuro se debe implementar un cambio de animaciones desde el código del personaje.

Falta de contenido para rellenar las habitaciones. – Otro de los puntos a corregir por el videojuego es la falta de objetos interesantes o que permitan interactuar al personaje. La versión actual, si bien es verdad que cumple con todas las funciones establecidas por el diseño, no incluye ningún incentivo para que el jugador siga explorando el interior de la nave después de haber encontrado la salida, por lo que se buscarán elementos que puedan atraer el interés del jugador, como puntos extra que se encuentren escondidos por la nave o bonificaciones de algún tipo.

3.7. RESULTADO

3.7.1. **Demostración del 'gameplay'**.

Enlace:
https://drive.google.com/drive/folders/1ZKvCFNnT4lfUNMeOCsc2fC0YEbswMYoL?usp=sharing

3.7.2. **Demostración del funcionamiento del sistema generador de niveles**.

Debido a la naturaleza del videojuego, el funcionamiento del sistema generador implementado no se puede observar por completo dentro de la 'demo', ya que el tamaño de una habitación ocupa toda la pantalla y nos impediría ver el mapa en su conjunto, por lo que se ha realizado un vídeo de demostración:

https://youtu.be/I9JTVvPgfwU

En el vídeo se muestran varios ejemplos de la generación del mapa. Cada habitación está representada por un código de 4 números que indica las conexiones de la sala, el proceso para obtener el código de cada habitación se encuentra en el documento anexo (**Estudio extendido del Sistema Generador de Niveles**). Además de que la sala de inicio está representada por un cuadrado rojo, y determina el punto desde el que el personaje aparece cuando comience la partida.

3.8. PREVISIÓN DE IMPACTO

Es importante recordar que nos encontramos ante un "adelanto jugable" y, por lo tanto, el producto final estaría sujeto a sufrir cambios en cualquiera de sus áreas. El videojuego presentado no ha sido diseñado para ser comercializado por sí mismo, pero muestra el potencial de la propuesta desarrollada.

Teniendo como referencia el tiempo empleado en la elaboración de la 'demo', se ha determinado que haría falta un tiempo de desarrollo aproximado de otras 28 semanas para poder lanzar un producto terminado. Durante este segundo periodo de tiempo, se terminaría el personaje actual para crear otros 4 personajes al menos, y se añadirá una temática diferente al mapa por cada uno de estos nuevos personajes.

Estableciendo de este modo una cifra de 12 euros por hora completa de trabajo, a un ritmo de 20 horas semanales durante un periodo de 28 semanas, se necesita hacer una inversión unos 7.000 euros para compensar el trabajo invertido en el desarrollo.

3.9. DIFUSIÓN

Con el objetivo de publicitar el producto presentado y atraer así a la mayor cantidad de clientes, se ha establecido una línea de actuación que se

centra en todos los aspectos que tienen que ver con las redes sociales y la publicidad del videojuego.

Antes de comenzar a mostrar nuestra idea a los jugadores, es necesario que tengamos claro cuáles son las características únicas del producto que pueden servir como reclamo, para hacer énfasis en ellas al publicitar el videojuego. Si el público objetivo está buscando por ejemplo un alto nivel de violencia, tendremos que buscar la manera de transmitirlo a través de las promociones o anuncios.

De este modo nos encontramos con dos opciones realmente accesibles para publicitar nuestro producto: por un lado nos encontramos con las diferentes redes sociales, para las que se podría elaborar un pequeño tráiler mostrando las características principales del videojuego; pero el proyecto además podría presentarse a un concurso de desarrolladores independientes, al igual que hicieron otros compañeros de la titulación con sus trabajos finales de grado.

3.10. PRESUPUESTO

Contando el periodo de tiempo comprendido entre los días 2 de septiembre y 9 de diciembre, nos encontramos ante una fase de desarrollo de 14 semanas.

Durante estas 14 semanas, se han empleado una media de 4 horas diarias 5 días a la semana, lo que se traduce en un tiempo de 280 horas de trabajo. Si el precio estándar para cada hora de trabajo es de 12 euros, daría un total de 3.360 euros en bruto por la mano de obra necesaria para fabricar esta 'demo'. A este precio se deberá añadir el dinero que habrá que pagar mensualmente por hacer uso de las licencias oficiales de los programas empleados.

4. CONCLUSIONES

4.1. VALORACIÓN DE LOS RESULTADOS

Para poder evaluar de un modo objetivo el producto realizado, tenemos que valorar si los resultados son coherentes o han conseguido materializar todos los objetivos propuestos en el apartado 3.4.2. Definición del diseño de acuerdo con los objetivos. Sin duda la 'demo' refleja cómo los dos apartados más trabajados han sido la jugabilidad y la estética.

El videojuego presentado es un producto completamente funcional y capaz de generar el mapa de forma autónoma al comenzar la partida, por lo que la elaboración de un prototipo jugable e implementación del sistema generador

de niveles ha sido todo un éxito. El personaje también es capaz de moverse libremente por el interior de cada habitación y ejecutar un combo de ataques.

Este modelo de videojuego es una "base expansible" ideal para continuar con su desarrollo. Como se ha mencionado a lo largo de todo el trabajo (especialmente en los apartados 3.5.1.4 y 3.5.2.3. dedicados a plantear el posible desarrollo en los meses posteriores a la exposición del trabajo), la 'demo' servirá como una plantilla a partir de la cual podremos seguir añadiendo contenido, resultando en variantes que añaden re-jugabilidad y valor al producto, gracias a modificaciones que conllevan un tiempo relativamente corto de implementación.

La experiencia que ofrece Infinite Andrómeda es el resultado de estudiar y recoger características de múltiples títulos, buscando la combinación perfecta que se traduzca en un 'gameplay' adictivo, que esté al nivel de otros juegos similares con los que compite. El estilo definido para la realización de personajes y fondos, se considera que ha sido capaz de resolver satisfactoriamente las necesidades planteadas por el diseño, y no sólo eso, sino que añade un toque de personalidad del que se beneficia el videojuego.

Las animaciones representan de una forma clara y eficaz las acciones del personaje, aunque el resultado final no haya sido pasado a limpio para mostrar la apariencia que tendría el protagonista en la versión terminada del videojuego.

De este modo, es verdad que no todos los objetivos se han podido ver reflejados en el trabajo final de la misma forma, este sería el caso de *Ofrecer una experiencia adaptable*. Si bien el producto ofrece al jugador una serie de diferentes acciones para realizar durante la partida, la gestión de recursos ha sido un apartado que ha tenido que ser parcialmente pospuesto para su futura implementación, ya que otras áreas del proyecto resultan mucho más vitales para su correcto funcionamiento. De este modo, la adaptabilidad del producto será uno de los aspectos pendientes a desarrollar para el lanzamiento del videojuego.

Finalmente, se destaca la importancia de la aplicación de todos los conocimientos obtenidos durante el proceso de formación, como la capacidad de establecer una metodología que marque todo el proceso de trabajo, la posibilidad de realizar un juicio crítico y analizar las tendencias actuales en el diseño y las tecnologías creativas, o la formación en el uso de distintas herramientas que nos permita escoger la mejor técnica para resolver cada parte del proceso.

4.2 VALORACIÓN DE LA EFECTIVIDAD DEL PROCESO DE TRABAJO

Se considera que la dinámica de trabajo establecida ha permitido desempeñar todas las diferentes tareas involucradas en el proceso de diseño y desarrollo de la forma más efectiva y organizada posible, evitando problemas de coordinación entre las distintas áreas del proyecto. Esta función se considera esencial en la elaboración del trabajo, ya que una mala planificación podría haber puesto en peligro los plazos establecidos y por tanto podría perjudicar a todo el proyecto en conjunto.

4.3 APORTACIONES PROFESIONALES

La realización de un prototipo tan ambicioso ha sido sin duda una tarea difícil, que al estar desarrollado por una sola persona ha requerido de una gran capacidad para planificar y ejercer distintas funciones. Esta gestión de todas las distintas partes que componen el proyecto ha tenido un efecto beneficioso desde un punto de vista profesional ya que, al desempeñar todas las labores de diseñador, programador y animador de la producción, se ha adquirido un conocimiento de primera mano del funcionamiento y los límites de todas y cada una de las partes que son necesarias en la producción un buen videojuego.

En conclusión, Infinite Andrómeda ha sido la culminación de un proceso que comenzó hace muchos años, impulsado por videojuegos que marcaron a toda una generación de niños, que soñaban con los mundos imposibles de Mario[27] o las locas aventuras de Crash Bandicoot[28]. Y hoy más que nunca, ese sueño de poder crear un nuevo mundo producto de tu imaginación, está al alcance de todo el mundo, gracias a las tecnologías de libre acceso como Unity o Game Maker Studio[29].

[27] *Mario* es una franquicia de videojuegos distribuida y desarrollada por *Nintendo.*
[28] Crash Bandicoot es el nombre de una saga de videojuegos que ha ido cambiando tanto de distribuidora como de desarrolladora, y sigue las aventuras de su protagonista Crash.
[29] Game Maker Estudio es otro de los mejores motores de videojuegos gratuitos.

BIBLIOGRAFÍA

WEBGRAFÍA

Página web oficial de Unity**:**
https://unity.com/es

Página web de todos los productos de Adobe:
https://www.adobe.com/es/products/photoshop.html

Página principal del Adobe Animate:
https://www.adobe.com/es/products/animate.html

"La interpretación de Berlin":
http://www.roguebasin.com/index.php?title=Berlin_Interpretation

Sitio web desde el que se puede acceder a las distintas "Convenciones internacionales de desarrolladores de Roguelikes":
http://www.roguebasin.com/index.php?title=IRDC

¿Qué significa la clasificación PEGI?:
https://pegi.info/es/node/19

Página oficial de Street Fighter:
https://streetfighter.com/street-fighter-30th-anniversary-collection/

Página oficial de Tekken:
https://es.bandainamcoent.eu/tekken

Página oficial de Smash Bros España:
https://www.smashbros.com/es_ES/

LITERATURA

"Disney Animation: The Illusion of Life" (1981). Thomas, Frank; Ollie Johnston.

LUDOGRAFÍA

Crawl. – Videojuego
Desarrolladora: Powerhoof.
Distribuidora: Powerhoof.
Fecha de publicación: 2014

Jetpack Joyride. – App
Desarrolladora: Halfbrick Games.
Distribuidora: Halfbrick Games.
Fecha de laznzamiento: 2011

Enter the Gungeon. – VIdeojuego
Desarrolladora: Dodge Roll.
Distribuidora: Devolver Digital.
Fecha de inicio de la saga: 2016

Street FIghter. – Saga de videojuegos
Desarrolladora: Capcom.
Distribuidora: Capcom.
Fecha de inicio de la saga: 1987

Tekken. – Saga de videojuegos
Desarrolladora: Namco.
Distribuidora: Namco Bandai Games.
Fecha de inicio de la saga: 1994.

Smash Bros. – Saga de videojuegos
Desarrolladora: Bandai Namco Studios Sora Ltd.
Distribuidora: Nintendo.
Fecha de inicio de la saga: 1999.

Mario. – Saga de videojuegos
Desarrolladora: Múltiples.
Distribuidora: Nintendo.
Fecha de inicio de la saga: 1981.

Crash Bandicoot. -Saga de videojuegos
Desarrolladora: Múltiples.
Distribuidora: Múltiples.
Fecha de inicio de la saga: 1996.

INDICE DE FIGURAS

ANEXOS

1º Anexo. - **Historia, Origen y Evolución de los Roguelike.**
https://drive.google.com/file/d/1ds7cNbTaxPrzFghUlydK4QOUT58uwn3I/view

2º Anexo. – **Estudio en profundidad de los referentes históricos y visuales.**
https://drive.google.com/file/d/1L6HQ_8JY4SMELKaCrL3VJq5EuGPkAYx7/view?usp=sharing

3º Anexo. – **Estudio del público objetivo a través de la encuesta.**
https://drive.google.com/file/d/1LocwZcMDGe1oTUhtlNyMVz_RqZ24Te77/view?usp=sharing

4º Anexo. – **Funcionamiento extendido del sistema generador de niveles.**
https://drive.google.com/file/d/1QLsUuRn2pAYhXudhGn-Ub7FgXsnvhle3/view?usp=sharing