

Universidad San Jorge

Escuela de Arquitectura y Tecnología

**Grado en Diseño y Desarrollo de
Videojuegos**

Proyecto Final

GAME FEEL – Estudio de caso

Autor del proyecto: Álvaro Tomás Lozano

Director del proyecto: Jorge Echeverria Ochoa

Zaragoza, 07 de septiembre de 2022



Este trabajo constituye parte de mi candidatura para la obtención del título de Graduado en Diseño y Desarrollo de Videojuegos por la Universidad San Jorge y no ha sido entregado previamente (o simultáneamente) para la obtención de cualquier otro título.

Este documento es el resultado de mi propio trabajo, excepto donde de otra manera esté indicado y referido.

Doy mi consentimiento para que se archive este trabajo en la biblioteca universitaria de Universidad San Jorge, donde se puede facilitar su consulta.

Firma

Fecha

07/09/2022

A handwritten signature in black ink that reads "ALVARO". The letters are stylized and connected. Below the signature, there are several horizontal lines drawn with a pen, suggesting a signature line or a decorative underline.

Dedicatoria y Agradecimiento

Primero, quiero agradecerles a mi padre y a mi madre la oportunidad que me han brindado para estudiar lo que he querido durante 5 años y apoyarme en todo momento de mi vida. Gracias por vuestro esfuerzo y cariño que me habéis dado estos años. También, a mis familiares cercanos que han estado ahí y se han portado como una verdadera familia en tiempos difíciles.

Agradecer a todos mis amigos y compañeros que he hecho durante el transcurso de la carrera, que me han aportado tantos momentos buenos y me han apoyado en los malos. En especial a Javier Jiménez por ser un buen amigo y darme su ayuda cuando lo necesitaba.

Agradecer a los profesores de la Universidad San Jorge que he tenido durante el transcurso de mi carrera por el conocimiento y la ayuda que me han aportado, que siempre me han apoyado cuando lo que he necesitado y han tenido paciencia conmigo hasta en los momentos más duros.

Gracias a todos.

Contenido

Resumen	1
Palabras Clave / Keywords	1
Abstract.....	1
1. Introducción	3
2. Estado del Arte	6
2.1. Game Feel.....	6
2.1.1. <i>Fundamentos del Game Feel.....</i>	<i>6</i>
2.1.2. <i>Definición de las experiencias de la sensación de juego.</i>	<i>8</i>
2.2. Creación del Game Feel.....	10
2.2.1. <i>Game Feel como sensación estética del control</i>	<i>10</i>
2.2.2. <i>Game Feel como la experiencia de aprender, practicar y dominar una habilidad.....</i>	<i>12</i>
2.2.3. <i>Control en Tiempo Real</i>	<i>15</i>
2.3. Métricas del Game Feel	18
2.3.1. <i>Resumen de las Métricas</i>	<i>19</i>
2.3.2. <i>Métrica de Respuesta</i>	<i>19</i>
2.4. Otras referencias sobre el Game Feel	21
3. Objetivos	26
4. Metodología.....	27
4.1. Elección de metodología	27
4.2. Evolución del proyecto	28
4.3. Tareas, tiempo y reuniones.....	29
5. Desarrollo	33
5.1. Tecnologías y herramientas	33
5.2. Decisiones y elección de la demo para aplicar el Game Feel	33
5.3. Objetivos y pautas del desarrollo.....	35
5.4. Implementación de los cambios a las mecánicas	37
5.5. Diseño y definición de los niveles	41
5.6. Interfaz y build para móvil.....	45
6. Estudio económico	50
7. Resultados.....	55
7.1. Descripción del estudio	55
7.2. Diseño de las encuestas	58
7.3. Resultados de las encuestas y análisis	60
7.3.1. <i>Resultados del Nivel 1</i>	<i>61</i>
7.3.2. <i>Resultados del Nivel 2</i>	<i>63</i>
7.3.3. <i>Resultados del Nivel 3</i>	<i>64</i>
7.3.4. <i>Respuestas libres</i>	<i>66</i>
7.3.5. <i>Análisis de los resultados</i>	<i>67</i>
7.3.6. <i>Test final y resultados</i>	<i>69</i>
7.3.7. <i>Conclusiones del estudio.....</i>	<i>71</i>
8. Conclusiones.....	73
9. Bibliografía	75

Anexo I – Propuesta de proyecto	77
Anexo II – Reuniones	78

Resumen

La sensación de juego es un concepto que pese a ser relativamente nuevo lleva presente desde el inicio de los videojuegos ya que siempre se ha intentado crear un videojuego que sea divertido, agradable y en general una buena experiencia para el jugador. En los últimos años conforme la industria del videojuego ha ido creciendo este término se ha popularizado entre los desarrolladores y se han realizado trabajos sobre él y como crear una buena sensación de juego.

El objetivo de este proyecto ha sido estudiar el concepto de la sensación de juego, crear una demo jugable que aplique esta sensación y realizar un estudio sobre ello. Más concretamente, la demo ha sido realizada como múltiples experiencias en las que los usuarios tendrían distintas sensaciones de juego y finalmente se recopilarían estas sensaciones en un estudio para determinar cuál ha sido la mejor experiencia.

Palabras Clave / Keywords

Sensación de juego, Game Feel, movimiento, control, jugabilidad, apartado jugable, pulsación, métrica.

Game Feel, movement, control, playable aspect, playability, press, metric.

Abstract

Game Feel is a concept that despite being relatively new has been present since the beginning of video games since the objective has always been to create a video game that is fun, enjoyable and in general a good experience for the player. In recent years, as the video game industry has grown, this term has become popular among developers and work has been done on it and how to create a good gaming sensation.

The objective of this project has been to study the concept of Game Feel, create a playable demo that applies this sensation and carry out a study on it. More specifically, the demo has been made as multiple experiences in which users would have different sensations of the game and finally these sensations would be compiled in a study to determine which has been the best experience.

1. Introducción

Desde los inicios de los videojuegos se ha intentado que el videojuego sea un objeto de entretenimiento para las masas, desde los más jóvenes hasta los más adultos. El desarrollo de un videojuego conlleva trabajar distintos apartados que son completamente diferentes entre sí como por ejemplo el arte o apartado visual, apartado sonoro, mecánicas jugables, diseño de niveles, etc. Todos estos apartados conjuntos crean un videojuego, y a su vez, crean una sensación de juego.

Cuando alguien juega a un juego espera que la jugabilidad se sienta bien, que al pulsar los botones se realicen las acciones esperadas de la forma que cree que van a realizarse, que el movimiento sea fluido y satisfactorio, que el propio jugar por jugar sea entretenido, pero en el momento que existe una ausencia de este se crea un problema para el jugador y el diseñador.

Cuando se percibe esta falta de sensación gratificante al jugar, se sacan a relucir apartados defectuosos en el juego como podría ser un movimiento tosco del personaje o una sensación de que el juego responde mal a las interacciones con el controlador. Esto pues supone un fallo de diseño a la hora de crear un juego, pues los jugadores no encuentran satisfactoria la experiencia y puede dar lugar a malas ventas, críticas negativas, etc.

Por lo tanto, ¿cómo se define la buena sensación que debe tener un jugador cuando juega a un juego? Es aquí donde los diseñadores de videojuegos definen el concepto de Game Feel, más concretamente en el libro "Game Feel - A Game Designer's Guide to Virtual Sensation" [1], donde su autor, Steve Swink, describe el Game Feel como "el control en tiempo real de objetos en un espacio simulado enfatizado por el pulido".

Estas bases sentarían un referente para los diseñadores de videojuegos que quieren entender cómo se puede crear un juego divertido e interesante para el jugador, con unas pautas y ejemplos que permitirían guiar al diseñador sobre qué es considerado una buena sensación de juego y que no. Con esta referencia [1] y aplicado a este proyecto, resulta relevante entender cómo se puede conseguir una buena sensación para el jugador y que fundamentos la definen.

El objetivo de este proyecto se ha basado en dos partes: la primera, explicar los conceptos del Game Feel desde las fuentes más sólidas y complementar con aportes actuales al concepto. La segunda, aplicar lo aprendido sobre el Game Feel en un videojuego al que se disponga acceso del código, modificarlo y alterar su sensación de juego, y finalmente poder realizar un estudio con usuarios finales, permitiendo así obtener unas estadísticas en base a los resultados obtenidos y determinar qué sería lo necesario para crear una sensación de juego adecuada, con la que luego se crearía una demo final y se pondría de nuevo a prueba a los usuarios para verificar si la sensación ha mejorado.

2. Estado del Arte

En esta sección se hablará sobre los fundamentos del Game Feel, cómo se crea, y las métricas que resulten relevantes para el desarrollo del proyecto. Además, se incluirá un apartado con nuevas cuestiones y estudios sobre el tema.

2.1. Game Feel

En este apartado se explicarán los principios fundamentales del Game Feel y como se definen las experiencias de la sensación de juego.

2.1.1. Fundamentos del Game Feel

Los fundamentos en los que se basa la sensación de juego son el control en tiempo real, el espacio simulado y el pulido. Estos fundamentos se definen en [1] como “los tres bloques de construcción del Game Feel”, basándose en la experiencia que puede sentir el jugador a la hora de jugar al videojuego.

En el aspecto del control en tiempo real, el autor se refiere al apartado en el que el usuario puede interactuar con el videojuego de cualquier forma posible, ya sea a través de un mando en una consola, un teclado en un PC o el movimiento de los brazos en VR.

El usuario quiere provocar una interacción en el videojuego a través de una acción, y espera que el juego reaccione de una manera concreta produciendo una respuesta. Por ejemplo, en Super Mario Bros [2], cuando el jugador pulsa el botón “A”, la respuesta esperada es que Mario salte en la pantalla.

Al final, la interacción entre el jugador y el videojuego se compara con dos interlocutores que envían y reciben mensajes mutuamente. El jugador manda un mensaje, la maquina lo interpreta, crea una reacción en el juego que el jugador percibe de forma visual o sonora y vuelve a producir una respuesta en el mismo, tal y como se ve reflejado en la Figura 1.

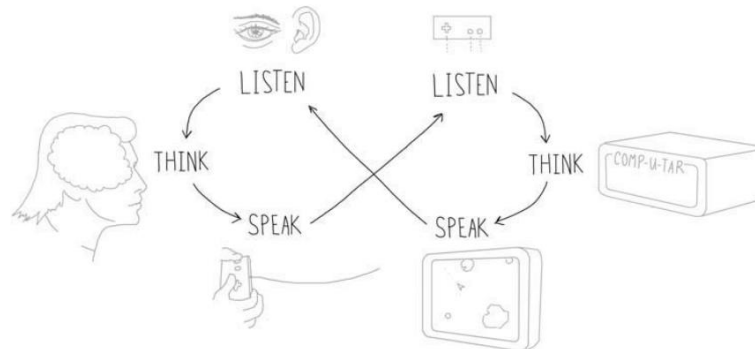


Figura 1 - Flujo de mensajes entre el jugador y el ordenador. Fuente: [1]

El control en tiempo real necesita ser complementado por el resto de los fundamentos de la sensación de juego, ya que, si un jugador intenta mover un personaje en un espacio vacío, por mucho que la respuesta realice una acción, no sirve de mucho si no hay un entorno en el que ejecutar ese control del movimiento. Es este problema lo que lleva al siguiente punto.

El espacio simulado es lo que define el entorno en el que el usuario puede visualizar las consecuencias que tienen sus acciones con las señales que manda al juego. En un espacio virtual se puede conseguir una simulación de físicas similar a la realidad donde el usuario pueda apreciar la sensación de movimiento en su personaje como puede ser la aceleración y velocidad, mientras que, con la ausencia de este, no se puede conseguir una respuesta aceptable para el usuario ya que sus acciones no tendrían consecuencias en el entorno.

Poniendo de ejemplo un juego de plataformas, si no existe un espacio sobre el que moverse, manejar al personaje o saltar deja de tener significado.

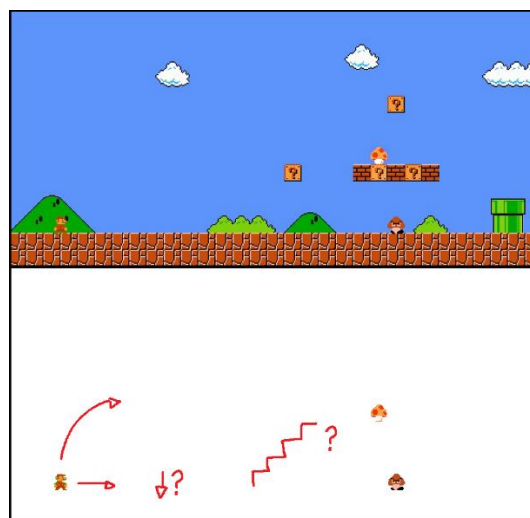


Figura 2 - Nivel 1 de Super Mario Bros (arriba) / interpretación sin espacio simulado (abajo).

Como se puede apreciar en la Figura 2, la interpretación del primer nivel de Super Mario Bros sin un espacio disponible crea una confusión para el jugador.

Sin una interpretación del espacio en el videojuego no se puede crear una sensación de juego. En la imagen aparecen tres elementos. El jugador representado como Mario, un enemigo que se mueve hacia los lados, y un objeto por encima de él. Para llegar hasta el objeto no se puede saber el camino que tiene que seguir el jugador dado que no hay ninguna indicación visual de qué entorno le rodea. Al igual, una colisión con el entorno puede indicar un espacio en el juego sin necesidad de tener una representación visual, ya que un bloque del juego puede crear una colisión con el personaje y evitar que este avance hacia el eje en el que se dirige. Por lo tanto, para que exista un control en tiempo real se necesita un espacio en el que ese control y respuesta tenga sentido.

El tercer fundamento trata el pulido de las características del videojuego que aportan a la sensación de juego. Cualquier efecto que mejore de forma artificial una interacción como puede ser el movimiento con un cambio en la animación del personaje cuando se realiza una acción más rápida, un temblor en la cámara cuando cae un objeto pesado en la escena o simplemente los efectos generales que aportan al apartado visual del juego.

Todos esos elementos crean un énfasis en la interacción del jugador con el videojuego, ya que pueden cambiar por completo la forma en la que se disfruta un juego. En referencia a [1], Steve Swink pone el ejemplo de "De Blob" en el que el personaje jugable se contrae y se expande mientras se mueve ya que la representación de este se define como una masa redonda y viscosa y que, gracias a esta alteración de la animación, y gracias al pulido de este apartado, consigue un efecto de algo blando y añade un énfasis a la jugabilidad, que, de otro modo, daría la sensación de estar moviendo una masa rígida como una piedra.

Finalmente, la unión de estos tres bloques fundamentales crea la sensación de juego. Para que exista una sensación de juego se requiere que el juego posea el control en tiempo real, la simulación espacial y el pulido.

2.1.2. Definición de las experiencias de la sensación de juego.

Una vez se completan los tres bloques de la sensación de juego, se pueden sacar a relucir las experiencias de la sensación jugable.

Steve Swink destaca en [1] cinco experiencias que forman parte de la sensación de juego:

- La sensación estética del control.
- El placer de aprender, practicar y ser habilidoso.
- La extensión de los sentidos.
- La extensión de la identidad.
- La interacción con una realidad física y única dentro del juego

La primera experiencia es la sensación que provoca el controlar un personaje, manejar un coche en un circuito o volar una nave en el espacio, es decir, la experiencia que supone el sentimiento de control del apartado jugable.

La segunda es el desarrollo del jugador en la habilidad que conlleva superar los retos de un videojuego. Esta experiencia se refleja en cualquier juego que disponga de una condición de victoria y de derrota, ya que, para mejorar en algo, tiene que haber una condición negativa que refuerce ese placer por seguir adelante. El jugador puede empezar desde no sabiendo jugar, a manejar los controles básicos, y finalmente dominarlos en su totalidad para aprovechar al máximo las capacidades de control del juego. Por supuesto, esto no se limita al apartado mecánico de un juego, es decir, manejar los controles con destreza, sino también al apartado intelectual que requiere un juego, como puede ser resolver puzles en un juego de aventura o entender en profundidad las mecánicas de un juego competitivo.

La tercera es la percepción del cuerpo en el espacio de juego, la experiencia de que el jugador sea consciente del tamaño y posición que ocupa en el entorno y cómo eso le afecta, ya sea por ejemplo en un juego de tanques donde un misil sale disparado hacia él y siendo consciente del espacio que ocupa y el tamaño del misil, poder predecir si va a impactar sobre él o no. Steve Swink hace una similitud con la conducción en la vida real, donde conforme se adquiere experiencia conduciendo, se acaba percibiendo el tamaño que ocupa un coche como si fuese una extensión de él mismo, pese a que el espacio ocupado por una persona es mucho menor, permitiendo así saber cuánto puede acercarse a otro coche, si va a poder ser aparcado en una plaza o si va a caber por una calle. [1]

La cuarta hace referencia a la representación de la persona dentro del propio juego. En juegos como Tetris resulta difícil crear una experiencia que extienda la identidad del jugador ya que simplemente se manejan fichas moviéndose en un recuadro, pero, en juegos como Super Mario Bros donde se toma el control de un personaje con forma humana, resulta más sencillo. Esto es debido a que el jugador tiende a verse representado en lo que percibe y controla. Por poner un

ejemplo, si el jugador es eliminado en Call Of Duty, su respuesta típica sería “me han matado” en vez de “mi personaje ha muerto”, porque en ese momento la identidad del jugador pasa a ser percibida como la identidad del personaje jugable.

Finalmente, la quinta es la experiencia que produce el aspecto físico del videojuego, generalmente enfocado en las colisiones con los objetos del entorno, tamaño y peso respecto al jugador, y como se plasma esa sensación de espacio real en un entorno virtual.

En la Figura 3, se pueden observar los tipos de experiencias que se crean al destacar los bloques fundamentales de la sensación de juego.

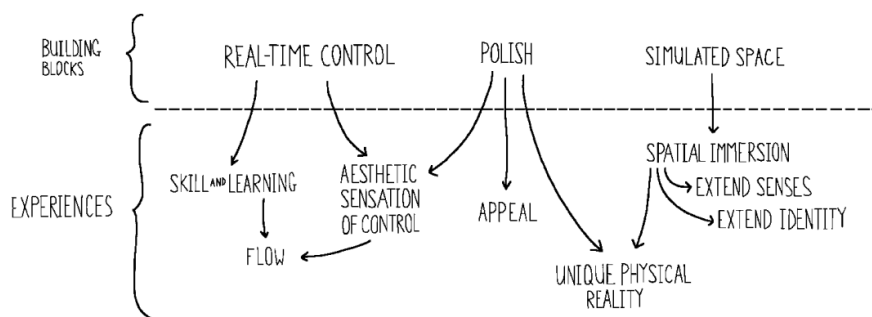


Figura 3 - Diagrama de los bloques fundamentales traducidos en experiencias. Fuente: [1]

2.2. Creación del Game Feel

En el apartado 2.1 se han tratado las experiencias que genera el videojuego en el jugador, y para afirmar que un juego tiene una buena sensación de juego, tienen que producirse algunas de estas experiencias.

Dado que el enfoque de este proyecto es la creación o modificación de un pequeño videojuego al que aplicar los conocimientos adquiridos sobre el Game Feel, resulta relevante hablar sobre cómo se pueden crear estas experiencias en un juego y debido al apartado en el que se va a centrar el proyecto, se ha decidido profundizar en el apartado de la estética del control, habilidad y controles intuitivos.

2.2.1. Game Feel como sensación estética del control

Según el tipo de control que disponga el juego se puede conseguir una verdadera sensación estética de control. Juegos que no manejan un avatar, juegos de estrategia, juegos de cartas, o

juegos por turnos, entre otros, no disponen de la capacidad para crear una sensación estética de control dado que su jugabilidad no se enfatiza en esa área.

Los juegos que se pueden especializar en esta sensación son aquellos que disponen de un objeto o personaje jugable que se mueve en un espacio virtual. Para ello, los movimientos deben asignarse a los inputs que va a realizar el jugador. Poniendo de ejemplo un movimiento en dos dimensiones, lo básico es que el jugador pueda realizar movimientos en el eje X (o Y en algunos casos), y en caso de que haya verticalidad en el juego se incluye el eje Y. Necesitaría pues dos botones para trasladarse en cada una de las direcciones por cada eje. En caso de ser un movimiento con gravedad en el eje Y, un botón que diese una capacidad de salto, como se puede observar en la Figura 4.

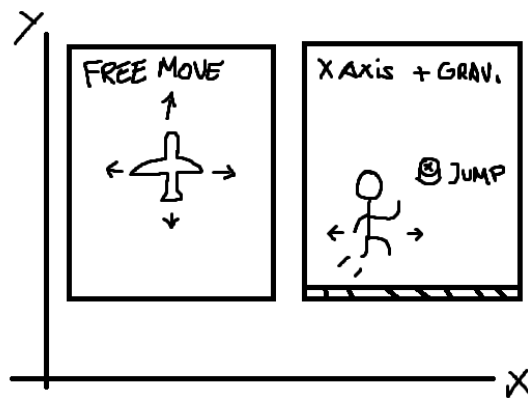


Figura 4 - Distinción entre espacios en dos dimensiones con movimiento libre y movimiento con gravedad.

Steve Swink denomina en [1] el mapeo de acciones como una forma de “expresión estética” en la que el jugador puede utilizar el movimiento como una herramienta de mociones para desplazarse y jugar con el entorno virtual. Estas acciones denotan unas reglas o normas de lo que se puede realizar con los inputs que introduce el jugador y que se aplican en el espacio jugable.

Por supuesto, el set de movimientos y acciones que puede realizar el jugador no determinan lo que éste experimenta por sí solos, si no la experiencia sería muy aburrida. En algunos juegos, según el pulido del apartado del movimiento, se puede llegar a sentir una experiencia divertida con tal de utilizar el movimiento como herramienta. Poniendo de ejemplo, Super Mario 64 [3], un juego en el que se dispone de un set avanzado de movimientos como el salto básico, salto doble, salto triple, salto agachado y otros más, que permiten al jugador expresarse en cómo quiere moverse. Con ello, el simple hecho de practicar el movimiento a través del espacio puede resultar en una experiencia entretenida, pero para ello debe haber retos, ya sean

impuestos por el propio videojuego o por el jugador. Por ejemplo, un reto del videojuego sería obtener la estrella del nivel en el que está el jugador, por el cual debe avanzar a través de una serie de plataformas, superar a ciertos enemigos o evadir trampas que puedan entorpecerle, mientras que un reto propuesto por el jugador puede ser alcanzar una plataforma concreta con el mínimo número de botones pulsados o con la mayor precisión posible.

2.2.2. Game Feel como la experiencia de aprender, practicar y dominar una habilidad

Ahora, con estas reglas y retos, se introduce el concepto de habilidad en un videojuego. No es lo mismo la habilidad de la que dispone un jugador tras haber jugado 5 minutos a un juego que otro jugador que lleva varias horas. El manejo de los controles es distinto, la precisión con la que se introducen los inputs cambia y el control del espacio que manejan ambos jugadores se vuelve dispar. Cuanto más complejas sean las normas y reglas del juego, y conforme más horas se dediquen los jugadores a entenderlas, mejor se vuelve su experiencia jugable y su habilidad para superar los retos. Por supuesto esto no se aplica únicamente a la habilidad mecánica con los inputs sino también a la habilidad de control sobre el estado del juego y la toma de decisiones que puede resultar en la superación de un reto o el fallo intento de completarlo.

De aquí, se relaciona la habilidad con la sensación de juego de tres formas:

- El desafío altera la forma que tienen de jugar los jugadores y cómo experimentan la sensación de control en base a las metas y recompensas.
- La sensación de juego varía dependiendo de la habilidad del jugador.
- Los controles son intuitivos para los jugadores cuando pueden sentir que lo que "piden" al juego es lo que "ocurre" al momento.

El desafío supone un incentivo para el jugador y le exige seguir experimentando con los controles del juego para poder seguir avanzando y progresando. El jugador debe sentir que lo que está haciendo tiene un propósito y para ello se le puede ir disponiendo de desafíos que resolver, ya sea alcanzar una plataforma usando una nueva técnica de movimiento adquirida o usando lo que ya conocía y descubriendo por su propia cuenta cómo llegar. Por norma general, son estos desafíos los que permiten al jugador explorar las capacidades de movimiento de las que dispone el videojuego y desarrollar nuevas tácticas a la hora de enfrentarse a un problema.

Estos desafíos son compuestos por dos partes: las metas y los impedimentos. Y estas dos partes son lo que empiezan a marcar la habilidad de un jugador. Las metas deben ser cuanto

más concisas posibles para guiar al jugador a su objetivo y los impedimentos deben ser lo suficientemente complejos como para exigirle al jugador sin que se aburra.

Según el autor de [1], "estas dos herramientas, impedimentos y metas, permiten a los diseñadores dar forma al control en tiempo real con una sensación concreta. Las metas enfatizan ciertas partes del movimiento mientras que los impedimentos eliminan otras. El resultado es la sensación que el diseñador quería tener". Con un apartado jugable bien implementado, una buena sensación de control del movimiento y múltiples elementos con los que interactuar, el control se convierte en una habilidad para el jugador.

Esto lleva al segundo punto y es que la sensación de juego varía según la habilidad del jugador. Conforme un jugador va acumulando horas con el juego se presupone que se va volviendo más habilidoso. Aunque en la teoría esto sea así, en la práctica se pueden encontrar casos muy ambiguos.

Hay perfiles de jugadores que prefieren no dedicar mucho tiempo al aprendizaje de este o que en el momento que encuentran una dificultad mayor a lo que conocen, prefieren parar de intentarlo y dejar de lado el videojuego. En ese caso, la sensación de juego que habría obtenido el jugador sería una mala experiencia en el ámbito jugable. Puede haber estado disfrutando el apartado artístico y lúdico del juego como la historia y los escenarios, pero el apartado jugable le ha superado y prefiere dedicar su tiempo a otra cosa. Esto genera una experiencia dispar para los jugadores y dos personas que han jugado al mismo juego pueden no haber recibido la misma experiencia o sensación al jugarlo.

Para ello debe tenerse en cuenta un ciclo de progreso en el que, por naturaleza, el jugador al principio fallará en lograr las metas, pero poco a poco conseguirá solucionarlas.

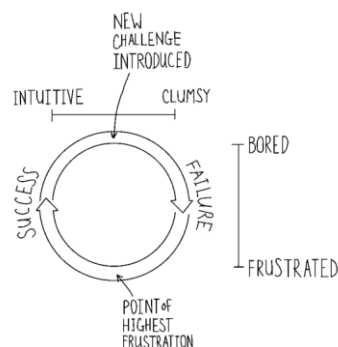


Figura 5 - Ciclo de la habilidad del jugador y la sensación de juego. Fuente: [1]

En la Figura 5 se representan las experiencias del jugador cuando se enfrenta a los retos y las sensaciones que aparecen conforme se va siguiendo el ciclo. Para lograr la meta se tiene que encontrar algo de frustración, de modo que cuando se complete el desafío, sea satisfactorio haber llegado a la meta. Los controles se deben sentir intuitivos en el momento de lograrlo y algo torpes cuando se está aprendiendo, ya que, si todo es exactamente igual que cuando se logra la primera meta, no se crean desafíos en la sensación estética del control.

Por lo tanto, se necesita un balance entre la experimentación de la frustración y el aburrimiento para poder encontrar un punto medio en el que el jugador se entretenga sin aburrirse demasiado y sin frustrarse demasiado, ya que, en cualquiera de los dos extremos, puede llevarle a abandonar la experiencia.

Este punto medio es el que se denomina como "Flow" o flujo, también conocido como "entrar en la zona", en el cual el nivel de habilidad que exige el juego es parejo con la habilidad del jugador y este entra en un estado en el que su punto de concentración se centra en el juego para poder explotar al máximo su potencial. En este estado es normal que para el jugador las horas se sientan como minutos y que el disfrute de la experiencia alcance su pico más álgido.

El desarrollo de este flujo no es algo que se pueda conseguir de forma constante. Es algo que ocurre de forma natural y supone la experiencia ideal para el jugador. El desarrollador puede intentar crear situaciones donde esta experiencia sea más fácil de alcanzar para el jugador, pero es algo que ocurre con el desarrollo natural de la habilidad del jugador y con una buena curva de dificultad en el videojuego.

Finalmente, el tercer punto define lo que un jugador espera cuando utiliza los controles de un juego. Para el jugador, es importante que las acciones que toma se vean representadas en el juego cuanto antes. No tiene sentido que si quiere saltar a una plataforma mientras corre, al pulsar el botón de saltar, tarde más de lo que debería y provoque su caída al vacío. Cuando esto ocurre se crea una sensación de que su intención no se está realizando de forma correcta en el juego y da una impresión negativa del apartado jugable. Esto puede ser causado por una falta de habilidad por parte del jugador, o por un fallo en el diseño o programación del juego. Por lo tanto, para crear una buena sensación de juego es importante que los controles se sientan lo más precisos posibles. Esto se traduce en que es necesario crear una respuesta del videojuego casi inmediata desde que el jugador declara su intencionalidad pulsando un botón hasta que se realiza la acción en la pantalla.

La rapidez de la respuesta en los controles es de las partes más importantes que definen el apartado jugable en los videojuegos que requieren habilidad y precisión por parte del jugador.

En un juego de estrategia por turnos quizá no sea el factor más determinante. Poniendo un ejemplo en juegos físicos, en el ajedrez no es tan relevante que una persona se mueva lentamente para realizar el movimiento de su ficha respecto a lo que puede ser relevante en un partido de fútbol donde el futbolista debe realizar una finta y chutar el balón. La intencionalidad del jugador se ve reflejada en estos casos por su cuerpo real y la respuesta depende de su estado físico, sólo que en según qué tipo de exigencias pida el juego, se puede permitir un retardo en la respuesta. Esto lleva al siguiente tema sobre el control en tiempo real donde se tratará como recibe las respuestas el jugador y como afectan al ciclo del juego.

2.2.3. Control en Tiempo Real

La interacción del jugador con el juego crea un ciclo continuo de “preguntas” del jugador y “respuestas” del juego. Continuamente el jugador está mandando instrucciones al juego que tienen que ejecutarse en el menor tiempo posible y este debe reaccionar correctamente a esas instrucciones.

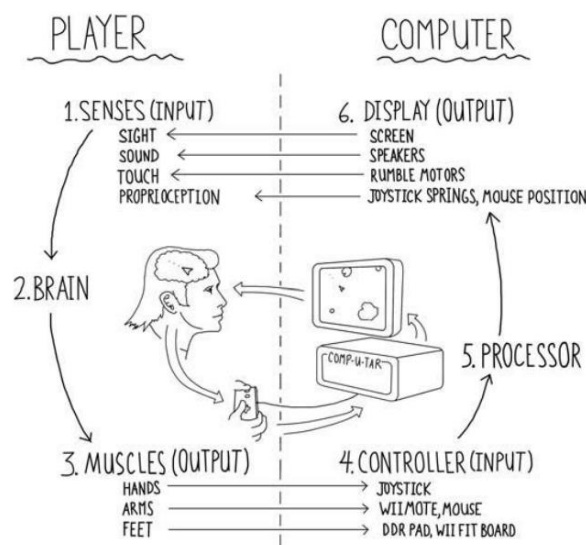


Figura 6 - Ciclo de interacción entre el jugador y el dispositivo en un videojuego. Fuente: [1]

En la figura 6 se puede observar cómo funciona el ciclo de interacción entre el jugador y el juego:

1. Primeramente, el jugador percibe a través de los sentidos un estímulo en el juego que le va a incitar a hacer una acción.
2. Después, el cerebro analiza ese estímulo y en base a los conocimientos de los que dispone decide cual es la respuesta correcta.
3. La señal se manda a las extremidades, ya sea pulsar un botón, agitar el brazo, mover un joystick...

4. Se pulsa el botón del mando y se registra la acción en el controlador del mando.
5. Esta se manda a través del cable al dispositivo de juego y se procesa.
6. Una vez procesada, se refleja la acción en el juego y vuelve a crear un estímulo para el jugador.

Este ciclo de interacción es el que determina cómo funciona el control en tiempo real, pues el ciclo debe ser constante y sin interrupciones. Un ciclo que tarde demasiado en procesar las pulsaciones de los botones del jugador crearía un retardo en la respuesta por pantalla y habría un corte en la sensación de juego que se crea cuando el ciclo es constante.

Para un ser humano, desde que se procesa la información hasta que se manda la respuesta a los músculos se estipula que tarda una media de 240 milisegundos. Esta se divide en tres procesos:

- El procesador perceptivo con una media de 100ms y un rango entre 50ms y 200ms.
- El procesador cognitivo con una media de 70ms y un rango de 30ms y 100ms.
- El procesador motor con una media de 70ms y rangos entre 25ms y 170ms.

Por lo tanto, para que el flujo de interacción entre el jugador y el juego se mantenga constante, el juego debe tener un proceso desde que se recibe la respuesta hasta que se procesa en pantalla de unos 100ms mínimo.

Esto se debe a que el límite en el que el ser humano detecta una imagen en movimiento se ronda entorno a las 10 imágenes por segundo, por lo que, para obtener una mayor sensación de control en tiempo real, el juego debe reproducir una imagen cada 100ms y procesar la respuesta del jugador como mínimo en 240ms.

Entonces para que un juego mantenga una sensación de control en tiempo real, en el apartado del dispositivo, se deben cumplir una serie de factores y valores:

- La impresión de que el juego está en movimiento.
- Una respuesta inmediata.
- Continuidad del ciclo de respuesta.

Para que se dé una impresión de movimiento, como se ha dicho antes, el ser humano percibe movimiento a los 10 fotogramas por segundo. Esta cifra roza el límite, por lo que se pretende que cuanto mayor sean los fotogramas por segundo, mayor sensación de movimiento tendrá el



jugador. La cifra que se busca alcanzar en los tiempos actuales ronda entre los 30 fps y 60 fps (fotogramas por segundo) dado que a mayor sea esta cifra, menor será el sentimiento de que el juego va lento y más inmersiva será la experiencia. También, actualmente en los juegos competitivos y en el mundo de los videojuegos se ha formado una moda de intentar alcanzar la mayor cantidad de fotogramas por segundo posibles, por ejemplo, en los juegos competitivos donde se recomienda tener un monitor de 144hz para poder visualizar los 144 fps en el juego y poder estar a la altura de la competición. [4]

Esto es debido a que el ojo humano puede alcanzar a ver detalles a más de 60 fps y reaccionar entorno a ello. Aunque no hay estudios que puedan asegurar con certeza la cantidad de imágenes por segundo que puede procesar un ser humano, se puede ver que en los dispositivos de realidad virtual se busca alcanzar un procesamiento de fotogramas de unos 90 fps bajando hasta la mínima de 72 fps, ya que, de otro modo, el jugador puede empezar a marearse. [5]

En cuanto a la respuesta, hay un margen en el que se puede considerar instantánea y un margen en el que se considera lenta. Hasta un máximo de 100ms se considera una respuesta estándar. El problema sucede cuando ese tiempo se alarga, y por supuesto en el mundo real la respuesta se considera inmediata, pero en el procesado de un videojuego esta respuesta se alarga. En el libro, Steve Swink cita a Mick West, un programador-diseñador del primer Tony Hawk en el que define el "lag" como el retardo desde que el usuario interactúa con el dispositivo hasta el momento en el que obtiene la reacción del juego y también destaca que un tiempo de respuesta de entre 50ms y 100ms es lo óptimo para que el jugador sienta la sensación de inmediatez de la respuesta. [1]

Como se puede observar en la Figura 7, cualquier tiempo comprendido entre los 0ms y 100ms se considera instantáneo, mientras que a partir de los 150ms se considera lenta hasta un máximo de 240ms donde se rompería totalmente la inmersión del control en tiempo real.

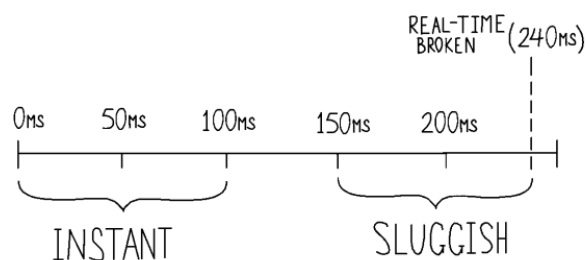


Figura 7 - Percepción del jugador en base al tiempo de respuesta. Fuente: [1]

Finalmente, para que el movimiento y la inmediatez de las respuestas tengan un sentido, se necesita un ciclo de respuesta que sea continuo.

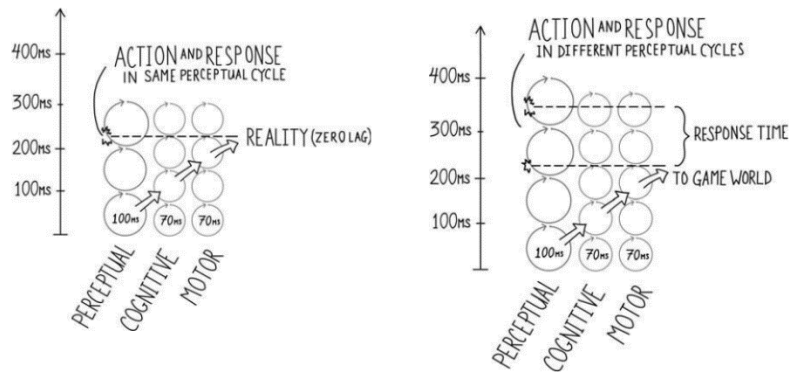


Figura 8 - Ciclo de acción y respuesta. Mundo real vs mundo virtual. Fuente: [1]

En la Figura 8 se muestra la diferencia entre los ciclos de acción y respuesta. Mientras que en la vida real se encuentra una respuesta inmediata con nada de retardo, en el mundo virtual tenemos un tiempo de respuesta a parte del propio retardo de los sentidos. Esto significa que el juego tiene que estar recibiendo cualquier tipo de entrada de control en cualquier momento para mantener la sensación de control en tiempo real. Como se ha dicho antes, estar procesando todo cada 100ms es suficiente como para obtener la sensación.

Sin embargo, esta respuesta puede variar dependiendo de la situación en la que se encuentre el dispositivo, ya puede ser que este cargando demasiados efectos en pantalla y la tasa de imágenes decaiga, o por propias mecánicas del videojuego le interese al desarrollador que la respuesta visual no se correlacione con la entrada del jugador, como por ejemplo en los juegos de lucha donde una vez realizado un movimiento, el jugador debe esperar hasta que finalice toda la animación del movimiento hasta poder volver a moverse o atacar, e incluso en estos juegos no se corta la sensación de control.

Por lo tanto, la respuesta se puede alterar sin interrumpir la sensación de control, y se puede crear un ajuste al juego que permita esas alteraciones sin que el jugador vea que ha habido una interrupción en el flujo de control.

2.3. Métricas del Game Feel

En este apartado se hablará brevemente las métricas que se utilizan para medir la sensación de juego y se profundizará en la métrica de respuesta.

2.3.1. Resumen de las Métricas

Las métricas se tratan en [1] como métricas de entrada, métricas de respuesta, métricas de contexto, métricas de pulido, métricas de metáfora y métricas de reglas. Cada una trata partes específicas del juego:

- Entrada: Los inputs que componen el juego, que tipo de controles se utilizan, por qué se han usado los botones que se han usado...
- Respuesta: La reacción del juego frente a la información que le manda el usuario, como opera las pulsaciones del jugador...
- Contexto: Como se relaciona el espacio virtual con la interacción, el control en tiempo real con las mecánicas que implementa el juego...
- Pulido: La mejora visual, sonora, efectos que tienen lugar en el juego y mejoran artificialmente la impresión del jugador...
- Metáfora: La expectativa del jugador respecto a las representaciones virtuales de los objetos...
- Reglas: Las relaciones en las variables que crean retos para el jugador, alteran sensaciones...

Concretamente para este estudio se ha barajado utilizar distintas métricas como la de contexto, reglas y respuesta, pero para acotar el estudio en una sola métrica se escogió la métrica de respuesta, ya que resulta relevante a la hora de medir la sensación de juego cuando se modifiquen los parámetros que afectan al control del personaje y su movimiento.

2.3.2. Métrica de Respuesta

La métrica de respuesta se trata de la información que envía el jugador al dispositivo, donde puede ser procesada e interpretada por el código del juego y que luego se percibe en el mundo virtual. Con la entrada de una pulsación de un botón, se debe reflejar en el videojuego esa pulsación en base a las mecánicas que este implemente, ya sea el movimiento direccional, un salto, agacharse, etc.

Steve Swink habla en [1] sobre el ADSR (*Attack, Decay, Sustain, Release*), un término musical que describe una curva de cómo progresa la amplitud del sonido de un instrumento musical a lo largo del tiempo como en la Figura 9.

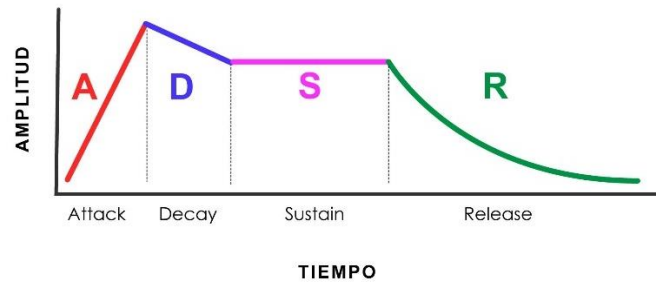


Figura 9 - Representación gráfica del ADSR. Fuente: <https://djexpressions.net/que-es-el-adsr-y-como-podemos-aprovecharlo/>

Este término lo traslada a un entorno digital, donde la métrica establecida para esa pulsación es la velocidad que alcanza el personaje. Poniendo de ejemplo, un personaje moviéndose en un entorno bidimensional en el que puede correr a la izquierda o a la derecha, cuando se pulse el botón de moverse a la derecha, habrá un tiempo desde que se pulsa el botón, se mantiene pulsado, y se suelta. En el tiempo que transcurre la pulsación y el levantamiento del botón debe producirse un efecto en el mundo virtual.

Esto se puede representar con una gráfica de ADSR donde realmente solo estará compuesta por ASR (Attack, Sustain, Release). Esto es debido a que el mapeo de una acción a un botón convencional solo tiene dos estados, o pulsado, o no pulsado.

En este caso se analiza el ASR con la velocidad que adquiere del personaje cuando se pulsa el botón como se puede observar en la Figura 10.

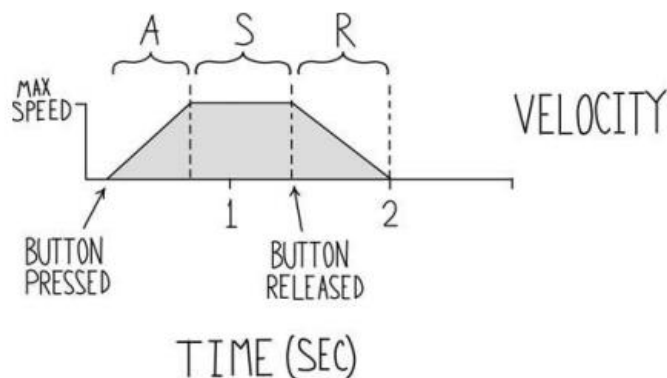


Figura 10 - Representación de la velocidad con la pulsación de un botón. Fuente: [1]

El *Attack* supondrá la primera fase en la que el botón se presiona, donde se empieza a crear el *momento* de la aceleración y poco a poco el personaje va ganando velocidad.

El *Sustain* supone el momento en el que la velocidad alcanza su máximo valor mientras se sigue pulsando el botón.

Finalmente, el *Release* conforma la parte final donde tras soltar el botón la velocidad comienza a disminuir hasta reducirse a cero completamente.

La variación de tiempo entre que comienza el *Attack* y empieza el *Sustain* es una de las partes más importantes a la hora de crear una buena sensación de juego. Dependiendo del tiempo que se tarde en iniciar la acción y llegar al punto álgido, el jugador puede sentirse frustrado en caso de que sea muy tardío o si es demasiado temprano puede crear una sensación de inmediatez que se vuelva incómoda.

Este análisis del ASR se utilizará a la hora de crear la sensación de juego para la demo y finalmente servirá como referencia para el análisis de los resultados que se obtengan en las pruebas finales.

2.4. Otras referencias sobre el Game Feel

Dado a que la primera parte del estado del arte se ha centrado en explicar el concepto del Game Feel y su aplicación tomando de base la referencia [1], es relevante hablar también sobre nuevos aportes sobre el concepto o trabajos pasados que hagan referencia al mismo.

El concepto de Game Feel o sensación de juego, como bien dice el nombre, se basa en una "sensación" de una persona a la hora de interactuar con un juego por lo que su definición no es algo inamovible ya que se basa en algo totalmente subjetivo.

Steve Swink ya habló sobre el tema un año antes de lanzar [1] donde definía el Game Feel como "el ingrediente secreto" en [6] y daba unas trazas de cómo crear la sensación de juego. En él, describía como se desgrana Super Mario 64 hasta el nivel de jugabilidad más mínimo: el movimiento. Este es el apartado que más influye en la jugabilidad de Super Mario 64 y en el artículo crea unas pautas sobre cómo crear un "ecosistema del jardín" refiriéndose a las partes necesarias de un videojuego para crear esa sensación de juego satisfactoria.

Para conseguir diseccionar los problemas que incumben a cada apartado del juego se recomienda dividir los apartados en diversas partes:

- La entrada o input, que representa la interacción del usuario con el juego.

- La respuesta, que representa lo que procesa el juego.
- El contexto, sobre como el espacio da sentido al movimiento.
- El pulido, que aporta a la experiencia general con el uso de animaciones, efectos y sonidos.
- La metáfora, que refleja lo que representa en el juego algo a lo que el jugador está conectado.
- Las reglas, que definen unas medidas para enfatizar la sensación de control.

Estas partes luego se plasmarían en [1] y conformarían el apartado de "Métricas del Game Feel".

También hay artículos científicos sobre el tema como "Designing Game Feel. A survey" [7], en el que se propone que el concepto de control en tiempo real que se aporta en [1] solo se aplica a videojuegos que dispongan de un movimiento controlado por el jugador y que este sea en tiempo real, por lo que ignora completamente los juegos que no dispongan de esa característica.

Además, propone un punto de vista distinto respecto a los elementos fundamentales de diseño de videojuegos, partiendo de unos dominios de las experiencias del jugador representadas como "Faceta física", "Amplificación" y "Soporte", para después diseccionar los apartados que conforman el diseño de videojuegos en "Movimiento y acciones", "Significado de eventos", "Manipulación del tiempo", "Persistencia" y "Encuadre de escenas".

TABLE I
GAME FEEL DESIGN ELEMENTS OVERVIEW.

Design Element	Physicality	Amplification	Support	Key References
Movement and Actions				
Basic Movement	•			[89]–[95]
Gravity	•			[92], [96], [97]
Terminal Velocity			•	[92]
Coyote Time			•	[98], [99]
Invincibility Frames			•	[100]–[102]
Corner Correction			•	[28], [103]
Collision Shapes	•		•	[104]
Button Caching			•	[92]
Spring-locked Modes	•			[5], [105], [106]
Assisted Aiming			•	[28], [107], [108]
Event Signification				
Screen Shake	•	•	•	[8], [73], [109], [110]
Knock-back & Recoil	•	•		[8], [111]
One-shot Particle Effects		•	•	[73], [112]–[117]
Cooldown Visualisation			•	[118]–[121]
Ragdoll Physics	•	•	•	[1], [122]
Colour Flashing		•		[8], [71], [123]
Impact Markers		•	•	[83], [124]
Hit Stop		•	•	[81], [83], [125], [126]
Audio Feedback	•	•	•	[127]–[129]
Haptic Feedback	•	•	•	[83], [130]
Time Manipulation				
Freeze Frames		•	•	[83]
Slow Motion		•	•	[83]
Bullet Time		•	•	[131], [132]
Instant Replays		•	•	[83]
Persistence				
Trails			•	[112], [127], [133]
Decals & Debris			•	[127], [134]
Follow-Through	•			[135]
Fluid Interfaces	•		•	[136]–[138]
Idle Animations			•	[135], [139], [140]
Scene Framing				
Highlighting			•	[141]–[143]
Dynamic Camera		•	•	[71], [144]–[149]

Figura 11 - Tabla sobre los elementos del Game Feel. Fuente: [7]

En la tabla de la Figura 11 se atribuyen distintos dominios de las experiencias jugables a cada parte de los aspectos de diseño, donde se puede apreciar que cosas tan simples como el “Movimiento básico” o la “Gravedad” se le atribuyen a la “Faceta física” del juego, ya que conforman la parte que se aprecia físicamente dentro del entorno de juego, mientras que la “Velocidad Terminal” o los “Fotogramas de invencibilidad” son cosas que aportan un apoyo a la jugabilidad y experiencia del jugador.

La diseminación de los elementos del diseño de videojuegos en facetas más pequeñas, con una guía que determina a que apartado del dominio van a ser atribuidas, resulta en un gran apoyo para los desarrolladores de videojuegos que quieren entender la sensación de juego y como diversos efectos en el movimiento, eventos generados a lo largo del juego o algo tan simple como la utilización de la cámara en la escena pueden ser relevantes a la hora de crear una buena experiencia de juego.

Previo a este proyecto se ha realizado otro trabajo final de grado en la Universidad San Jorge relacionado con la cuestión del Game Feel [8], enfocado más en el pulido y control en tiempo real a la hora de crear una buena sensación de juego. A diferencia de este, el estudio de caso que se realizará en este proyecto implica la creación de una demo que altere la jugabilidad a lo largo del juego, creando así diferentes experiencias para el jugador. Con ello, se hará un estudio alrededor de los parámetros que modifican la experiencia y las sensaciones que han obtenido los jugadores, pudiendo así conseguir medir cuales son las mejores partes de cada experiencia y determinar una buena sensación de juego.

3. Objetivos

Los objetivos del proyecto se establecieron en la propuesta que se puede encontrar en el anexo. Estos no han cambiado puesto que se ha seguido el planteamiento inicial y se ha desarrollado en base a los mismos.

Tal y como se define en el documento de la propuesta, los objetivos del proyecto son:

- Conocer el estado del arte en el concepto de Game Feel de videojuegos
- Creación y aplicación de Game Feel en un videojuego
- Descripción de estudio de caso

Estos objetivos han conformado la parte de estado del arte, desarrollo y resultados del proyecto. Sobre el primero se ha estudiado sobre el concepto del Game Feel y explicado los fundamentos sobre las bases más sólidas y estado actual del concepto. Sobre el segundo se ha aplicado lo aprendido sobre una demo y llevado a un estado en el que haya una experiencia adecuada para probar. Y finalmente se ha realizado un estudio sobre lo desarrollado para determinar la mejor sensación de juego posible con usuarios reales y como sus sensaciones han influido para crear una mejor experiencia.

4. Metodología

La metodología del proyecto se estableció en la primera reunión con el director del proyecto (se puede encontrar el acta en el anexo, página 77).

4.1. Elección de metodología

Lo primero que se discutió qué metodología debía seguir. Principalmente se decidió que para el proyecto se debería utilizar una metodología que permitiese el seguimiento de las tareas que se iban encomendando, saber cuáles quedaban pendientes y cuales estaban cerradas definitivamente. Con esta idea en mente, se propuso utilizar una metodología Kanban que permitiese tener un tablón con las distintas tareas que conforman el proyecto y poco a poco ir llenando la tabla con nuevas ideas y tareas completadas. La decisión final de usar Kanban fue debido a la familiarización con el tipo de metodología, la facilidad de uso y la comodidad de poder visualizar el estado del proyecto con simplemente dar un vistazo al estado del tablero.

Una vez elegida la metodología, se buscó una herramienta que permitiese a los usuarios agregar a otros al tablón de forma colaborativa para así poder tener un seguimiento paralelo. Se escogió Trello debido a la familiarización con otros proyectos y su facilidad de uso, permitiendo crear fichas de tareas personalizadas con color para poder segmentar las tareas en los apartados del proyecto. Finalmente, conforme se fuesen haciendo reuniones y el proyecto fuese avanzando, el tablón debía cambiar.

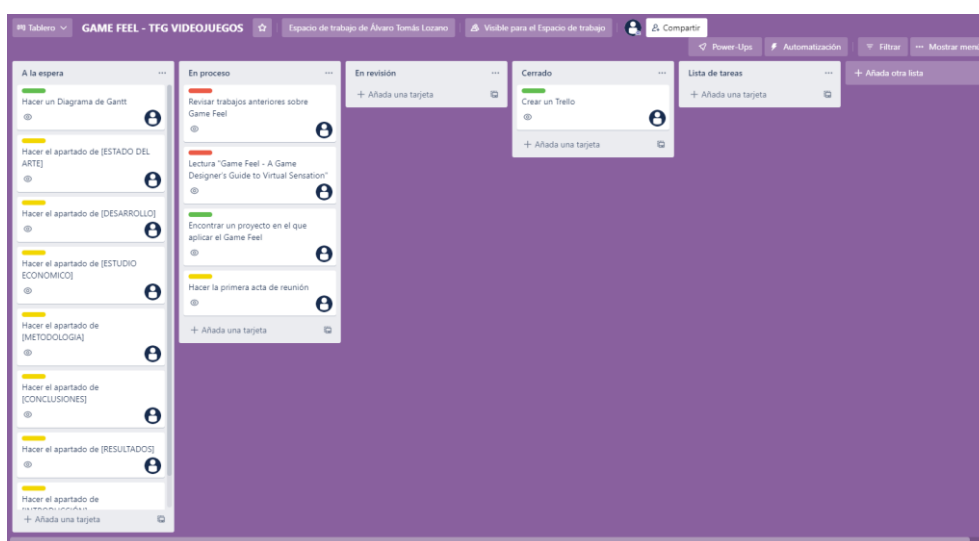


Figura 12 - Primer estado del tablón de Trello.

En la Figura 12, se puede encontrar el tablón inicial con las tareas iniciales que se pensaron para el proyecto. Las tareas al principio serían muy sencillas, pues aún no estaba definida toda la estructura del proyecto, mientras que en el tablón final debían ser tareas específicas.

Al llegar al final del proyecto, la estructura del tablón cambió para separar las tareas cerradas en distintas partes del proyecto como son la documentación, la escritura de la memoria, el desarrollo o la fase final del estudio. En la Figura 13 se plasma como acabó el tablón al final del proyecto.

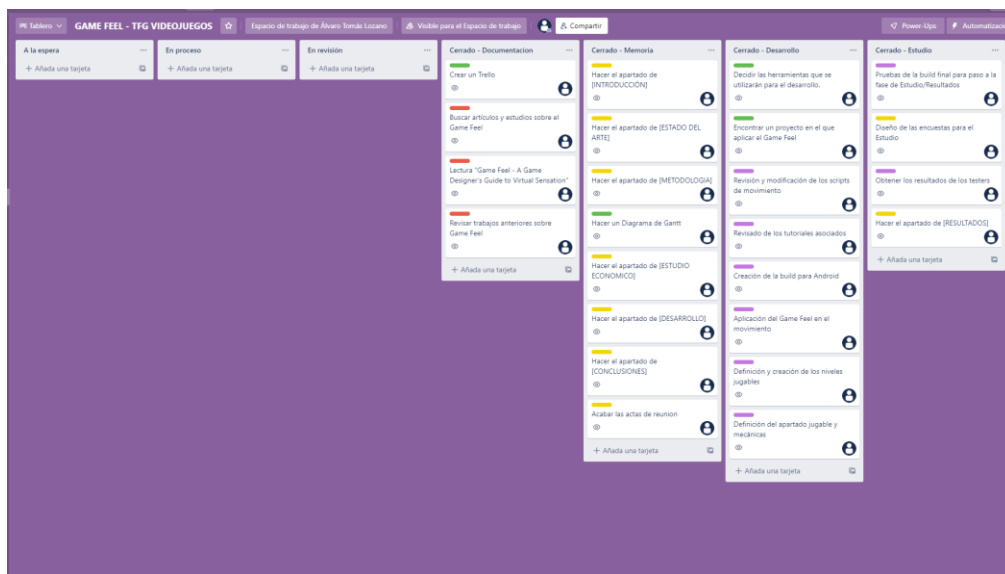


Figura 13 - Estado final del tablón de Trello.

Al principio se pensó en agrupar tanto la memoria y resultados en la misma categoría, pero como se fue desarrollando la memoria al mismo tiempo que el resto del proyecto, se consideró que se podría separar la fase final de resultados en una fase propia.

4.2. Evolución del proyecto

El proyecto se dividió en tres fases diferenciadas: La fase de documentación, la fase de implementación y la fase de resultados.

La primera se basaba en el estudio sobre el tema del Game Feel, la lectura de [1] y el estudio de nueva documentación sobre el concepto como artículos científicos, proyectos pasados, etc. Esto era necesario ya que para poder realizar las siguientes fases se debía entender el concepto de Game Feel. Por lo tanto, se escribiría primero el apartado del estado del arte con los

conocimientos adquiridos para así tener una guía sobre cómo se desarrollaría la demo final que serviría para la fase final.

Una vez terminada la parte de documentación se comenzaría el desarrollo de la demo. Primero se necesitaría definir las herramientas que se utilizarían, luego que objetivos se querían obtener con esa demo, y una vez definido todo se comenzaría a desarrollar.

Finalmente, una vez la demo estuviese acabada, se deberá probar la demo con usuarios reales y obtener sus opiniones sobre el juego pudiendo así realizar el estudio del que se expondrán los datos recabados en el apartado de resultados del proyecto. Una vez terminado, se dedicaría el resto del tiempo a dar un toque final al resto de apartados pendientes y preparar la presentación final.

4.3. Tareas, tiempo y reuniones

Las tareas que se propusieron al comienzo del proyecto fueron principalmente tareas de documentación y elección de la metodología para ir realizando un seguimiento del proyecto. En las reuniones se irían definiendo las nuevas tareas a realizar, y en ellas se puede desglosar la lista de tareas que conformarían finalmente este proyecto.

Para la fase de documentación:

- Elección de la metodología a seguir.
- Lectura del libro "Game Feel: A Game Designer's Guide to Virtual Sensation".
- Lectura de proyectos pasados que trataran sobre el Game Feel.
- Búsqueda de artículos y videos sobre el tema.
- Redacción de la memoria con lo aprendido.

Para la fase de desarrollo:

- Elección de las tecnologías necesarias.
- Elección del juego para el estudio de caso.
- Elección de las modificaciones aplicadas a la demo jugable.
- Implementación y modificación del código para crear la experiencia deseada.
- Creación de una build para móvil.

Para la fase del estudio:

- Diseño de la encuesta final.
- Recolección de opiniones de los usuarios/testers.
- Redacción y análisis sobre los resultados obtenidos en el estudio.

El tiempo de desarrollo del proyecto entero ha tomado parte a lo largo de varios meses dado que se ha ido invirtiendo tiempo poco a poco en la documentación, y se ha intensificado el tiempo invertido concretamente en los meses de verano.

Con estas tareas separadas, se realizó un diagrama de Gantt para plasmar el tiempo invertido y los meses correspondientes al desarrollo del proyecto que se puede ver en la Figura 14. Los meses se agrupan para acortar la extensión del diagrama en aquellas fases que se haya dedicado el tiempo a las mismas tareas.

Respecto al tiempo empleado que se muestra en la Figura 14 se considera que ha sido un tiempo adecuado para realizar todo lo que se quería en este proyecto, pero sí que se podrían haber acortado los meses que se dedicaron a la fase de documentación, desde que se empezó el proyecto hasta que realmente arrancó, pudiendo así concentrar los días trabajados en menos meses. Al inicio se esperaba terminar ese apartado en unos pocos meses, pero se alargó por motivos personales, y pese a un con todo, se consiguió completar el proyecto sin ningún problema.

	NOVIE. - ABRIL	MAYO	JUNIO	JULIO	AGOSTO	SEPTIEMBRE
FASE DE DOCUMENTACIÓN/MEMORIA	10 DÍAS	5 DÍAS	4 DÍAS	6 DÍAS	7 DÍAS	1 DÍAS (REVISION)
FASE DE DESARROLLO				9 DÍAS	3 DÍAS	
FASE DE RESULTADOS					6 DÍAS	1 DÍA

TIEMPO DE DOCUMENTACIÓN Y ESCRITURA MEMORIA	192 HORAS
TIEMPO DEL DESARROLLO DE LA DEMO	72 HORAS
TIEMPO DE RESULTADOS/ESTUDIO	42 HORAS

DÍA = 6 HORAS

TOTAL = 306 HORAS

Figura 14 - Diagrama de Gantt de las horas invertidas al proyecto.

5. Desarrollo

En este apartado de desarrollo se va a tratar las diferentes secciones que han compuesto la implementación del proyecto, desde la elección del videojuego para aplicar la sensación de juego hasta su desarrollo completo con las métricas aplicadas.

5.1. Tecnologías y herramientas

Las tecnologías utilizadas para el desarrollo han sido el motor Unity 3D que permite crear un videojuego desde cero, y Visual Studio Code como editor de código.

Unity 3D se lanzó al mercado en el año 2005 y desde entonces no ha parado de crecer tanto en comunidad como en desarrollo. Cada año sigue siendo actualizado y es uno de los motores de videojuegos más utilizados en el mundo, contando con una comunidad y soporte enorme y un plan gratuito que permite la publicación de videojuegos con su motor sin pagar por la licencia mientras los beneficios sean menores a 100.000 dólares al año.

Visual Studio Code es un IDE personalizable por extensiones que puede utilizarse a la par con Unity y se puede usar para programar en C#. Gracias a la gran customización que tiene el editor se ha elegido herramienta de código predeterminada en vez de Visual Studio Community.

5.2. Decisiones y elección de la demo para aplicar el Game Feel

En las primeras reuniones se concretó que el experimento sobre el Game Feel se realizase sobre algún juego en el que se dispusiese acceso al código, ya fuese un proyecto de algún año anterior, algún juego de un compañero que estuviese dispuesto a cederlo, o un proyecto personal que fuese sencillo de modificar.

Tras analizar los objetivos del proyecto y cambiar la visión de cómo realizar estas pruebas, se decidió que finalmente se haría un pequeño juego desde cero o partiendo de una plantilla que permitiese modificar el juego lo suficiente como para poder aplicar las métricas que se iban a utilizar para el estudio.

Como elección del sistema de juego de esta demo se pensó en crear un pequeño juego de plataformas en 2D que aplicase las métricas de respuesta y analizase la sensación de los jugadores conforme se alteraban las respuestas de sus acciones.

El juego debía poder ser ejecutado en móviles para poder así realizar más fácilmente un estudio de caso en el que varios usuarios probasen el juego y diesen sus opiniones en un cuestionario.

Unity dispone de un compilador para Android que permite portear un juego hecho para PC a una plataforma móvil. Junto a esto, dispone también de diversas plantillas que permiten a los usuarios crear un nuevo proyecto partiendo de una base establecida. En este caso, Unity dispone de una que permite generar un entorno de desarrollo para un juego de plataformas en 2D como se puede observar en la Figura 15, y por lo tanto reforzó la decisión de crear la demo basándose en este planteamiento.

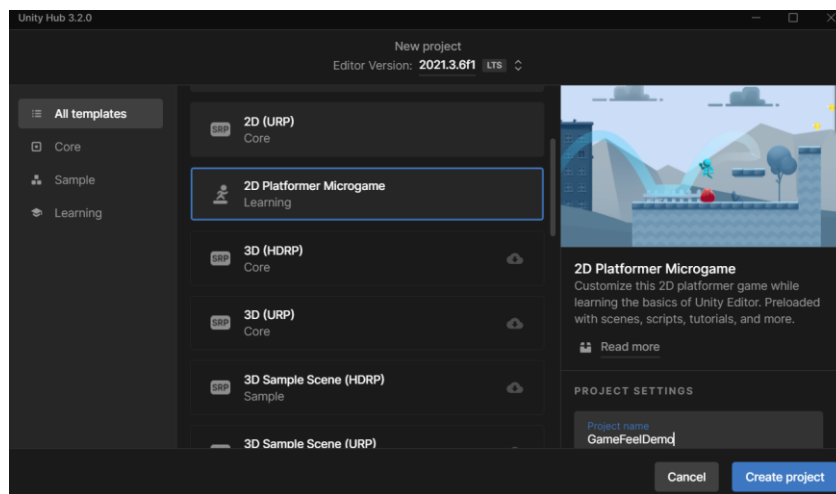


Figura 15 - Plantillas para la creación de proyectos al abrir Unity Hub de Unity 3D.

Una vez elegida la plantilla, cuando se crea el proyecto, se despliega la vista del editor de Unity, vista en la Figura 16, que permite al usuario comenzar a editar los componentes, scripts, objetos de la escena, etc.



Figura 16 - Editor de Unity con la plantilla de Minijuego en 2D.



Ya una vez dispuesto el entorno de desarrollo, se debía plantear qué se quería desarrollar y medir con la demo, cuál era su propósito, etc.

Para empezar, se debía definir cómo iba a ser el juego. Ya que se planteó un videojuego de plataformas en 2D, este tipo de juegos disponen de varios niveles, como, por ejemplo, Super Mario Bros que dispone de 8 mundos y 4 niveles por cada mundo [2] por lo que sería buena idea implementar varios niveles donde aplicar el apartado de la sensación de juego.

La idea que se tenía en mente en un principio era modificar un juego para crear una mejor sensación de juego, pero debido a que es algo que ya se hizo en el proyecto [8] se optó por crear una experiencia distinta. La modificación del juego resultaría en un experimento donde se sometería al jugador a través de distintas pruebas donde se iría cambiando el movimiento del personaje de modo que la sensación de juego cambiase, y por lo tanto crease experiencias diferentes para el jugador, pudiendo así al final obtener feedback por parte de los sujetos de prueba y en base a sus sensaciones definir cuáles serían los parámetros aproximados con los que se obtendría el mejor Game Feel.

5.3. Objetivos y pautas del desarrollo

El objetivo de esta demo es crear diversas sensaciones para el jugador y que con las métricas aplicadas se obtenga una solución sobre cuáles son las modificaciones que crean una mejor experiencia. Para lograr ese cambio de sensaciones se va a centrar el desarrollo de la demo en el apartado del movimiento, por lo que las métricas de respuesta y los parámetros relevantes a las funciones de movimiento van a ser el foco del proyecto.

Dicho esto, se comienza definiendo algunas tareas que conforman el comienzo del desarrollo, tal y como se puede apreciar en la Figura 17.

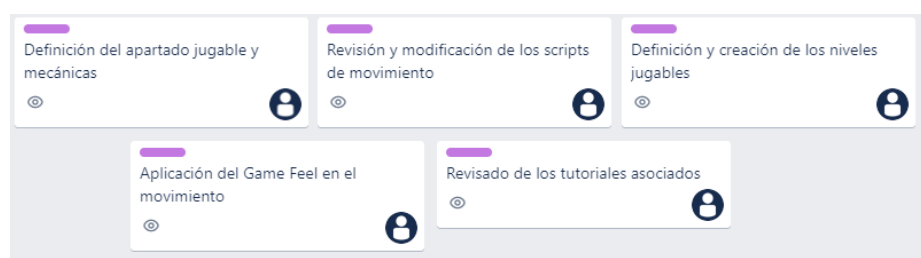


Figura 17 - Tareas asociadas al desarrollo de la Demo.

Conforme se fuese desarrollando el juego la idea es que se vayan viendo los tutoriales asociados, ya sea para aplicar una capa de color a los escenarios, añadir nuevos niveles o retocar efectos de pulido.

La primera parte es definir las mecánicas que va a tener el juego y como se va a adaptar el esqueleto de la Demo al producto final. Ya que la métrica principal de aplicación del Game Feel va a ser la respuesta, se puede plantear un juego en el que sus mecánicas tengan un efecto en base a lo que se define en el apartado de métricas de respuesta.

En el apartado 2.3.2 se hablaba sobre las gráficas de ADSR y ASR donde se podía ver representado el tipo de movimiento de un personaje, con el que se podía medir los tramos de aceleración o velocidad del jugador en base a la entrada de la información, y en el apartado 2.3.1 se comentaba la importancia de la respuesta en la interacción del jugador con el videojuego.

La idea es que estos dos conceptos se complementen. La respuesta tiene que ser adecuada y debe crearse un flujo de feedback continuo entre el jugador y el videojuego, y para ello debe haber una correcta sensación de juego.

El punto que se quiere demostrar con esta demo es la influencia que tiene el control y la respuesta en el apartado jugable y cómo influye en el disfrute del juego. Para ello se variarán los parámetros que afectan al movimiento del personaje jugable y los tiempos de respuesta al movimiento de las acciones del jugador.

Una vez adaptadas las mecánicas y el código, se realizarán varios niveles donde se explore el cambio de estos parámetros para crear distintas sensaciones en los jugadores, pudiendo así medir el grado de satisfacción y disponer de distintos checkpoints donde realizar las mediciones. Los niveles servirán como entretenimiento para el jugador donde pueda explorar otros desafíos en el juego y además como punto de referencia a la hora de analizar la sensación de juego.

Cuando se complete el desarrollo del juego se creará una versión para Android para así facilitar la obtención de feedback sobre el juego y que los usuarios más casuales puedan probar rápidamente la demo sin necesidad de utilizar un ordenador.

Finalmente, una vez se obtengan suficientes respuestas en los formularios sobre la sensación de juego, se plasmarán los resultados en el apartado 7 y se analizarán para poder entender cuál es la mejor sensación de juego posible.

5.4. Implementación de los cambios a las mecánicas

El juego dispone de elementos básicos como un mapa, enemigos, coleccionables y un personaje jugable con sus scripts de control movimiento como se puede apreciar en la Figura 18.

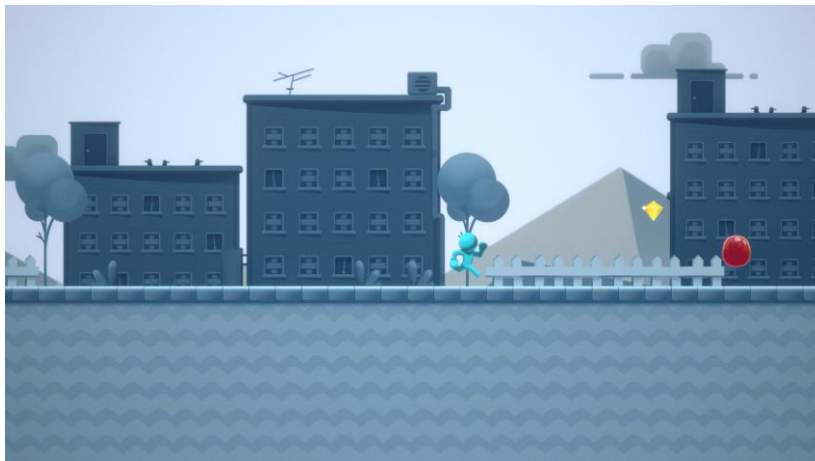


Figura 18 - Estado inicial del juego tras crear la plantilla.

Tener estos elementos fundamentales permite que se pueda comenzar con unos fundamentos sólidos sobre los que trabajar el aspecto del apartado jugable sin necesidad de preocuparse de crear un juego entero desde cero. Con esto, podemos encontrar una variedad de carpetas scripts en la Figura 19 que representan la estructura del proyecto en cuanto a código se refiere.

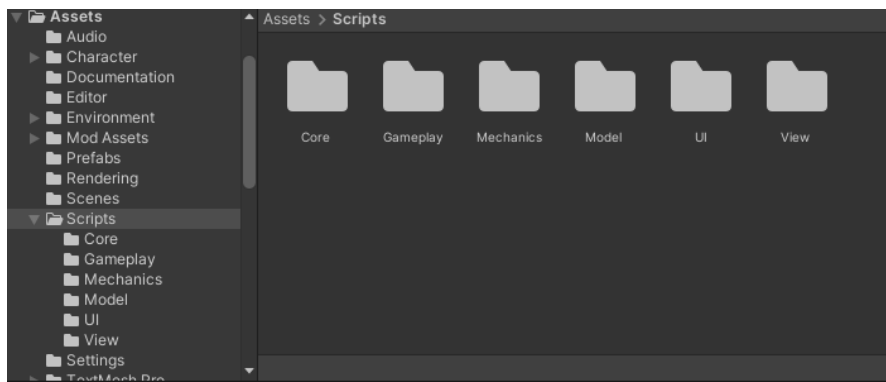


Figura 19 - Estructura del proyecto y código.

La parte relevante a la hora de cambiar el funcionamiento del personaje y la respuesta del juego se encuentra en el apartado de "Mechanics" donde se puede apreciar en la Figura 18 que existen varios controladores como el AnimationController, PlayerController, GameController o EnemyController. Concretamente, el PlayerController es de los más importantes de los que van a ser utilizados para modificar el juego ya que corresponde al apartado que implica el movimiento del jugador, y, por ende, refleja la respuesta que va a ver el jugador cuando comience a controlar su personaje.

Antes de comenzar a modificar los scripts es bueno revisar los tutoriales que ofrece el proyecto, entre ellos, el cambio de colores de los sprites, lo cual dará una mejor impresión para el jugador y dará una sensación de pulido en el juego. También hay que entender cómo funciona el objeto del personaje jugable. Este dispone de un Rigidbody que permite al objeto "actuar bajo el control de la física... recibir fuerza y torque..." [9] por lo que el movimiento del objeto se verá influenciado por las físicas simuladas por el juego. En la Figura 20 se puede ver el nuevo aspecto del juego y los componentes que forman el objeto del personaje jugable, entre ellos el Rigidbody y el PlayerController. Concretamente este último dispone de una variedad de parámetros modificables en el editor, donde se pueden apreciar parámetros que modifican principalmente al movimiento como un modificador de la gravedad, velocidad máxima, velocidad a la que se despega del suelo, por lo tanto, es interesante comenzar con la modificación de este.

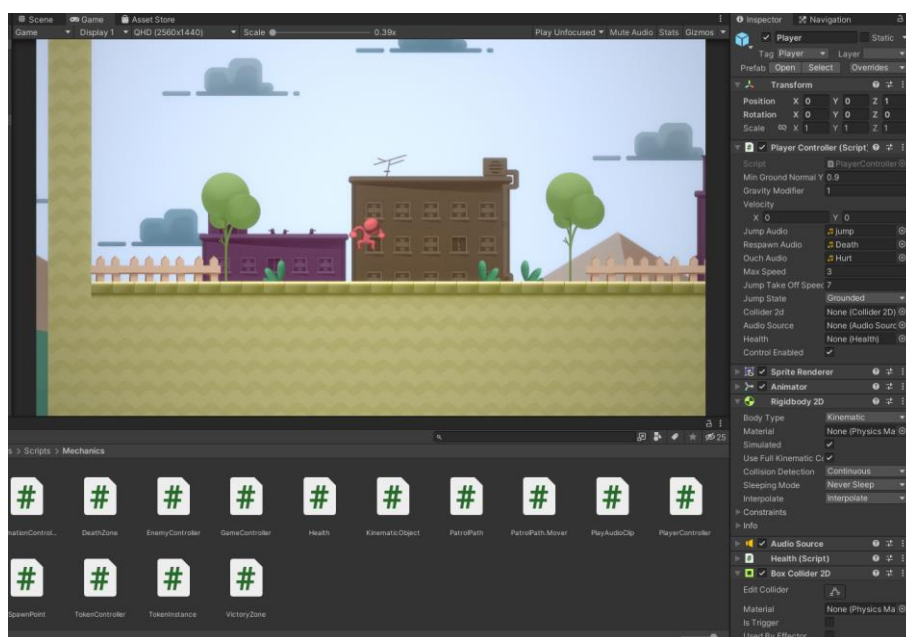


Figura 20 - Componentes del personaje jugable.



En el script del PlayerController se encuentran las funciones básicas de Unity como Awake() y Update() las cuales forman parte del ciclo de ejecución del juego [10]. En la primera se declaran aspectos relacionados con los componentes del objeto del jugador como pueden ser las fuentes de sonido, las cajas de colisiones, animaciones, etc. Y en la segunda se define lo que va a ser el flujo de juego que conlleva todo el apartado del movimiento del personaje en cada frame.

```
protected override void Update()
{
    if (controlEnabled)
    {
        move.x = Input.GetAxis("Horizontal");
        if (jumpState == JumpState.Grounded && Input.GetButtonDown("Jump"))
            jumpState = JumpState.PrepareToJump;
        else if (Input.GetButtonUp("Jump"))
        {
            stopJump = true;
            Schedule<PlayerStopJump>().player = this;
        }
    }
    else
    {
        move.x = 0;
    }
    UpdateJumpState();
    base.Update();
}
```

Figura 21 - Función Update() del personaje jugable.

En la Figura 21 se implementa en la función una comprobación de que el jugador puede usar sus inputs y en caso de que estén habilitados se calcula la posición del jugador en base a los controles que se pulsen.

Como el juego es en 2D mientras el jugador se mueva por el escenario y no esté saltando, solo es relevante el eje X. En caso de que el jugador este en el suelo, un estado determinado por la caja de colisión que indique que el objeto está en contacto con otro que dispone de una colisión fija, puede disponerse a saltar, lo cual cambiaria el estado del jugador a PrepareToJump.

Mientras se mantenga pulsado el botón de saltar el jugador aumentara la altura de su salto hasta una altura máxima, y si lo suelta comenzara a decrecer antes de llegar a su límite.

Ya que la idea es tener varios niveles donde cambien los parámetros del movimiento se debería crear un LevelController que permita manejar el nivel en el que se encuentra el jugador y un LevelExit que se les asignará a las puertas de salida para detectar cuando sale el jugador. Esto también permitirá al PlayerController saber en qué nivel está y qué cambios al funcionamiento de su script debe hacer. En la Figura 22 se define el esqueleto de los dos scripts y su funcionamiento básico.

```

public class LevelController : MonoBehaviour
{
    public static LevelController Instance { get; private set; }
    public int currentLevel;

    void Awake(){
        if (Instance != null && Instance != this)
        {
            Destroy(this);
        }
        else
        {
            Instance = this;
        }
    }

    public void UpdateLevel(int newLevel){
        currentLevel = newLevel;
        SceneManager.LoadScene(currentLevel);
    }
}

public class LevelExit : MonoBehaviour
{
    void OnTriggerEnter2D(Collider2D other){
        if(other.CompareTag("Player")){
            //change the level according to the current level.
            LevelController.Instance.UpdateLevel(LevelController.Instance.currentLevel + 1);
        }
    }
}
    
```

Figura 22 - Funcionamiento base del LevelController y LevelExit.

Ahora que ya se dispone de un control del nivel actual del jugador, se puede empezar a modificar las mecánicas de forma acorde al nivel.

Como bien se ha dicho antes, la intención de este estudio es poder aplicar cambios en cada nivel que creen distintas sensaciones en el jugador y así poder medir en que partes del juego ha encontrado una mejor sensación. Para ello se va a modificar el script del PlayerController.

Primero, el LevelController se instancia como un Singleton [11] al que luego accederá el PlayerController para determinar en qué nivel está. Luego el PlayerController ejecutará en su Start() una función llamada SetLevelParameters() que modificará los valores relevantes para el movimiento dependiendo del nivel en el que se encuentre. También en la función donde se calcula la velocidad se añadirán nuevos parámetros que modifiquen la sensación de movimiento.

Respecto a la función de SetLevelFeatures() se asigna una modificación de los valores por cada nivel existente. Por el momento, dado que solo existe un nivel, se comienza a trabajar con las modificaciones pertinentes al primer nivel. Los atributos que se van a modificar para alterar la jugabilidad van a ser *maxSpeed*, *jumpTakeOffSpeed*, *jumpModifier*, *jumpDeceleration* y *timeZeroToMax*. Este último se ha añadido a la formula del cálculo de velocidad para crear una aceleración en la que el jugador sienta que tarda más o menos en llegar al máximo de su velocidad. Con este parámetro se puede simular el movimiento como una gráfica de ASR como las ya vistas en el apartado 2.3.2 de la memoria, ya que así el jugador tendrá distintas sensaciones según el tiempo que tarda en llegar al límite de su velocidad. En la Figura 23 se encuentra la modificación de la función ComputeVelocity() con la que se calcula la velocidad



actual que tiene el personaje. Previamente, el personaje obtenía directamente la velocidad máxima cuando empezaba a moverse y ahora requiere de un tiempo específico para alcanzarla. Ahora, se calcula tomando como referencia la variable *accelRatePerSec* que es igual a la división entre la velocidad máxima y el tiempo requerido para llegar de cero al máximo.

```
if (move.x != 0){
    characterVelocity += accelRatePerSec * Time.deltaTime;
    characterVelocity = Mathf.Min(characterVelocity, maxSpeed);
}
else if (move.x == 0){
    characterVelocity = 0.0f;
}

targetVelocity = move * characterVelocity;
```

Figura 23 - Modificación del cálculo de la targetVelocity del personaje.

Estos parámetros han sido escogidos principalmente porque son parámetros asociados a la velocidad del personaje y a su capacidad de salto, con lo cual su modificación creará sensaciones distintas dependiendo de cuál sea su configuración. La estructura de la función de *SetLevelFeatures()* define en la Figura 24 y será la que se siga hasta el desarrollo final.

```
void SetLevelFeatures(){
    int level = LevelController.instance.currentLevel;

    switch(level){
        case 0: // First level, here should apply a higher max speed and higher jumps.
            // Modify
            // maxSpeed,
            // jumpMod...
        default:
            break;
    }
}
```

Figura 24 - Estructura del código de la función SetLevelFeatures().

5.5. Diseño y definición de los niveles

Una vez terminada la implementación de los cambios a las mecánicas se comenzó con el desarrollo de los niveles jugables. Basándose en enfoque del proyecto y el tipo de usuarios que iban a ser sujeto de estudio se debatió sobre cuantos niveles debía tener el juego. Antes de consensuar la idea de tener distintos niveles se pensó en tener un único nivel de una larga extensión donde se fuesen realizando triggers que cambiasen la dinámica del movimiento del juego y el jugador notase el cambio instantáneamente una vez pasase por esa zona. Esta idea se descartó ya que le podría resultar confusa para los jugadores a la hora de rellenar las

encuestas del estudio, ya que habría que delimitar los puntos como “checkpoints” en los que cambia la sensación de juego, y para hacerlo más claro, simplemente se podrían crear nuevos niveles donde se adaptase la sensación y luego realizar preguntas sobre cada uno de forma individual. Por lo tanto, se decidió pasar a crear varios niveles que implementaran los cambios a la jugabilidad.

Respecto a cuantos niveles se debían crear, esto dependía de la duración que debía llevarle a los jugadores para terminar la demo. Se estimó que para un jugador que no tenga mucha experiencia con juegos de plataformas, la demo debería durar entre unos 10 minutos con unos desafíos lo suficientemente sencillos como para poder completarla sin dejarla a medias. La dificultad vendría dada por la sensación de juego y los cambios al movimiento, por lo que los niveles no debían contener un plataformeo complejo o puntos demasiado críticos que requieran mucha habilidad.

Con esto en mente, se decidió extender el juego a tres niveles donde el jugador tuviese que alcanzar una meta que le movería al siguiente nivel, un número más que suficiente para realizar las pruebas con los cambios aplicados a la jugabilidad y que permitan obtener unos resultados significativos del estudio. Los parámetros de los niveles se definirían en base a su dificultad para completarlos. En un nivel más fácil se cambiaría para que el movimiento de una respuesta más desequilibrada mientras que en un nivel más complejo la respuesta debería ser más correcta para que el jugador pueda completarlo.

A través de múltiples pruebas internas se estableció un rango de valores que deberían tener los parámetros para que las pruebas no perjudicasen excesivamente la experiencia de los jugadores a la hora de realizar el estudio:

- maxSpeed – Entre 4.0 y 12.0.
- jumpTakeOffSpeed – Entre 3.0 y 8.0
- jumpModifier – Entre 0.7 y 1.2.
- jumpDeceleration – Entre 0.7 y 2.0.
- timeZeroToMax – Entre 0.5 y 1.75.

Los valores asignados a cada nivel en el código se explicarán definitivamente en el apartado de Resultados de la memoria donde se especificará cada valor con los resultados que se han obtenido del estudio.

Los niveles que han sido diseñados para esta demo se han creado teniendo en mente que la jugabilidad debe ser sencilla y corta por lo que se ha optado por hacer niveles cortos que no tengan muchas dificultades para pasarlos.

Además, el juego no disponía de un sistema de checkpoints así que se ha creado un sistema sencillo que cuando el jugador falle en el intento de completar el nivel, ya sea por caer al vacío o por chocar con un enemigo, le devuelva a un punto avanzado del nivel en vez de tener que volver a empezar desde el principio ya que si no la experiencia se podría volver muy frustrante al fallar al final del nivel y ver que tendría que volver a comenzar.

Cuando el jugador pase por el BoxCollider2D [12] del checkpoint, el punto de aparición se moverá a las coordenadas de ese checkpoint de modo que cuando el jugador muera o se caiga al vacío, reaparecerá en las nuevas coordenadas. Esto se ha implementado de forma sencilla, moviendo el vector de posición del punto de reaparición al punto en el que se encuentra el objeto del checkpoint. Tal y como se puede apreciar en la Figura 25, la caja que comprende la colisión con el jugador es de suficiente altura para que si el jugador pasa saltando por esa zona el checkpoint se active igualmente y no tenga que volver a empezar desde el principio.

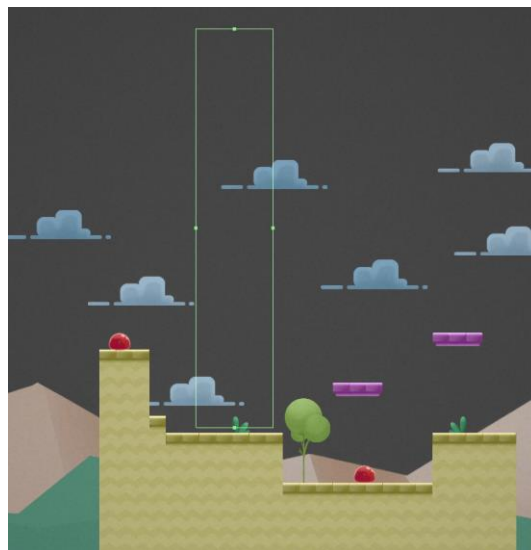


Figura 25 - Checkpoint del nivel 1.

Una vez terminado el sistema de checkpoints, se puede hablar sobre los niveles que se han creado definitivamente para el juego. En la Figura 26 se encuentra el primer nivel, uno sencillo en el que el jugador tiene que pasar a través de unos enemigos y saltar unas pocas



plataformas. La salida del nivel se determina por un cristal rojo que se encuentra al final y una vez entre en contacto con el jugador se le lleva al segundo nivel.

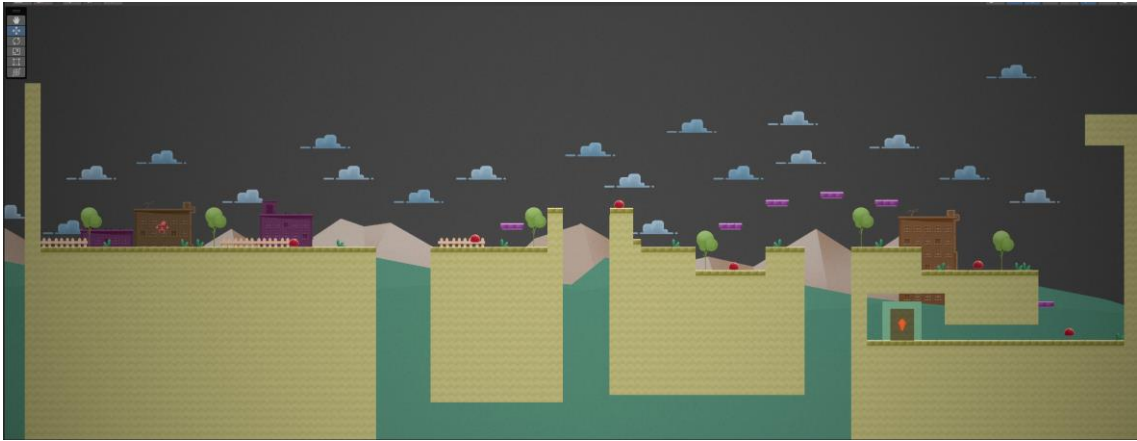


Figura 26 - Primer nivel de la demo.

El segundo nivel, visto en la Figura 27, simula un nivel subterráneo donde hay más un plataformeo más sencillo, pero con más movimiento de subidas y bajadas. El cambio del diseño respecto al anterior se debe a que visualmente da un aspecto diferente al juego y le genera nuevos estímulos al jugador con nuevas formas de moverse por el mapa aparte de saltar unas pocas plataformas.

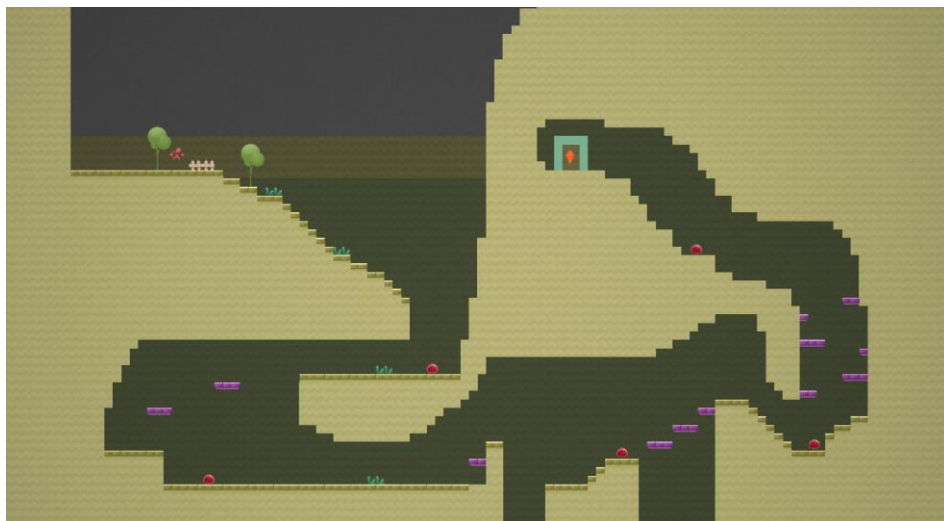


Figura 27 - Segundo nivel de la demo.

Finalmente, en el tercer nivel, representado en la Figura 28, se desarrolla un nivel más complejo que el resto, con más plataformas y menos enemigos. Tiene menos puntos críticos

donde el jugador puede fallar, pero pone más a prueba la habilidad de movimiento del jugador que los anteriores.



Figura 28 - Tercer nivel de la demo.

5.6. Interfaz y build para móvil

Para facilitar la fase de pruebas del estudio, se planteó crear una build para móvil, concretamente Android, ya que es el sistema operativo móvil más utilizado (y el SO más usado mundialmente en general) [13] y sería más sencillo obtener usuarios que pudiesen probar la aplicación. Según StatCounter, el sistema de Android superó en uso a Windows en 2017, un sistema operativo diseñado para ordenadores, y de forma minoritaria, teléfonos y tabletas como se puede ver en la Figura 29.

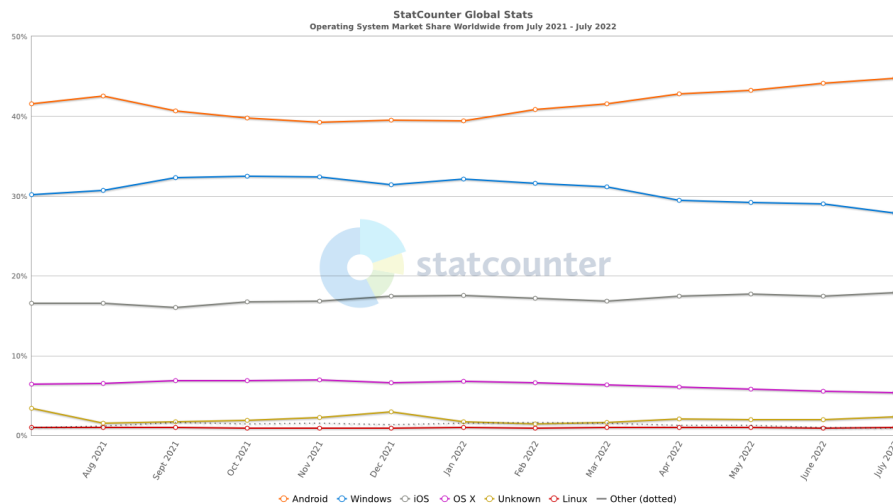


Figura 29 - Comparativa de los sistemas operativos más usados.
Fuente: <https://gs.statcounter.com/os-market-share>

Por tanto, en base a estos datos, y pudiendo realizar un *port* [14] a la plataforma de Android desde Unity, se consensuó que era una buena oportunidad para tener el juego disponible a más usuarios y facilitar las pruebas.



El juego se podía jugar de forma funcional en un ordenador, pero para realizar las pruebas en dispositivos móviles se necesitaba implementar otro sistema de control, ya que en los dispositivos móviles con pantalla táctil no se dispone de botones físicos que simulen las teclas de un teclado de ordenador, por lo que los inputs tendrían que realizarse de forma táctil.

Para lograr esto, se puede disponer en la pantalla de unos botones táctiles que representen los mismos botones con los que se mueve el jugador. Ya que los controles son sencillos, solo se necesitarían botones de movimiento de izquierda y derecha, y un botón de salto.

Esto se puede conseguir de forma sencilla modificando la forma en la que se calcula el movimiento, sustituyendo las variables de entrada de los inputs por las pulsaciones de los botones táctiles. Para ello, en el apartado del código del jugador se ha realizado un control con valores booleanos que evalúan cuándo los botones estaban siendo pulsados a través de unos eventos. En este caso en el PlayerController se le han añadido las variables *moveLeft*, *moveRight*, *jumpPressed* y *jumpReleased*, y se ha cambiado la función de Update() como se puede ver en la Figura 30.

```
if (Application.platform == RuntimePlatform.Android){
    if(moveLeft){
        move.x = -1.0f;
    }
    else if(moveRight){
        move.x = 1.0f;
    }
    else{
        move.x = 0.0f;
    }

    if (jumpState == JumpState.Grounded && jumpPressed)
        jumpState = JumpState.PrepareToJump;

    else if (jumpReleased && jumpPressed == false)
    {
        jumpReleased = false;
        stopJump = true;
        Schedule<PlayerStopJump>().player = this;
    }
}
else {
    move.x = Input.GetAxis("Horizontal");

    if (jumpState == JumpState.Grounded && Input.GetButtonDown("Jump"))
        jumpState = JumpState.PrepareToJump;

    else if (Input.GetButtonUp("Jump"))
    {
        stopJump = true;
        Schedule<PlayerStopJump>().player = this;
    }
}
```

Figura 30 - Comprobación de la plataforma y cambios a los inputs de movimiento.

Para poder moverse con los inputs táctiles y que estos se muestren, el juego se tiene que ejecutar en la plataforma de Android, mientras que en cualquier otra plataforma dispondría de los controles físicos como hasta ahora. No se ha tomado en consideración ninguna otra

plataforma en este desarrollo, pero a futuro, se debería cambiar esta implementación considerando otras plataformas como iOS. Luego, para representar los botones táctiles en el juego, se han creado unas figuras de botones desde cero con el programa Adobe Photoshop, los cuales se pueden ver en la Figura 31.

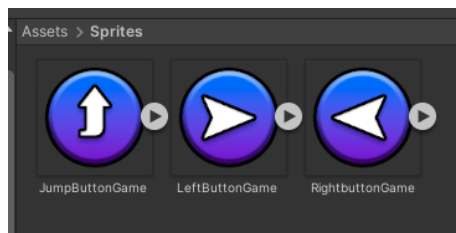


Figura 31 - Botones táctiles del juego.

Para añadir estos botones se tendría que crear un Canvas [15] que implementara unos botones con un código en su interior. Este código contendría modificaciones a los valores booleanos que se han nombrado anteriormente en el código del PlayerController, de modo que cuando uno de estos botones se presionara, cambiara el valor booleano de la variable y efectuase el movimiento en el personaje jugable. En la Figura 32 se encuentra el código utilizado para la funcionalidad de los botones.

```
public class ButtonClick : MonoBehaviour, IPointerDownHandler, IPointerUpHandler
{
    public PlayerController player;

    public void OnPointerDown(PointerEventData eventData)
    {
        if(gameObject.CompareTag("LeftButton")){
            Debug.Log("LEFT PRESSED");
            player.ButtonPress(0, true);
        }
        else if(gameObject.CompareTag("RightButton")){
            player.ButtonPress(1, true);
        }
        else if(gameObject.CompareTag("JumpButton")){
            player.JumpPress(true, false);
        }
    }

    public void OnPointerUp(PointerEventData eventData)
    {
        if(gameObject.CompareTag("LeftButton")){
            Debug.Log("LEFT RELEASED");
            player.ButtonPress(0, false);
        }
        else if(gameObject.CompareTag("RightButton")){
            player.ButtonPress(1, false);
        }
        else if(gameObject.CompareTag("JumpButton")){
            Debug.Log("JUMP RELEASED");
            player.JumpPress(false, true);
        }
    }
}
```

Figura 32 - Función de los botones del Canvas para control táctil.

Finalmente, una vez dispuesto el Canvas y los controles táctiles en el juego, la interfaz se vería tal y como se muestra en la Figura 33.



Figura 33 – Interfaz para dispositivos móviles de la demo.

Con esto, se creó la build para Android desde la pestaña de "Build Settings" de Unity3D y se comprobó la funcionalidad en un teléfono personal. Tras conseguir unos resultados satisfactorios, se decidió que la fase de desarrollo se daría por concluida y que se podría pasar a la fase del estudio con usuarios reales.

6. Estudio económico

En este apartado se hablará del coste que supondría realizar el proyecto de forma profesional, contando con la participación de una empresa que disponga de fondos para un empleado, equipo utilizado, etc.

Primero, los costes humanos suponen la parte más relevante del coste en cuanto al estudio económico. Contando las horas trabajadas de la tabla en la Figura 14, encontramos esta información:

- 192 horas trabajadas en la documentación sobre el concepto y la escritura de la memoria.
- 72 horas trabajadas en el desarrollo de la demo.
- 42 horas dedicadas a la concepción y análisis del estudio realizado.

Ya que el proyecto ha sido realizado de forma independiente, se va a tomar en cuenta lo que supondría el sueldo de un trabajador junior como desarrollador de videojuegos sacando datos en la página web de Glassdoor [16].

Con esto se puede obtener una estimación de los gastos que se necesitarían para pagar el proyecto en base a las horas trabajadas. Para tomar las horas trabajadas se ha tomado 1800 horas de referencia según una aproximación de las horas anuales de trabajo. Se excluyen las horas invertidas en el resultado del estudio y su creación, solo se toma en cuenta las de documentación y desarrollo, por lo que solo se tomara las horas dedicadas al inicio del proyecto en la fase de documentación/memoria y el resto de las del desarrollo.

Para el sueldo anual, se aplica el 30% de la Seguridad Social. El sueldo base de los programadores junior, tomando como referencia Mercury Steam [17], es entre 17.758 y 27.254 euros al año. Poniendo que se entra como programador junior iniciado, el sueldo sería de 17.758 euros brutos al año y aplicando el incremento de la SS, a la empresa le costaría 23.085 euros. Por lo tanto, en base a la paga por hora que recibe el programador de 9,86 euros, si se separa el tiempo total invertido para el desarrollo e investigación en unas 162 horas, la paga total que recibiría el programador es de 1597,32 euros por el proyecto, mientras que la empresa tendría que pagar 2076,51 euros. En la Tabla 1 se reflejan los costes para cada entidad.

PUESTO DE TRABAJO	SUELDO ANUAL	COSTE POR HORA	COSTE DEL PROYECTO	COSTE A LA EMPRESA
Program. Junior	17.758 €	9,86 €	1597,32 €	2076,51 €

Tabla 1 – Coste total del proyecto con un Programador Junior.

Respecto al coste del hardware empleado, se ha necesitado un portátil para trabajar en el proyecto junto a los periféricos correspondientes. Debido a que se han realizado videollamadas para las reuniones he decidido incluir también los gastos de unos cascos y una cámara web. Estos costes están reflejados en la Tabla 2.

HARDWARE	INVERSIÓN
Ordenador Portátil Asus ROG Strix G15	1199,99 €
Ratón Logitech G203	27,99 €
Webcam Microsoft LifeCam HD 3000	24,99 €
Cascos Logitech G432	39,98 €
COSTE TOTAL	1292,95 €

Tabla 2 – Coste del hardware personal para el proyecto.

Si se analizan los modelos de negocio posibles con el desarrollo de este proyecto, se puede crear un hipotético caso en el que el proyecto fuese un juego original en el que los usuarios accediesen a pagar un precio por él y crease rentabilidad para la empresa.

Para empezar, el juego debería tomar un extenso desarrollo en el que se ampliasen la cantidad de niveles, mecánicas y modos que tendría el juego. Poniendo que el coste del juego serían el triple o el cuádruple hasta alcanzar un estándar decente para un juego que pueda salir a la venta, el coste del juego aumentaría hasta los 8.306 euros sin contar todo lo que pueda costar el marketing del juego.

Partiendo de la base que se dispone, la idea del juego surge de crear diferentes experiencias cambiando la sensación de juego en diversos escenarios. Hay juegos que se aprovechan de un público de jugadores que aprecian juegos con una dificultad a veces injusta y que provocan reacciones que luego pueden trasladar a plataformas de video como podría ser Youtube o Twitch donde ahí reciben millones de visitas.

En ejemplos de este tipo de juego se pueden encontrar Syobon Action (o Cat Mario en occidente) [18], un juego que mantiene similitudes con Super Mario Bros, pero añade elementos que dificultan el progreso del jugador constantemente, como plataformas que se caen donde parece haber suelo corriente, objetos que matan instantáneamente al jugador, etc. O ALTF4 [19], un juego de plataformas en 3D donde el jugador debe correr por un mapa lleno de trampas injustas que requieren al jugador de realizar múltiples intentos hasta poder completarlo.

En el caso de ALTF4, el juego se vende por 2.50€ en la plataforma de Steam y dispone de más de 7500 reseñas, las cuales tienen que haberse realizado tras comprar el juego. Se puede decir que el juego ha sido un éxito apelando a este nicho de jugadores que disfrutan de este tipo de experiencias.

El principal modelo de negocio sería desarrollar en extensión el juego y lanzarlo a una tienda digital como podría ser Steam o Google Play. Suponiendo que el juego se acaba lanzando en estas plataformas, ambas disponen de un corte de beneficios en el que de cada venta se atribuyen un porcentaje de lo pagado por el juego.

Si se dispusiera el juego completo en la plataforma de Steam, hay que contar con que cada venta del juego va a suponer un 70% de los beneficios para la empresa. En el caso de que no recibiese más soporte en forma de actualizaciones, a parte de los posibles arreglos de errores, el juego se mantendría con el contenido base final, y no crearía gastos recurrentes para la empresa.

Tomando esto en cuenta, y viendo el modelo de negocio de otros juegos similares, si el juego tuviese un valor de 3.99€ en la tienda, se puede observar en la Tabla 3 que las unidades necesarias que tendría que vender el juego para comenzar a ser rentable para la empresa ascenderían hasta las 2.978 unidades.

COSTE DEL JUEGO	COSTE EN LA TIENDA	BENEFICIOS DE STEAM	BENEFICIOS EMPRESA	UNIDADES NECESARIAS
8.306€	3,99€	1,19€	2,79€	2.978

Tabla 3 – Unidades necesarias en Steam para la rentabilidad del juego.

Y en caso de Google Play, en el primer millón de dólares recibido por las ventas del juego, el corte de beneficios es de un 15% por lo que cada venta del juego supondría un 85% de beneficios para la empresa. Poniendo que el precio del juego se mantiene en 3.99€, se necesitarían de un total de 2.443 unidades vendidas para rentabilizarlo.

COSTE DEL JUEGO	COSTE EN LA TIENDA	BENEFICIOS DE GOOGLE	BENEFICIOS EMPRESA	UNIDADES NECESARIAS
8.306€	3,99€	0,60€	3,40€	2.443

Tabla 4 – Unidades necesarias en Google Play para la rentabilidad del juego.

Con el juego dispuesto en ambas plataformas sería más sencillo llegar a la meta de unidades necesarias, contando además que si se hacen descuentos los jugadores tienden a comprar más cuando hay rebajas en las plataformas, por lo que con un buen descuento las ventas del juego podrían dispararse.

7. Resultados

En este apartado se incluye el análisis del estudio para el proyecto, bajo que parámetros se realizó, cuáles eran los objetivos, los resultados que han sido obtenidos del playtesting de los jugadores y qué significan para la creación de una buena sensación de juego.

7.1. Descripción del estudio

En anteriores apartados se ha hablado sobre cuál era el objetivo que se tenía con este estudio. La idea principal era someter a los jugadores a distintas experiencias y así poder ajustar cual sería la mejor sensación de juego en base a los parámetros dados a cada experiencia y las sensaciones que han tenido los jugadores.

Para entender esto, se ha decidido hacer objeto de estudio las dimensiones en las que se aplican las modificaciones de las mecánicas. En el apartado de desarrollo se hablaba sobre los valores que iban a ser modificados en cada nivel para determinar una experiencia. Para tener una visión más clara, se ha realizado una tabla que se puede observar en la Figura 34, donde se definen las dimensiones del estudio, en este caso, la velocidad del personaje y el salto.

DIMENSIONES QUE MEDIR	VELOCIDAD	SALTO
maxSpeed	X	
jumpTakeOffSpeed		X
jumpModifier		X
jumpDeceleration		X
timeZeroToMax	X	

Figura 34 - Tabla de dimensiones a medir (velocidad y salto)

La velocidad va a ser afectada por los parámetros de *maxSpeed* y *timeZeroToMax*. El primero determina cual va a ser la velocidad máxima que puede alcanzar el personaje, mientras que el segundo, marca el tiempo en segundos que debe mantenerse el personaje en movimiento para alcanzar la velocidad máxima.

Respecto al salto, se ve afectada por tres parámetros distintos que son *jumpTakeOffSpeed*, *jumpModifier* y *jumpDeceleration*. El primero es el impulso inicial que tiene el jugador y que afecta a cuál va a ser su altura máxima; el segundo modifica como multiplicador la capacidad de salto del personaje; y finalmente el tercero afecta a la caída del personaje, haciéndola casi instantánea cuando se deja de pulsar el botón de salto o haciendo una caída con un arco más definido.

Estas dimensiones han sido escogidas debido a los parámetros que las afectan y como se pueden medir. Para este estudio si se quisiese crear un modelo estadístico sería demasiado complejo, ya que se trabajan con 5 variables diferentes que afectan a dos dimensiones distintas, por ello para que entre dentro del alcance del estudio ha resultado más interesante abordar más parámetros que permiten modificar la sensación y dedicarlo más a observar el valor descriptivo de los parámetros y su influencia, ya que por ejemplo si solo se modificase el valor multiplicador del salto, se perdería el jugo que tiene modificar la deceleración o el impulso que tiene al inicio y quedaría un estudio muy sencillo.

En el estado del arte se ha hablado sobre las métricas de respuesta y como estas se pueden representar en una gráfica ASR. De igual forma, los parámetros elegidos para estas dimensiones permiten representar esa dimensión como una gráfica ASR, ya que para ambas dimensiones se tiene una fase de *Attack* en la que el botón es pulsado y el valor empieza a incrementar, una fase *Sustain* donde el valor alcanza su punto más álgido, y una fase *Release* donde el botón se libera y empieza a decaer.

Para cada una de las dimensiones se diseña una gráfica ASR en base a la simulación que se haría al pulsar el input determinado para cada dimensión. Los datos de las gráficas no son acertados, pero intentan ser similares a los que se obtendrían en una prueba real. Estos datos se ven reflejados en las gráficas de las Figuras 35 y 36.

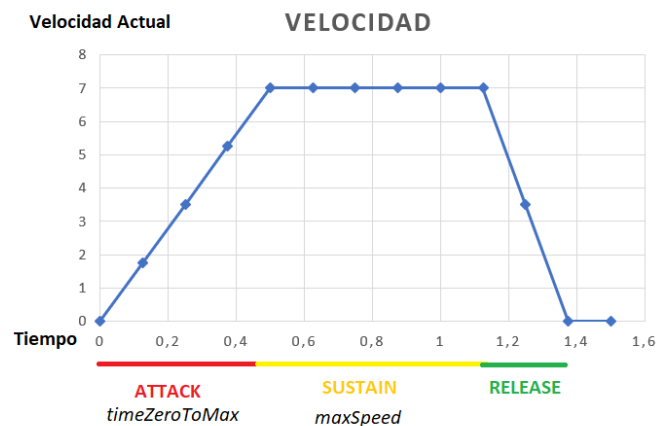


Figura 35 - Gráfica ASR de la velocidad del personaje a lo largo del tiempo.

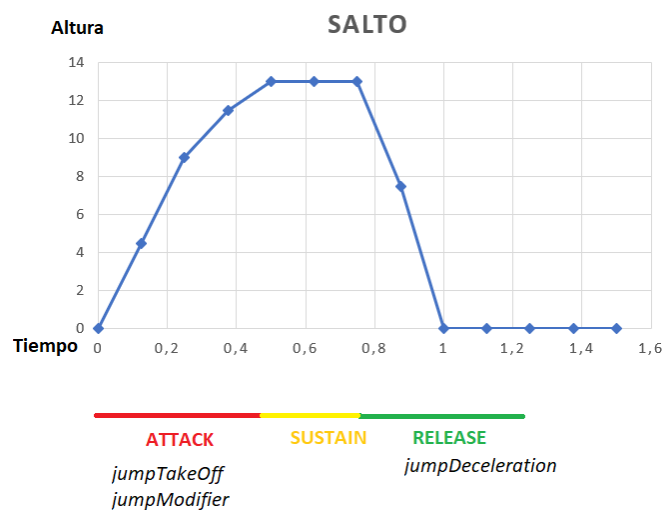


Figura 36 - Gráfica ASR de la altura del personaje en un salto a lo largo del tiempo.

Ambas gráficas disponen de curvas muy distintas, la velocidad es más lineal mientras que el salto adquiere una curva algo más pronunciada con una caída en picado. Al final, lo que modifican los parámetros del juego es, por una parte, el *Attack* y *Sustain* de la Velocidad y por la otra, el *Attack* y el *Release* del Salto.

Teniendo definido como se miden estas dimensiones y en base a qué, lo siguiente es determinar los valores que tienen cada parámetro y decidir cómo van a ser en cada nivel.

Los valores asignados a cada nivel han sido probados en pruebas internas, en el apartado 5.5 del desarrollo se comentaba el rango límite que debían tener los valores para generar una experiencia que fuese adecuada y no rompiera los límites de los niveles o los hiciera injugables.

Se planteo también el hacer que los jugadores jugasen el orden de los niveles aleatoriamente con los mismos valores asignados pero los parámetros están fijados al diseño de cada nivel para que este pueda completarse, poniendo de ejemplo que en un nivel donde las plataformas estén más altas, si se cogiese la configuración del nivel 2, no se podría completar.

En la Figura 37 se encuentra la tabla de los valores asignados a cada variable, creando una configuración específica para cada nivel. De los tres niveles que se han hecho, con estos valores se espera que:

- El nivel 1 disponga de una velocidad bastante rápida y un salto alto.
- El nivel 2 sea todo lo contrario, con una velocidad lenta y un salto corto.
- Y el nivel 3 con unos valores más intermedios con una velocidad y salto más estándar.

PARAMETROS / NIVEL	maxSpeed	jumpTakeOffSpeed	jumpModifier	jumpDeceleration	timeZeroToMax
NIVEL 1 - Rapido y Alto	12	8	1.2	0.25	0.5
NIVEL 2 - Lento y Bajo	3	4	1.5	0.5	1.2
NIVEL 3 - Medio y Medio	7	7	0.9	0.9	0.8

Figura 37 - Tabla de valores asignados a cada variable en cada nivel.

7.2. Diseño de las encuestas

Para realizar este estudio se ha tenido que elaborar una encuesta con una serie de preguntas relacionadas con la sensación que han tenido los jugadores mientras jugaban los niveles de la demo, y para tener una buena representación de cuáles eran sus sensaciones, se ha optado por utilizar una encuesta de escala Likert [20] la cual permite valorar la opinión de un usuario en un rango de valores como, por ejemplo, "Muy en desacuerdo" a "Muy de acuerdo", pudiendo tener un espectro de valores más neutrales que indiquen lo acuerdo o desacuerdo que estes con esa pregunta. En la Figura 38 se ve un ejemplo de lo que sería una escala Likert de 5 opciones o ítems.

Ejemplo de la escala de Likert

Totalmente en desacuerdo <input type="radio"/>	En desacuerdo <input type="radio"/>	Ni de acuerdo, ni en desacuerdo <input type="radio"/>	De acuerdo <input type="radio"/>	Totalmente de acuerdo <input type="radio"/>
---	--	--	-------------------------------------	--

Figura 38 - Ejemplo de escala Likert de 5 opciones. [20]

Por supuesto no se limita a un “acuerdo” y se puede representar con cualquier otro tipo de opinión, y ya que en este proyecto se habla de sensaciones, se ha pensado en que la escala de valores varíe entre “Muy mala” y “Muy buena” refiriéndose así a las sensaciones y experiencias del jugador. En base a esas sensaciones se podrá determinar donde se puede mejorar la sensación de juego en cada parámetro.

Para esta escala Likert se ha elegido hacerla de 5 opciones, aunque sí que hay escalas que llegan a las 7 o 10 opciones, se ha elegido esta debido a que crear muchas opciones puede crear indecisión en el usuario a pesar de que con más opciones se obtiene una mayor precisión. Ya que las preguntas se relacionan con sensaciones, se considera que la escala de 5 opciones es la más correcta para hacer este estudio. Finalmente, esta va a ir representada por: “Muy mala”, “Mala”, “Neutral”, “Buena” y “Muy buena”.

Respecto a las preguntas, hay que conseguir medir las sensaciones que provocan los efectos de la alteración de las variables, en este caso, se va a preguntar sobre la experiencia general en ese nivel, la sensación de aceleración, la velocidad que tenía el personaje, el control del personaje durante el salto, la rapidez con la que saltaba y la sensación de deceleración al caer. Cada una de las preguntas va relacionada con cada una de las variables, siendo estas, *timeZeroToMax*, *maxSpeed*, *jumpModifier*, *jumpTakeOffSpeed* y *jumpDeceleration* respectivamente.

Al comienzo de la encuesta también se preguntaría a parte cual era la plataforma en la que el usuario había probado la demo, ya que a los usuarios se le daría a su disposición la demo para PC y Android, y así poder detectar fallos que puedan ocurrir al realizar las pruebas en dos plataformas, como que por ejemplo los resultados de las encuestas del primer nivel de aquellos que jugasen en móvil fuesen muy favorables mientras que los que jugaron en PC lo valorasen negativamente.

La encuesta se realiza en un Google Forms, una plataforma de Google para crear encuestas de forma sencilla y rápida, permitiendo a los usuarios realizar las encuestas de forma totalmente anónima sin recoger ningún dato personal, y que además proporciona gráficas de los resultados obtenidos, las cuales luego se podrían exportar a la memoria del proyecto.

La estructura de la encuesta se conforma por 5 secciones, siendo la primera una introducción donde se presenta la encuesta, las escalas que representan cada valoración y la pregunta sobre la plataforma. Después, sigue con 3 secciones dedicadas a cada nivel donde se pregunta:

- "La experiencia general que he tenido en este nivel ha sido..."
- "Al comenzar a correr, la sensación de aceleración era..."
- "Mientras corría, la sensación de velocidad de mi personaje era..."
- "Cuando saltaba, la sensación de control de mi personaje era..."
- "Al comienzo del salto, la sensación de rapidez con la que saltaba era..."
- "Cuando soltaba el botón de salto, la sensación de deceleración era..."

Y en la última sección se realiza una pregunta sobre la experiencia que han tenido los usuarios dejando un espacio de respuesta libre. Con esta pregunta se pretende obtener un feedback más directo de los jugadores y que les permita extenderse sobre las sensaciones que han obtenido en cada nivel.

Cabe destacar que la pregunta respecto a la sensación de control del personaje durante el salto afecta a ambas dimensiones, por lo que es una forma de medir la sensación de velocidad del personaje durante el salto. Al saltar, el personaje puede influenciar hacia donde se mueve presionando los botones de movimiento horizontal, y, por lo tanto, afecta a su velocidad en el eje X mientras salta.

Esta pregunta sirve para medir si la velocidad influye demasiado a la hora de saltar, ya que en suelo puede tener un valor elevado y que mientras el personaje corre no haya ningún inconveniente, pero que, al momento de saltar, la velocidad influya tanto que el salto se vuelva totalmente incontrolable y no se cree esa sensación de precisión a la hora de saltar.

Con las encuestas listas y la demo preparada, se pasaría a la fase de testeo donde los usuarios prueban el juego y dan sus opiniones sobre las sensaciones que han tenido.

7.3. Resultados de las encuestas y análisis

Para este estudio se ha visto adecuado encuestar a 10 usuarios ya que no se va a realizar un estudio estadístico y que es una cifra que duplica los usuarios aconsejados por Jakob Nielsen para detectar problemas de usabilidad con un producto [21]. Algunos de estos usuarios dispondrían de más conocimiento sobre videojuegos y otros serían usuarios más casuales.

Como se ha dicho antes, la encuesta se ha dividido en cinco secciones con tres secciones para cada nivel.

En la primera sección se analiza la plataforma que han elegido los encuestados para probar la demo jugable. Se encuentra en la Figura 39 que 6 usuarios han elegido Android como plataforma y 4 Windows PC, lo cual prueba que ha sido de utilidad crear una build para Android dado que ha permitido que usuarios que posiblemente no dispusiesen de un PC para probarlo hayan participado en la encuesta.

¿En qué plataforma/s has probado la demo?
10 respuestas

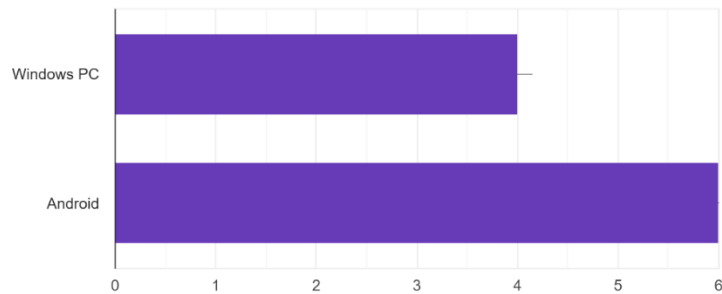


Figura 39 - Plataformas elegidas por los participantes en la encuesta.

7.3.1. Resultados del Nivel 1

En el primer nivel se encuentran unas gráficas ASR con valores aproximados de los que se encontrarían en el primer nivel en la Figura 40.

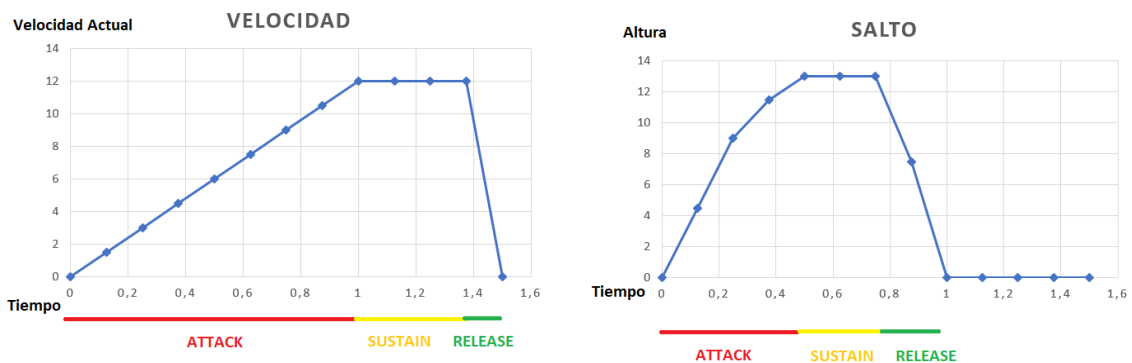


Figura 40 - Representación ASR de la velocidad y el salto del personaje en el Nivel 1.

En la Figura 40 se muestran los resultados obtenidos de las encuestas de la sección del primer nivel.

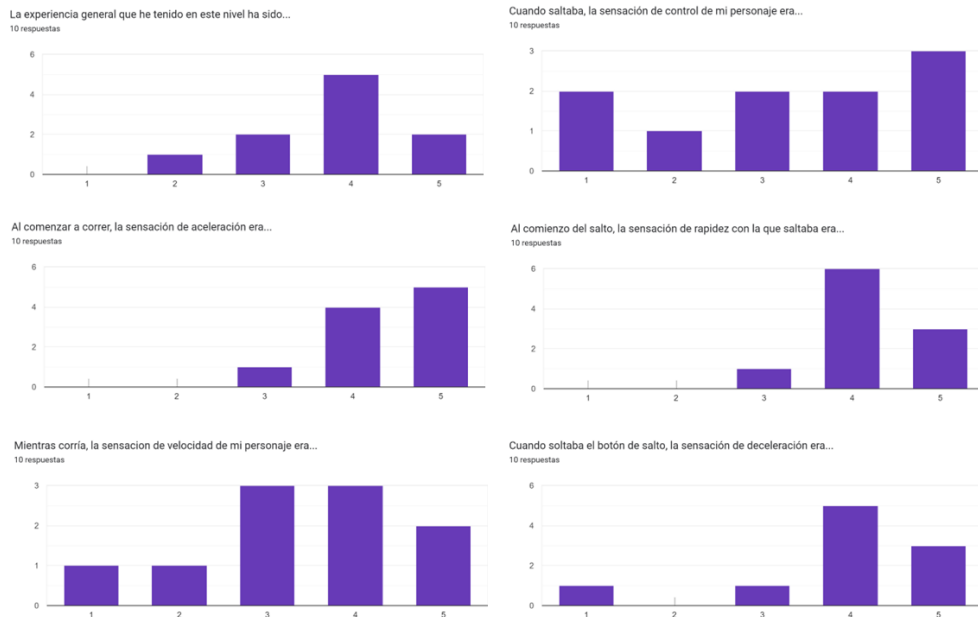


Figura 41 - Resultados del Nivel 1.

Lo primero es la experiencia obtenida en general que se ha obtenido en el nivel, para la mayoría de los usuarios ha sido una experiencia buena, con unos pocos que la han descrito como muy buena y otros pocos como neutral o mala.

En cuanto a la sensación de aceleración, se han obtenido resultados como muy buenos y buenos, con una sola opinión como neutral.

Para la sensación de velocidad del personaje se ha encontrado una disparidad de opiniones, con unas pocas calificándola como muy mala y mala, otras pocas como muy buena, y la gran mayoría en neutral y buena.

Así mismo, para la sensación de control en el salto, se encuentra una mayoría que opina que es muy buena, pero bastantes opiniones cercanas que la evalúan como muy mala, mala, neutral, y buena, por lo que podría considerarse un punto crítico a la hora de evaluarlo.

Respecto a la rapidez del salto y la deceleración, ambas opiniones se encuentran en un punto en el que se denominan como buena y algunas muy buena, con pocas negativas.

Por lo tanto, de este nivel se puede determinar que:

- La experiencia general ha sido buena.
- La sensación de aceleración ha sido correcta.

- La velocidad del personaje igual era demasiado elevada.
- El control durante el salto es conflictivo para según qué tipo de jugadores.
- La fuerza de salto está bien.
- La deceleración del personaje está bien.
-

7.3.2. Resultados del Nivel 2

Los valores que simulan las gráficas ASR del segundo nivel se encuentran en la Figura 42.

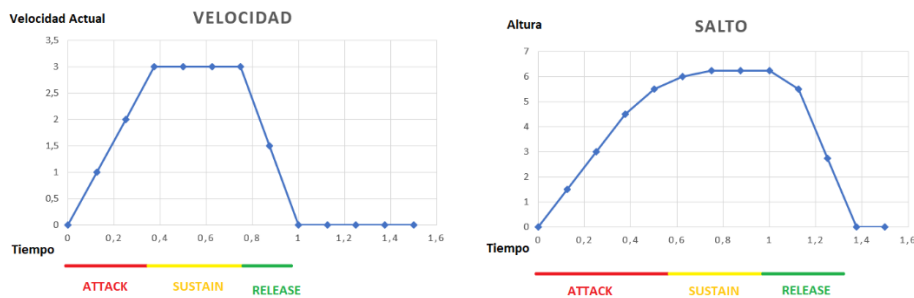


Figura 42 - Representación ASR de la velocidad y el salto del personaje en el Nivel 2.

Continuando con los resultados del segundo nivel, estos se plasman en la Figura 43.

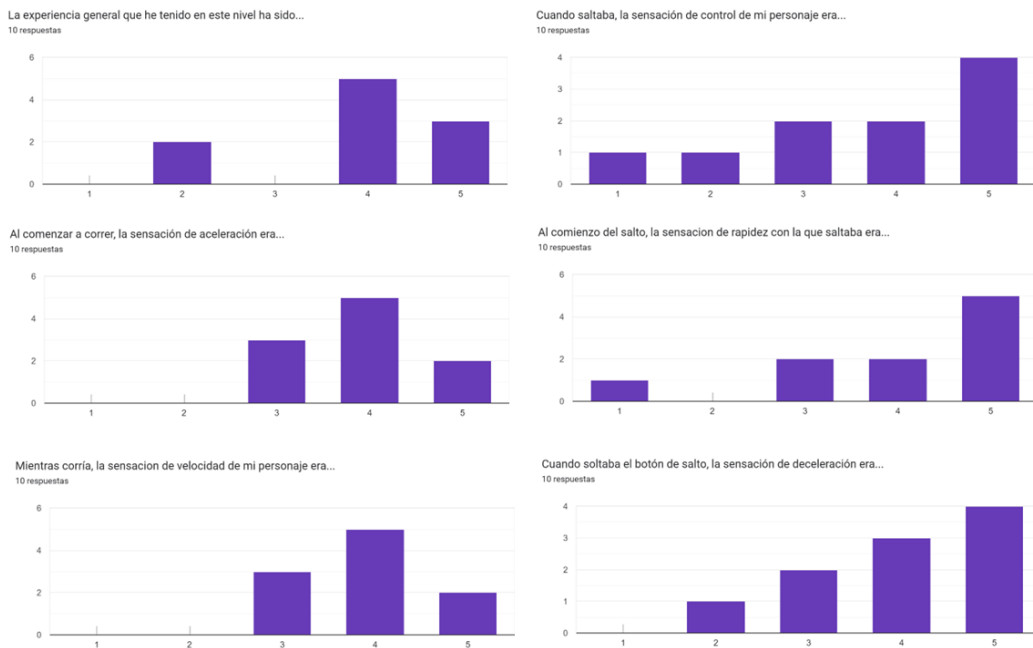


Figura 43 - Resultados del Nivel 2.

Para algunos usuarios la experiencia general ha sido mala mientras que para la mayoría ha sido buena y muy buena.

Tanto la sensación de aceleración como la de velocidad se encuentran con unos resultados iguales, siendo para la mayoría buena, luego unos pocos que ha sido neutral y la minoría seleccionándola como muy buena.

En cuanto a la sensación de control del salto hay bastantes opiniones dispares, pero se declara una mayoría que indica que la sensación es muy buena, con algunas pocas indicándola como buena o neutral y muy pocas como mala o muy mala.

La rapidez de salto tiene una gran mayoría de votos como muy buena, y unos pocos en buena y neutral, y un solo voto en muy mala.

En cuanto a la deceleración del salto, se marca como muy buena también, siguiéndola de buena, neutral, y un solo voto denominándola mala.

El análisis de los votos sobre este nivel define que:

- La experiencia general ha sido buena, pero con dos personas que no les ha gustado.
- La sensación de aceleración ha sido buena.
- La velocidad del personaje es correcta.
- El control durante el salto tiene muy buena valoración.
- La fuerza de salto está muy bien.
- La deceleración del personaje está muy bien.

7.3.3. Resultados del Nivel 3

En el tercer nivel se obtienen unos valores similares a los de las gráficas ASR de la Figura 44.

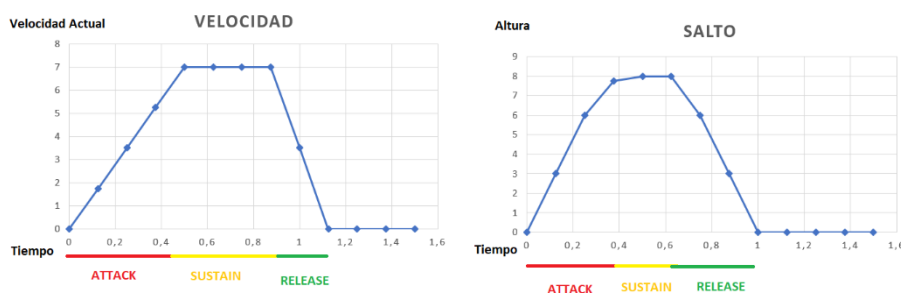


Figura 44 - Representación ASR de la velocidad y el salto del personaje en el Nivel 3.

Y los resultados obtenidos en las encuestas del tercer nivel se reflejan en la Figura 45.

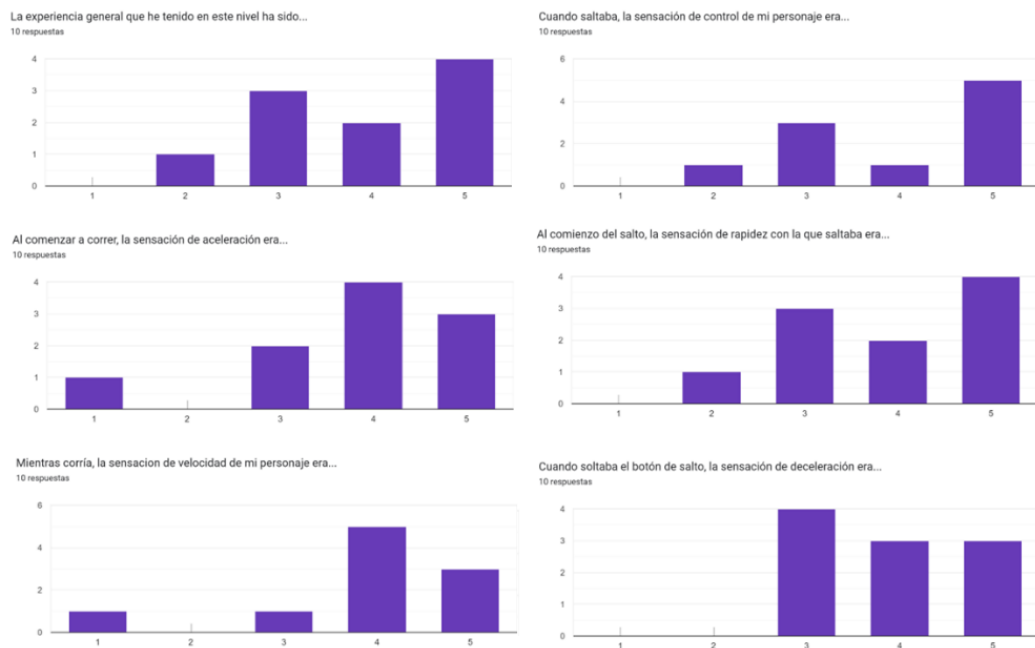


Figura 45 - Resultados del Nivel 3.

En el nivel 3 se encuentra que la experiencia ha mayoritariamente muy buena, pero con algunos votos mixtos entre neutral y buena, y un solo voto como mala.

La sensación de aceleración se ha calificado también como buena y muy buena con algún voto en muy mala y neutra.

Respecto a la sensación de velocidad ha calificado similar a la aceleración, con mayoría de buena y luego muy buena con algún voto en neutra y muy mala.

En la sensación de salto se ha calificado como una mayoría en muy buena seguida de una calificación neutra y unas pocas respuestas en mala y buena.

La rapidez del salto se ha calificado como muy buena, seguida de neutra y luego buena, con una respuesta como mala.

Finalmente, la sensación de deceleración se ha calificado con una mayoría neutra, seguida de muy buena y buena en la misma valoración.

Sobre los votos de este nivel se obtiene que:

- La experiencia general ha ido muy bien para algunos, pero neutra para otros.
- La sensación de aceleración ha sido buena en general.
- La velocidad del personaje es buena.
- El control durante el salto se ha notado muy bien para algunos, pero neutro para otros.
- La fuerza de salto tiene unas opiniones similares al control.
- La deceleración genera una sensación más neutra pero buena en general.

7.3.4. Respuestas libres

La sección de respuesta libre se ha dejado para la última en la que los usuarios recopilarían con sus propias palabras la experiencia que han tenido mientras jugaban.

Con esta pregunta se ha obtenido un feedback más directo y crítico de los usuarios permitiendo así conocer partes en las que el juego falla o como podrían mejorarse.

Entre las respuestas obtenidas, las más críticas y relevantes han sido:

- "En general la experiencia a lo largo de los 3 niveles ha estado bien: los controles están bien adecuados, aunque considero que la aceleración mientras se salta es demasiado alta y llega a generar algo de imprecisión, sobre todo en el último nivel donde más precisión se requiere."
- "El fallo con el salto cuando estas cerca de un desnivel y que en ocasiones la aceleración era demasiado rápida son las cosas que me han gustado menos, por el resto bien y muy disfrutable"
- "En el nivel 1 el personaje iba demasiado rápido, tanto que llegaba un momento en el que perdías el control. Saltaba mucho. El nivel 2 tiene una bajada de velocidad y de salto considerable, tanto que es hasta chocante. En el nivel 3 va perfecta la velocidad, aunque me gustaría que pudiera dar los saltos del nivel 1. El nivel 4 (el final) es un poco confuso."
- "La 1 iba muy rápido. La 2 y la 3 están muy bien, pero a mi gusto la velocidad ideal sería algo intermedio entre estas 2."

- “El nivel 1 es el que más "responsive" se siente en general y es la mejor experiencia para mí, y el 2 el que peor, el salto se bloquea si lo haces pegado a un borde. El nivel 3 es bastante agradable también en general y responde bien.”

7.3.5. Análisis de los resultados

Disponiendo ya de los resultados que se han obtenido de las encuestas, se pasa a comparar y analizar qué significan estos resultados.

En cuanto a la experiencia recibida, los tres niveles han resultado en una buena experiencia, pero sí que las mejores se han dado en el segundo y tercer nivel. Esto ya es un indicativo de que preferencias han tenido los jugadores.

Sobre la aceleración del personaje, se ve que los jugadores prefieren experiencias donde la aceleración sea más neutra, ni que alcance muy rápido la velocidad máxima, ni que tarde demasiado. Aun así, se ve un conflicto ya que en el primer nivel la aceleración es más baja y hace que le cueste mucho llegar a la velocidad máxima al personaje, pero al disponer de una velocidad tan alta los jugadores no perciben tanto la diferencia.

Respecto a la sensación de velocidad sí que ha habido más opiniones dispares en el primer nivel y se considera este como la peor sensación de velocidad. En el resto de los niveles ha habido más unanimidad sobre que la sensación de velocidad era buena en rasgos generales. Además, gracias a los comentarios, se puede ver que la opinión sobre el primer nivel es que el personaje “iba demasiado rápido”.

En el control del salto se denota también la rapidez del personaje, ya que al saltar este se ve influenciado por la velocidad horizontal que lleva. Aquí se puede ver claramente que el primer nivel es el que más opiniones dispares ha recibido, siendo los niveles dos y tres los que mejor se valora el control. También es cierto que un pequeño fallo en el salto mientras el personaje se mueve puede haber influenciado la sensación del control del salto del nivel dos como aparece en un comentario que dice “...y el 2 el que peor, el salto se bloquea si lo haces pegado a un borde...”.

Midiendo la rapidez del salto el nivel dos es el que tiene mejor nota como muy bueno junto al nivel tres, mientras que el nivel uno tiene la mayoría de nota en bueno. Esto refleja que los jugadores prefieren un salto que sea más ajustado y que no sea tan alto, explícitamente

comentado con "...el nivel 1 el personaje iba demasiado rápido, tanto que llegaba un momento en el que perdías el control. Saltaba mucho...", aunque también comenta sobre el nivel tres "...me gustaría que pudiera dar los saltos del nivel 1...", posiblemente debido a la dificultad de las plataformas que se diseñó en ese nivel.

Finalmente, sobre la sensación de deceleración, donde más se ha valorado correctamente ha sido en el segundo nivel, donde el salto era más corto y la deceleración mantenía una curva menos pronunciada al dejar de soltar el botón. En el tercer nivel ha habido una mayoría que lo sentía como neutral y en el primero lo notaban como bueno y muy bueno, pero con alguna nota más negativa.

Ya partiendo de los comentarios obtenidos, claramente se describen el segundo y tercer nivel como los mejores, donde la mejor experiencia resultaría en una mezcla de estos niveles. La queja general viene por el primer nivel en el que la velocidad y el control del salto es demasiado rápido y en el último nivel también hay parte de imprecisión a la hora de saltar.

Teniendo los comentarios en cuenta y el análisis de las respuestas, se puede crear una tabla de valores nuevos para el juego que lograrían una mejor experiencia general para el usuario. En la Figura 46 se muestran los valores que tendría cada uno basado en los resultados que se han obtenido y los comentarios de los usuarios, donde los valores estarían más cercanos a las experiencias de los niveles dos y tres con un algún ajuste en base a las opiniones recibidas como por ejemplo con la aceleración más cercana al nivel 3 y la velocidad máxima e impulso de salto más cercana al nivel 2 ya que permite un mejor control.

maxSpeed	jumpTakeOffSpeed	jumpModifier	jumpDeceleration	timeZeroToMax
4	5	1.25	0.7	1.1

Figura 46 - Nuevos valores para los parámetros en base a los resultados obtenidos.

La gráfica ASR que simularía la velocidad y el salto en este último caso se refleja en la Figura 47.

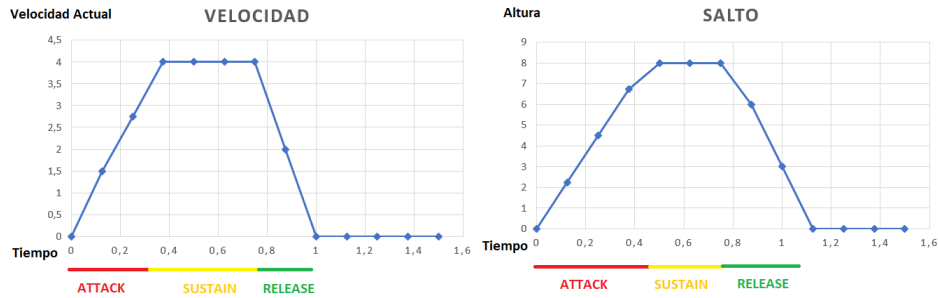


Figura 47 - Representación ASR de la velocidad y el salto del personaje en la versión final.

7.3.6. Test final y resultados

Una vez aplicados los nuevos valores a los parámetros que modifican la sensación de juego se vuelve a compartir con unos pocos usuarios, en este caso con 5 usuarios sería más que suficiente para ver cómo han influido estos valores en sus sensaciones.

La encuesta en este caso se compone de las mismas preguntas, pero solamente en una sección, ya que todos los niveles van a disponer de los mismos valores. En este caso el resultado esperado es que los valores de las respuestas se muevan entre "buena" y "muy buena". Para esta prueba no resulta relevante volver a comprobar en que plataforma han jugado y se les ha pedido rellenar una nueva encuesta, la cual solo contendría la introducción, un cuestionario con las mismas preguntas que se han hecho sobre los niveles, pero aplicado a todo el juego, y luego la parte de respuesta libre.

Los resultados obtenidos en esta fase final de pruebas se plasman en la Figura 48, donde se puede apreciar que la mayoría de las sensaciones han sido muy buenas, con unanimidad en la sensación de aceleración, y una mayoría de buenas sensaciones en el control del personaje al saltar. Por estos resultados se puede ver que la versión final ha superado con creces las notas de la anterior versión.

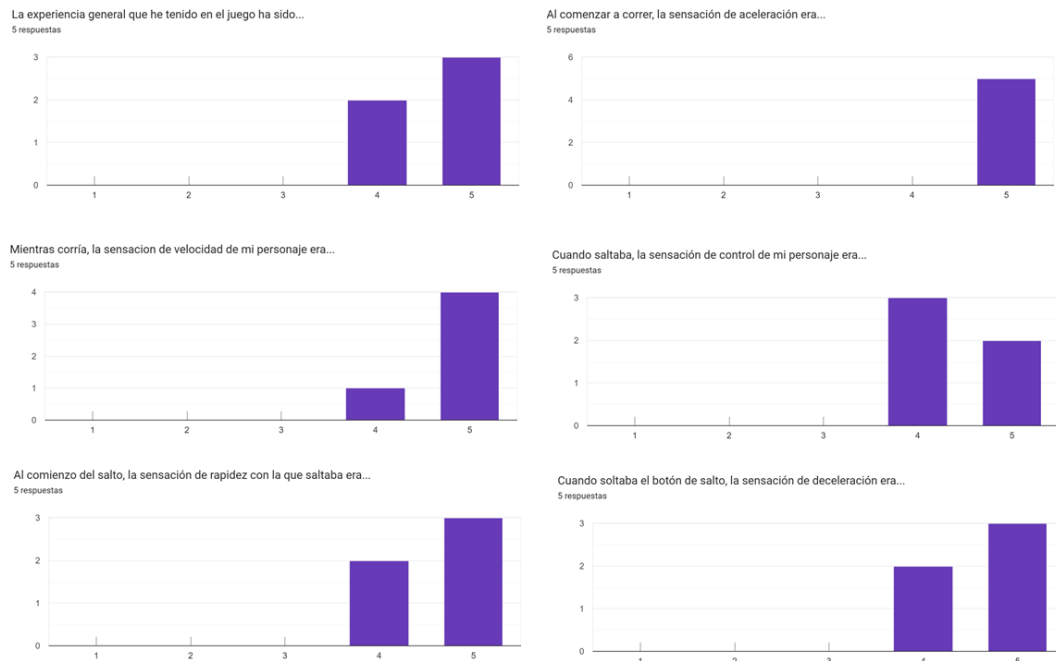


Figura 48 - Resultados del test final.

Pasando a las respuestas libres, se encuentran 4 respuestas que han aportado los usuarios y que dicen:

- "Noto mejoría en el control del movimiento"
- "Muy intuitivo, como te dice el cerebro que debería responder y como responde coinciden a la perfección."
- "El salto, velocidad y control que tenía del personaje esta versión me han parecido más cómodos que la anterior, como fallo una zona donde no podías ver el final de la caída y no podías saber si caías en una plataforma o vacío."
- "La mejora de controles ha sido sustancial en comparación a los otros controles. No sé hasta qué punto es que ya me conocía el mapa y las mecánicas o que en verdad son más apropiados estos controles."

Todas coinciden con que ha habido una gran mejoría en el control del personaje respecto al anterior. Declaran el juego como "intuitivo" y "cómodo", y sacan a relucir algún apartado mejorable que corresponde al diseño de niveles.

En definitiva, con este test final se ha logrado conseguir una gran mejoría en cuanto a la valoración de los usuarios y con ello determinar cómo definir una sensación de juego mucho mejor. Los resultados han sido claramente beneficiosos para el juego y la demo final ha logrado

contentar a los usuarios, creando una experiencia agradable a la hora de jugarlo y pudiendo así entretener al jugador.

7.3.7. Conclusiones del estudio

Finalmente, a modo de conclusión, con lo demostrado en este proyecto, los métodos utilizados y el estudio realizado podrían aplicarse, con un enfoque más profesional, en juegos comerciales, pudiendo así obtener una mejor sensación de juego y disfrutando de una mayor calidad para el usuario final. Por supuesto, requeriría de un equipo profesional dedicado a estudiar como mejorar el producto ya que este proyecto ha tenido un alcance ajustado para una persona y puede llegar a ser mejor con un equipo completo.

Lo obtenido en este proyecto y lo presentado en el apartado 3 se ha cumplido en su totalidad con un tiempo dedicado correcto, tanto los apartados de documentación, el desarrollo de la demo jugable, el estudio realizado sobre el Game Feel y la obtención de una demo final con un buen grado de satisfacción por parte de los usuarios.

8. Conclusiones

El Game Feel, pese a ser un concepto relativamente nuevo, sirve como guía y apoyo a los diseñadores de videojuegos que quieren crear la mejor experiencia posible para los usuarios. Ser conocedor de cómo crear estas experiencias y buenas sensaciones es una de las partes fundamentales que se deben enseñar a la hora de aprender a diseñar juegos.

Este proyecto ha servido como puente en el desarrollo de videojuegos hacia el diseño de videojuegos, y sobre todo como solucionar problemas sobre cómo abordar las mecánicas de un videojuego o como mejorar las sensaciones que provocan.

El estudio del concepto del Game Feel ha resultado extremadamente conveniente, desde las fuentes más sólidas, ha permitido que este proyecto haya salido adelante. La elección del juego sobre la que aplicarlo ha ayudado a profundizar más en cómo puede afectar la sensación de juego al usuario y poder probar más a fondo distintos métodos de modificar las sensaciones.

Por la parte del diseño del estudio, se considera que es un apartado fundamental que se debería realizar en multitudes de proyectos, que realizar este tipo de estudios puede solventar muchos problemas que les surgirán a muchísimos desarrolladores a lo largo de su carrera, por lo que este trabajo puede servir como ejemplo para tratar de mejorar la sensación de juego en juegos más profesionales.

Respecto al tiempo dedicado, ha sido más que correcto para realizar este proyecto. Se ha dedicado bastante tiempo al aprendizaje sobre el concepto de Game Feel y el cómo mejorar los apartados jugables de un videojuego, y por el apartado del desarrollo y resultados se ha logrado alcanzar la meta y obtener una demo final con todo a punto en el tiempo requerido.

De cara al futuro y con más tiempo invertido, este proyecto podría expandirse al resto de métricas definidas en [1] y conseguir mejorar la sensación de juego en muchos más apartados, obteniendo así una experiencia jugable aún mejor, pudiendo aplicarse finalmente a videojuegos comerciales y expandir más el concepto del Game Feel en el sector del videojuego.

Para finalizar, el proyecto ha resultado en una mejora personal y servirá en el futuro a la hora de desarrollar un videojuego, ya sea en un equipo o de forma individual.

9. Bibliografía

- [1] S. SWINK, GAME FEEL, A game's designer guide to visual sensation, -: Morgan Kaufmann, 2008. <https://gamifique.files.wordpress.com/2011/11/2-game-feel.pdf>.
- [2] WIKIPEDIA, «Super Mario Bros.,» 1985. https://es.wikipedia.org/wiki/Super_Mario_Bros.
- [3] WIKIPEDIA, «Super Mario 64,» 1996. https://es.wikipedia.org/wiki/Super_Mario_64.
- [4] VIOTEK, «Why are 144Hz monitors important for gaming,» <https://viotek.com/144hz-monitors-important-gaming/>.
- [5] XCORR, «What's the maximal frame rate humans can perceive?,» 2020. <https://xcorr.net/2011/11/20/whats-the-maximal-frame-rate-humans-can-perceive/>.
- [6] S. SWINK, «Game Feel: The Secret Ingredient,» 2007. <https://www.gamedeveloper.com/design/game-feel-the-secret-ingredient>.
- [7] M. J. MARTIN PICHLMAIR, «Designing Game Feel. A survey,» 2021. <https://ieeexplore.ieee.org/abstract/document/9399794>.
- [8] A. BERRAONDO, «Estudio del efecto de diferentes parámetros de diseño de videojuegos en el Game Feel o Sensación de Juego,» 2021. <https://repositorio.usj.es/handle/123456789/615>.
- [9] UNITY3D, «Rigidbody – Documentación oficial,» 2021.1. <https://docs.unity3d.com/es/2021.1/Manual/class-Rigidbody.html>.
- [10] UNITY3D, «Order of execution for event functions,» 2021.3. <https://docs.unity3d.com/Manual/ExecutionOrder.html>.
- [11] R. GURU, «Singleton,» <https://refactoring.guru/es/design-patterns/singleton>.
- [12] UNITY3D, «BoxCollider2D – Documentación oficial,» 2021.1. <https://docs.unity3d.com/es/2021.1/Manual/class-BoxCollider2D.html>.
- [13] STATCOUNTER, «OS Market Share,» 2022. <https://gs.statcounter.com/os-market-share>.
- [14] WIKIPEDIA, «Port (informática),» [https://es.wikipedia.org/wiki/Port_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Port_(inform%C3%A1tica)).
- [15] UNITY3D, «Canvas – Documentación oficial,» 2021.1. <https://docs.unity3d.com/es/2021.1/Manual/UICanvas.html>.
- [16] GLASSDOOR, «Sueldos de programador de videojuegos,» 2022. https://www.glassdoor.es/Empleo/espa%C3%B1a-videojuegos-empleos-SRCH_IL.0,6_IN219_KO7,18.htm?srs=JOBS_HOME_RECENT_SEARCHES.
- [17] GLASSDOOR, «Sueldos de programador de juegos en Mercury Steam,» 2022. https://www.glassdoor.es/Sueldo/Mercury-Steam-Entertainment-Game-Programmer-Sueldos-E556029_D_KO28,43.htm.
- [18] WIKIPEDIA, «Syobon Action (Cat Mario),» 2007. https://en.wikipedia.org/wiki/Syobon_Action.
- [19] STEAM, «ALTF4,» 2021. <https://store.steampowered.com/app/1137460/ALTF4/>.
- [20] QUESTIONPRO, «¿Qué es la escala de Likert y cómo utilizarla?,» <https://www.questionpro.com/blog/es/que-es-la-escala-de-likert-y-como-utilizarla/>.
- [21] M. BERTHOLLET, «Heurísticas de usabilidad para la interfaz de usuario de Nielsen,» <https://bringconnections.es/tipos-heuristicas-usabilidad-nielsen/>.

Anexo I – Propuesta de proyecto

Nombre alumno: Álvaro Tomás Lozano

Titulación: GRADO EN DISEÑO Y DESARROLLO DE VIDEOJUEGOS

Curso académico: 2021-2022

1. TÍTULO DEL PROYECTO

Game Feel: Estudio de caso

2. DESCRIPCIÓN Y JUSTIFICACIÓN DEL TEMA A TRATAR

El proyecto versa sobre el concepto de "Game Feel" en videojuegos y la aplicación de este concepto en un estudio de caso.

3. OBJETIVOS DEL PROYECTO

Los objetivos del proyecto son:

- Conocer el estado del arte en el concepto de Game Feel de videojuegos
- Creación y aplicación de Game Feel en un videojuego
- Descripción de estudio de caso

4. METODOLOGÍA

La metodología se fijará en las primeras reuniones con el tutor.

5. PLANIFICACIÓN DE TAREAS

La planificación se especificará en las primeras reuniones con el tutor.

6. OBSERVACIONES ADICIONALES

Anexo II – Reuniones

REUNIÓN: 1

Fecha: 29/10/2021	
Hora comienzo: 11:00	Hora finalización: 12:00
Lugar: Remoto desde casa	
Elabora acta: Álvaro Tomás	
Convocados: Álvaro Tomás, Jorge Echeverria	

Orden del día / Acta

No.	Asunto	Acuerdo
1	Definir el estado del arte del proyecto.	
2	Definir las fuentes en las que se va a basar el proyecto.	
3	Hablar sobre trabajos anteriores basados en el Game Feel.	001
4	Definir la metodología con una herramienta de gestión de proyectos.	002
5		
6		
7		
8		
9		
10		

Resumen de acuerdos

Número	Acuerdo	Plazo	Responsable
001	Leer trabajos relacionados con el Game Feel.	Siguiente reunión	Álvaro Tomás
002	Montar y preparar un Trello para su uso.	Siguiente reunión	Álvaro Tomás
003			
004			
005			
006			

REUNIÓN: 2

Fecha: 13/01/2022	
Hora comienzo: 17:00	Hora finalización: 18:00
Lugar: Remoto desde casa	
Elabora acta: Álvaro Tomás	
Convocados: Álvaro Tomás, Jorge Echeverria	

Orden del día / Acta

No.	Asunto	Acuerdo
1	Revisión del Trello y su estructura.	
2	Proponer nuevas fuentes para el proyecto.	
3	Hablar de la elección para el juego que se va a tratar.	001
4	Trabajo sobre la memoria antes de desarrollar el videojuego.	002
5		
6		
7		
8		
9		
10		

Resumen de acuerdos

Número	Acuerdo	Plazo	Responsable
001	Decidir que juego se va a utilizar para el desarrollo.	Siguiente reunión	Álvaro Tomás
002	Comenzar con el estado del arte de la memoria.	Siguiente reunión	Álvaro Tomás
003			
004			
005			
006			

REUNIÓN: 3

Fecha: 28/03/2022	
Hora comienzo: 17:00	Hora finalización: 18:00
Lugar: Remoto desde casa	
Elabora acta: Álvaro Tomás	
Convocados: Álvaro Tomás, Jorge Echeverria	

Orden del día / Acta

No.	Asunto	Acuerdo
1	Revisión del estado del arte y decisión del juego.	
2	Continuar sobre las fuentes obtenidas.	001
3	Comentarios sobre la bibliografía y edición.	002
4		
5		
6		
7		
8		
9		
10		

Resumen de acuerdos

Número	Acuerdo	Plazo	Responsable
001	Continuar con la documentación y estado del arte.	Siguiente reunión	Álvaro Tomás
002	Revisión de los saltos de línea.	Siguiente reunión	Álvaro Tomás
003			
004			
005			
006			

REUNIÓN: 4

Fecha: 18/07/2022	
Hora comienzo: 16:30	Hora finalización: 17:00
Lugar: Remoto desde casa	
Elabora acta: Álvaro Tomás	
Convocados: Álvaro Tomás, Jorge Echeverria	

Orden del día / Acta

No.	Asunto	Acuerdo
1	Revisión del estado del arte y del desarrollo	001
2	Proponer trabajo en metodología.	002
3	Comentar como se asignan las referencias al proyecto.	003
4		
5		
6		
7		
8		
9		
10		

Resumen de acuerdos

Número	Acuerdo	Plazo	Responsable
001	Seguir con el desarrollo de la demo.	Siguiente reunión	Álvaro Tomás
002	Escribir sobre el apartado de metodología.	Siguiente reunión	Álvaro Tomás
003	Referenciar bien la bibliografía a lo largo de la memoria.	Siguiente reunión	Álvaro Tomás
004			
005			
006			

REUNIÓN: 5

Fecha: 18/08/2022	
Hora comienzo: 17:00	Hora finalización: 18:00
Lugar: Remoto desde casa	
Elabora acta: Álvaro Tomás	
Convocados: Álvaro Tomás, Jorge Echeverria	

Orden del día / Acta

No.	Asunto	Acuerdo
1	Revisión de la demo jugable, memoria y planteamiento del estudio.	
2	Propuesta de aplicaciones al estudio.	001
3	Propuesta de mejoras para la memoria.	002
4		
5		
6		
7		
8		
9		
10		

Resumen de acuerdos

Número	Acuerdo	Plazo	Responsable
001	Definir el estudio, preguntas, y comenzar a redactar las ideas.	Siguiente reunión	Álvaro Tomás
002	Mejoras de comentarios sobre la memoria.	Siguiente reunión	Álvaro Tomás
003			
004			
005			
006			

REUNIÓN: 6

Fecha: 30/08/2022	
Hora comienzo: 17:00	Hora finalización: 18:00
Lugar: Remoto desde casa	
Elabora acta: Álvaro Tomás	
Convocados: Álvaro Tomás, Jorge Echeverria	

Orden del día / Acta

No.	Asunto	Acuerdo
1	Revisión del estudio y la memoria.	
2	Propuestas de mejoras en apartados de introducción, estudio económico, conclusiones y finalización del estudio.	001
3		
4		
5		
6		
7		
8		
9		
10		

Resumen de acuerdos

Número	Acuerdo	Plazo	Responsable
001	Mejorar todos los apartados y finalizar el estudio.	Siguiente reunión	Álvaro Tomás
002			
003			
004			
005			
006			

REUNIÓN: 7

Fecha: 04/07/2022	
Hora comienzo: 15:30	Hora finalización: 16:00
Lugar: Remoto desde casa	
Elabora acta: Álvaro Tomás	
Convocados: Álvaro Tomás, Jorge Echeverria	

Orden del día / Acta

No.	Asunto	Acuerdo
1	Revisión de la memoria final.	001
2		
3		
4		
5		
6		
7		
8		
9		
10		

Resumen de acuerdos

Número	Acuerdo	Plazo	Responsable
001	Finalizar la memoria con las correcciones y entregarla.	07-09-2022	Álvaro Tomás
002			
003			
004			
005			
006			