
This is the **published version** of the bachelor thesis:

Lin, Xiang; Bernal del Nozal, Jorge, dir. Análisis y mejora de la Inteligencia Artificial y mecánicas de un videojuego de acción. 2022. (958 Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/264197>

under the terms of the  license

Análisis y mejora de la Inteligencia Artificial y mecánicas de un videojuego de acción

Xiang Lin

Resumen—La Inteligencia Artificial es una característica cada vez más importante en el mundo que vivimos y pese a que los videojuegos las implementan de una forma distinta, lo cierto es que aun así tienen una importancia central en ellos. Por ello, en este proyecto se ha analizado el videojuego creado por otro estudiante universitario y se ha buscado mejorar notablemente la “experiencia de usuario” al introducir cambios en la Inteligencia Artificial de sus enemigos, su algoritmo de generación de niveles y su diseño de niveles, entre otras cosas. Como resultados, los cambios aplicados han mejorado notablemente la experiencia de usuario y esto se ha corroborado a través de la opinión de varios jugadores que han probado y comparado la versión original de PolygonSouls con sus iteraciones posteriores.

Palabras clave—PolygonSouls, Inteligencia Artificial, experiencia de usuario, jugadores, Godot, GDScript, videojuego, dificultad, accesibilidad.

Abstract—Artificial Intelligence is a increasingly important feature in the world we live in and despite the fact that video games implement then in a unique way, the truth is that they still have a central importance in them. For this reason, this project has analyzed the videogame created by another university student and has sought to considerably improve the “user experience” by introducing changes in the Artificial Intelligence of its enemies, its level generation algorithm, and its level design, among other things. As a result, the applied changes have greatly improved the user experience and this has been corroborated through the opinion of several players who have tested and compared the original version of PolygonSouls with its later iterations.

Index Terms— PolygonSouls, Artificial Intelligence, user experience, players, Godot, GDScript, video game, difficulty, accessibility.



1. INTRODUCCIÓN

1.1 Motivación del proyecto

La Inteligencia Artificial (I.A.) es uno de los componentes más importantes para que un videojuego sea desafiante y divertido. Esta característica es intrínseca en los juegos que tienen NPCs, pero también está presente en otros muchos géneros según las necesidades y visión del equipo de desarrollo.

Cuando una I.A. está bien implementada y tiene un objetivo claro, puede hacer que un juego sea único y mucho más interesante y un ejemplo de ello es el *Xenomorfo* de *Alien: Isolation*. En este juego, el monstruo acecha al jugador mediante un complejo sistema de decisiones que le permite saber en todo momento dónde está aproximadamente su presa. Esto genera narrativas orgánicas únicas para cada persona, algo que hace que una *experiencia de usuario* sea memorable.

Sin embargo, en la industria hay bastantes más ejemplos de I.A.s poco cuidadas. Como referentes famosos podemos encontrar casos como el de los ciudadanos de *Cyberpunk 2077* (antes de la versión 1.5), quienes, ante un tirador amenazante, huían en línea recta y no variaban su trayectoria incluso si éste les adelantaba. Otro ejemplo de esto es la circulación de vehículos en *Cities Skylines*, puesto que debido a una mala codificación se genera más tráfico del que debería. Dicho problema ha sido arreglado por los propios jugadores a través de “*mods*” para el juego.

En este proyecto utilizaremos como base el TFG [12] de otro estudiante universitario debido a que su juego tiene implementada una I.A. sencilla y poco profunda. Se le intentará aplicar cambios a distintos niveles para mejorar notablemente la experiencia de juego.

1.2 Objetivos del proyecto

El objetivo principal del proyecto es mejorar la I.A. de los enemigos, implementar algunos enemigos nuevos y mejorar la función de generación de niveles para que de esta forma la experiencia de juego mejore.

Asimismo, existen objetivos secundarios. Por ejemplo, añadir al juego nuevos diseños, mejorar la *Interfaz de Usuario* (I.U.) o hacer que el juego tenga más de un solo nivel. No obstante, estos cambios solo se buscarán una vez el núcleo de proyecto ya esté bien cimentado.

E-mail de contacto: adrian.lin.xiang@gmail.com

Mención realizada: Computació

Trabajo tutorizado por: Jorge Bernal del Nozal
(Ciencias de la Computación)

Curso 2021/22

1.3 Riesgos del Proyecto

Riesgo	Definición	Impacto	Contingencia
Ausencia de videojuego a analizar	No conseguir un juego con una I.A. sobre el que hacer el proyecto	Medio	Buscar un juego que sirva como alternativa.
Finalización de las prácticas extracurriculares	Al terminar las prácticas extracurriculares se ha de devolver el portátil de empresa	Medio	Mantener el proyecto en la nube y en diferentes dispositivos para poder trabajar en cualquier PC.
Rotura del ordenador o portátil de trabajo	El PC o portátil podría estropearse.	Medio	Mantener el proyecto en la nube y diferentes dispositivos
Actualización importante de la herramienta Godot	Godot podría actualizarse a una nueva versión que generase problemas con proyectos antiguos	Bajo	Se debería revisar el proyecto y adaptar los assets o, en su defecto, utilizar una versión anterior de Godot.
Caída del sistema informático de la UAB	Que ocurriese un nuevo ataque informático o que los servidores de la UAB se cayesen de forma indefinida	Medio	Trabajar y revisar el trabajo en local y hacer las entregas cuando el sistema vuelva a funcionar.

1.4 Requisitos del proyecto

Hardware

En cuanto a hardware, solo es necesario poder correr *Godot* y el juego resultante. Para ello, hace falta tener al menos *Windows 7*, soporte para *OpenGL 2.1* y unos cuatro (4) GB de RAM.

Por otro lado, debido a que la herramienta cuenta con una interfaz gráfica interactiva, en portátil es recomendable contar con un ratón para agilizar enormemente el trabajo.

Software

Respecto a los programas que se usarán, como se ha indicado anteriormente, la herramienta principal será *Godot 3.4.4* y el lenguaje de programación principal *GDScript*, el cual es similar a *Python*.

Por otro lado, el videojuego podría replicarse en otros motores gráficos como *Unreal Engine* o *Unity*, los cuales cuentan con bastantes recursos y mucho soporte gracias a sus grandes comunidades.

El primer motor gráfico se basa en C++ y permite visuales y físicas bastante más realistas. Además, tiene el sistema de scripts visuales (*Blueprints*) para simplificar el proceso de programación. Por su lado, *Unity* es ampliamente más usado en proyectos pequeños y está basado en C#, por lo que es más sencillo y fácil de aprender.

Sea como fuere, dado que el proyecto previo fue realizado con *Godot+GDScript* y el objetivo es mejorar la I.A., no crear un videojuego, se continuará en el mismo entorno y lenguaje que el proyecto anterior.

1.5 Metodología

Se utilizará la metodología *Kanban* [11]. Esta decisión se ha tomado ya que es bastante sencillo e iterativo y nos permite identificar fácilmente las tareas que están por hacer, en proceso o finalizadas. Además, podemos utilizar *Kanban* en un proyecto individual, a diferencia de alternativas como *Scrum*.

Asimismo, para asegurarnos de que el ritmo de trabajo es estable y adecuado, se irán haciendo reuniones semanales de supervisión con el tutor.

1.6 Planificación Temporal

Se han hecho algunas estimaciones del tiempo que se necesitará para hacer las tareas más importantes del proyecto, las cuales podrían cambiar en función de las necesidades de cada etapa. La planificación temporal está representada en un diagrama de *Gantt* mostrado en el Anexo.

Se ha empezado a utilizar *Shortcut* [9] para organizar el trabajo en formato *Kanban*. Dichas tareas están atomizadas lo máximo posible y algunas de ellas cuentan con una *deadline*, así como unos *flags* que indican si tienen alguna tarea previa obligatoria.

2. ESTADO DEL ARTE

En lo que al estado del arte se refiere, tal como lo indica el propio autor original, *PolygonSouls* quiere transportar la experiencia de los juegos tipo *Souls-like*. Si bien el estilo de juego es muy distinto, podemos ver algunas similitudes como la presencia de hogueras o que algunos enemigos adaptan su estilo de combate según nuestra forma de atacar. Aun así, se han tomado ideas de otros títulos muy conocidos como *Resident Evil 4* o la saga *Pokémon*.



Figura 1: *The Binding of Isaac*, uno de los rogue-likes más conocidos y jugados. "TheGamerMaldito - BINDING OF ISAAC: REBIRTH #1 - ¡Primera partida!":

Por otro lado, es de mencionar el parecido de *PolygonSouls* con juegos del género *rogue-like*, los cuales suelen tener una vista cenital o isométrica. Los títulos con más similitud podrían ser: *The Binding of Isaac* (Fig. 1), *Hades* o *Enter the Gungeon*.

Todos estos juegos destacan por tener una creación de mundos procedural, jefes de nivel y diferentes tipos de habilidades para lidiar con nuestros enemigos. Además, todos ellos han sido ampliamente alabados tanto por la crítica, como por la comunidad de jugadores, razón por la que me inspiraré en ellos para intentar proponer nuevas mejoras.

3. DESARROLLO

Durante el desarrollo del proyecto, lo primero que se ha hecho es estudiar el entorno de trabajo, es decir, *Godot 3.4.4*. Para ello, se ha recurrido principalmente al estudio de la documentación online de Godot [1], vídeo tutoriales de YouTube [2] [3] [4] y a “*Godot Engine Game Development Projects*”, un libro sobre la herramienta prestado por el tutor del TFG.

Lo siguiente que se ha hecho es estudiar el código de *PolygonSouls*. Se han revisado todas las clases, subclases y sus respectivas líneas de código, y algunas de ellas se han comentado en detalle para poder entender mejor su funcionamiento. Después de esto, se han implementado diversos cambios para mejorar la I.A. de los enemigos y la generación del nivel y, por otro lado, aumentar la calidad general del juego.

Asimismo, cabe destacar que se ha contactado con el autor original de *PolygonSouls* (A. J. P. Dedeu [5]) para preguntar algunas dudas acerca del juego base.

Los principales cambios del título están recogidos en un documento [10] donde se ha registrado el momento donde se introdujeron y su tipología. De esta forma, cada nueva característica se ha clasificado con las etiquetas: cambio, mejora, bug o *feedback*.

Por otro lado, desde un punto de estudio, si nos fijamos en la naturaleza de los cambios, se podrían clasificar en los siguientes puntos: algoritmo de creación de niveles, cambios generales de los enemigos, mejoras del jefe final, mejoras generales y de calidad de vida y problemas/bugs encontrados. A continuación, explicaré los diversos cambios aplicados dentro de sus respectivos campos.

3.1 Algoritmo de Creación de Niveles

El algoritmo original de *PolygonSouls* no era demasiado profundo, puesto que no tenía en cuenta ninguna característica. La función simplemente asignaba los roles de salas especiales de forma aleatoria lo que provocaba que a veces apareciera la sala del jefe final justo al lado de la sala inicial.

Para solventar esto, se ha trabajado en dos algoritmos para identificar la lejanía de cada habitación respecto a la sala inicial. Esta información nos permite crear las salas especiales en función de su distancia para posicionar las salas especiales adecuadamente.

El primer algoritmo asignaba niveles a medida que se creaban las salas, por lo que era problemático cuando se podía llegar a una habitación a través de diferentes caminos con lejanías distintas. Por ello, se optó por crear uno nuevo que recorriese todas las salas una vez estuviesen ya creadas.

La nueva función comprueba si la sala actual tiene vecinos y si estos no tienen asignados un nivel de lejanía o tienen uno superior al que se les quiere asignar, se les coloca el valor $K+1$, siendo K el nivel de lejanía la sala actual tal como se ve en la Figura 2. De esta forma, la sala inicial tiene distancia cero (0), sus vecinas distancia uno (1), las contiguas a éstas últimas distancia dos (2) y así sucesivamente hasta que no queden salas por asignar.

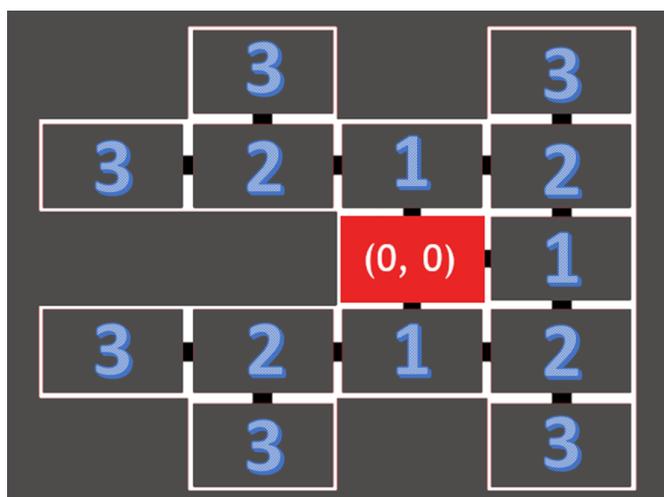


Figura 2: Asignación de niveles de lejanía de las salas.

A la hora de asignar roles a las salas, dado que tenemos solamente dos habitaciones especiales “jefe final” y “hoguera” tomamos la siguiente estrategia: Para la sala del jefe final, se busca el nivel de lejanía más alto que haya en esta iteración, luego se busca cuál es su índice dentro del *array* de salas y, una vez lo tenemos, se le asigna dicho rol. Para la sala de hoguera, se escoge el nivel seleccionado en la fase anterior, se divide entre dos y se redondea al alza. Entonces, del mismo modo, buscamos el índice de la primera habitación que tenga dicho nivel y se le asignará el rol mencionado previamente.

Otro problema importante ha sido la generación de enemigos, la cual tiene que ver con cómo se genera el mundo. En la versión inicial, esto se hacía mediante un límite inferior de uno (1) y uno superior de cinco (5) y una función elegía un número aleatorio dentro de dicho rango para generar ese número de enemigos. No obstante, esto hacía que el juego se sintiese demasiado aleatorio e inconsistente, ya que en algunas ocasiones las primeras salas podían ser demasiado difíciles o demasiado fáciles.

Para solventarlo, se decidió crear un algoritmo que estableciese una dificultad baja y la aumentase a medida que el jugador recorriese la mazmorra, es decir, que tuviese unos límites superiores e inferiores un poco más dinámicos. Utilizando la herramienta matemática desmos [8], se probó distintas funciones que cumpliesen los requisitos que se buscaban y finalmente se optó por:

$$\text{Indicador de Amenazas: } f(x) = \frac{-50[-y]}{x+12} + 6$$

La gráfica de la función (sin contar la variable Y) se puede ver en la [Figura 3](#) y genera dos límites al infinito: el horizontal en seis (6) y el vertical en menos doce (-12). Sin embargo, aquí solo nos importa el límite horizontal, ya que será quien establecerá el máximo efectivo de enemigos.

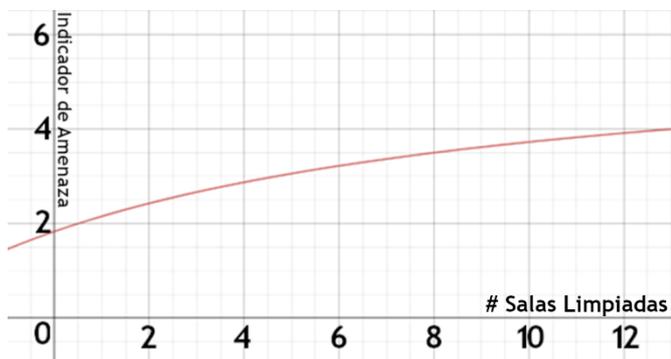


Figura 3: Curva de la fórmula del Indicador de Amenazas

En la fórmula, X representa el número de salas que hemos limpiado de enemigos, e Y representa el "indicador de muertes" del jugador (explicado más adelante). Finalmente, el resultado de la fórmula representa el valor del "indicador de amenazas", valor que aumenta a medida que el jugador avanza por la mazmorra.

Dicho indicador, al entrar en cada habitación, se multiplica por dos valores para establecer la cantidad mínima y máxima de enemigos que pueden aparecer. No obstante, cabe mencionar que además de los resultados calculados, hay un límite forzado de dos (2) y siete (7) enemigos para que la fórmula nunca ofrezca valores demasiado altos o bajos.

Gracias a esto, en la primera sala siempre aparecen entre dos (2) y tres (3) enemigos, en las siguientes entre dos (2) y (4), ... Y en las últimas salas antes del jefe final aparecen entre cuatro (4) y seis (6) adversarios. Esto hace que el juego se sienta más justo y mejor diseñado para que sea un reto divertido y progresivo.

Por otro lado, el indicador de amenaza también se utiliza para elegir el set de enemigos que pueden aparecer. Dado que el *spinEnemy* (que veremos más adelante) es el enemigo más complicado y complejo, éste solo puede aparecer cuando el indicador de amenaza está por encima de "3.00". En otras palabras, en una partida normal no aparecerá hasta que hayamos limpiado alrededor de cinco (5) salas.

Por último, se ha de hablar del sistema de dificultad dinámico inspirado en *Resident Evil 4*, el cual adapta el juego a la habilidad del jugador.

En *PolygonSouls 2.0* no hay un selector de dificultad, ya que se ha implementado un mecanismo que guarda cuántas veces hemos muerto en cada sesión de juego. Dicho número está representado como Y en la fórmula del indicador de amenazas e internamente se divide para disminuir su peso. De esta forma, cada vez que perezamos el indicador de amenazas bajará, haciendo que cada vez aparezcan menos enemigos y retrasando la aparición del *spinEnemy*.

Con estos dos cambios respecto a la aparición de enemigos se ha buscado alisar y pulir la curva de dificultad del juego para que sea más justo y accesible a las personas menos diestras en este tipo de videojuegos.

3.2 Cambios generales de los enemigos

En lo que se refiere a los tres enemigos originales vistos en la [Figura 4a](#) (*spinEnemy*, *kamikazeEnemy* y *turretEnemy*), se han abordado varios cambios importantes. Primero de todo, se ha editado la apariencia, naturaleza y comportamiento de todos ellos. Y, por otro lado, se ha añadido un nuevo tipo de amenaza, la cual se ha denominado "*bouncerEnemy*".

Si nos fijamos en el aspecto de los nuevos enemigos en la [Figura 4b](#), se ha intentado mejorar su claridad visual. De esta forma es más fácil reconocerlos e identificar qué hacen de un solo vistazo.

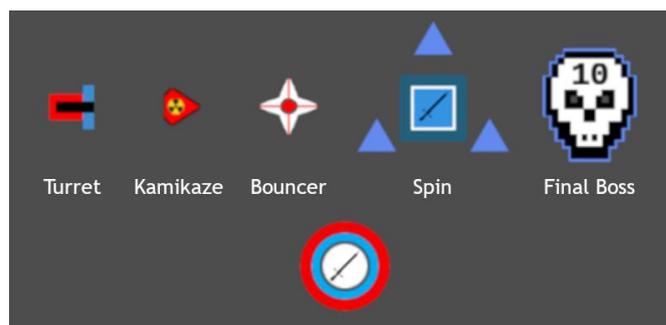
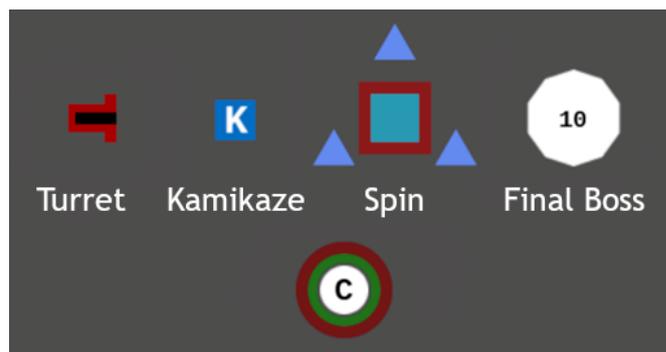


Figura 4a y b: Comparativa visual de los enemigos de *PolygonSouls* en la versión 1.0 y 2.0

Además, para guiar mejor al jugador, se ha establecido una paleta de colores que se relacionará en todo momento

con diversas ideas. Así, el color azul claro representa inmunidad/energía, mientras que el color rojo representa lo contrario: partes vulnerables donde el jugador puede atacar. Respecto a los cambios realizados a los enemigos, comenzaremos hablando por el *SpinEnemy*.

Este enemigo muestra ahora su tipo de ataque y utiliza la misma "nomenclatura" que el jugador. Además, su vida se muestra en color azul claro para representar que tiene un "escudo de energía" que lo hace invulnerable. Para perder sus escudos (representado por triángulos) tenemos que darle un golpe al núcleo y cuando éste ha perdido todos ellos, su vida pasa a ser de color rojo para transmitir que ahora es vulnerable y su vida puede bajar.

Una novedad es que cuando el *spinEnemy* pierde el 50% de su vitalidad, éste cambia totalmente a color rojo y se debilita, por lo que ataca más lento y empieza a huir del jugador. Cuando está a una distancia adecuada, deja de disparar y empieza a regenerarse. En esta fase, si el jugador entra en su zona, si aún no se ha recuperado lo suficiente (50%) seguirá huyendo y si tiene al menos la mitad de su vida volverá a ser agresivo abalanzándose sobre el jugador.

En general, el *spinEnemy* es el enemigo con la I.A. más compleja que tenía la versión original de *PolygonSouls* y lo que se ha intentado es que sea más fácil de entender y que tenga más capas de complejidad a la hora de decidir qué hacer. Esto se ha reflejado muy positivamente en la opinión de los jugadores que se expondrá en el apartado de resultados.

Si hablamos de la *TurretEnemy*, ahora su parte frontal es de color azul claro, por lo que es invulnerable por esa cara. Esto nos obliga a tomar una estrategia distinta atacando cuerpo a cuerpo o flanqueándola mientras recarga, ya que ahora se queda parada mientras no pueda atacar. Después, cuando está preparada para disparar es cuando actualiza su giro y apunta al jugador.

Estos cambios no son muy profundos, pero consiguen hacer que el enemigo tenga una identidad propia y genere situaciones más interesantes. En la versión inicial, eran amenazas muy poco importantes y que eliminabas sin mucho esfuerzo, pero ahora te obliga a que les prestes más atención.

Sobre el *KamikazeEnemy*, en esta nueva versión, tiene que recibir tres golpes en lugar de uno para ser destruido y por cada ataque se vuelve más rápido y grande. Además, se le ha creado un nuevo tipo de ataque único que no respeta el triángulo de tipos para siempre haga un gran daño. Esto se ha hecho porque antes era un enemigo muy poco amenazante que casi nunca alcanzaba al jugador por ser demasiado lento y si lo lograba es posible que solamente le hiciera 10 puntos de daño.

El *kamikazeEnemy* es el enemigo que visualmente ha cambiado más. Se han potenciado algunas de sus mecánicas para que sea un enemigo más relevante, pero su esencia se ha mantenido igual. Como detalle, aunque su velocidad y tamaño aumenten, el tamaño de su hitbox no varía. Esto hace que el peligro sea casi el mismo, pero el jugador siente una mayor tensión, algo que se ha visto reflejado en los resultados.

Por último, hemos de hablar del *bouncerEnemy*, el nuevo enemigo que se ha incorporado al plantel. En cuanto a movimiento, contábamos un enemigo inmóvil y dos que principalmente iban hacia el jugador. Por ello, se buscó crear una nueva amenaza que cambiase un poco las reglas del juego.

Debido a esto, se creó este enemigo que no va hacia el jugador, sino que tiene un patrón de movimiento simple y fácil de entender. Sin embargo, su complejidad no reside en cómo se mueve, sino en cómo se relacionan sus movimientos respecto a otros enemigos. Además, sus disparos se anulan con los del jugador, lo cual provoca bastantes situaciones inesperadas.

El *bouncerEnemy* ataca en las cuatro direcciones cardinales siempre y cuando detecte que el jugador ha entrado en sus radios de vigilancia. Esto hace que el enemigo no sea un gran peligro por sí solo, pero sí que, al aparecer junto a otras amenazas, pueda generar situaciones donde involuntariamente corta rutas de escape al jugador, aumentando así la tensión y la dificultad de forma orgánica. Esto es algo que algunos jugadores de prueba han sido capaces de percibir y es algo que mejora notablemente la complejidad y diversión del juego.

En líneas generales, todos los enemigos han recibido cambios, nuevas características y se ha buscado que sus características sean bastante más identificables. Algunas mejoras menores no se han recogido, pero más allá de las descritas aquí, todos los enemigos han sido revisados para que se comporten o reaccionen distintos según cómo afectaba al "game feel" de *PolygonSouls 2.0* y/o respecto al *feedback* de los jugadores.

3.3 Mejoras del jefe final

Como resumen, el jefe final ha sido totalmente trabajado desde cero y de su versión inicial no queda nada. El antiguo enemigo era artificialmente difícil, injusto y demasiado simple como para ser considerado la culminación de la aventura. Por ello, se ha escogido una filosofía totalmente opuesta para poder representar que él es el clímax de la mazmorra, además de que sea un reto que te exija poner a prueba todas las mecánicas que ofrece *PolygonSouls 2.0*.

Lo primero que se ha hecho es cambiarle el *Sprite* para darle una temática concreta. Y para hacerlo más carismático se le ha creado diferentes animaciones, se le

han creado líneas de diálogo nuevas y nuevos ataques tal como se puede ver en las [Figuras 5a y 5b](#). Asimismo, sus ataques también han sido cambiados para que tengan forma de hueso para que tenga coherencia estética.

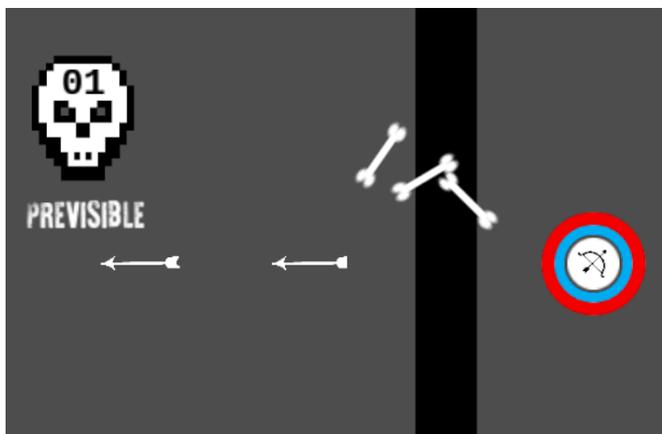


Figura 5a y b: Distintas fases del Jefe Final de PolygonSouls 2.0.

Después, respecto al combate contra, se han diseñado tres fases diferenciadas que obligarán al jugador a utilizar sus tres tipos de ataque.

En la primera, tendremos que atacar cuerpo a cuerpo haciendo un guiño al combate original. En la segunda recurriremos a los veloces ataques mágicos. Y en la última fase necesitaremos de los longevos ataques de rango para golpear al jefe. Para explicar todo esto de forma natural, si el jugador no está haciendo bien las mecánicas propuestas, el jefe hace algunos comentarios que sirven como pista.

Visualmente, el jefe de la mazmorra sigue las mismas reglas respecto a los colores. Comienza la batalla con un escudo azul representando la invulnerabilidad, en la segunda fase el escudo pasa a ser rojo porque está desgastado y en la tercera fase desaparece haciendo referencia a que estaría totalmente expuesto y por ello busca distanciarse de ti.

El reto ahora es bastante más satisfactorio, divertido, desafiante y justo y representa mejor lo que debería ser el último combate de un videojuego. Anteriormente era un enemigo sin personalidad, ni sentido, mientras que ahora se le ha dado cierta personalidad y trasfondo. En general,

los pocos jugadores que han probado la última versión del juego han valorado muy positivamente este cambio.

3.4 Mejoras generales y de calidad de vida

Para mejorar el “quality of life” de PolygonSouls 2.0, se han realizado algunos cambios que desafían en mayor o menor medida la filosofía inicial del título para que la experiencia de juego mejore.

Lo primero que podemos ver a simple vista es el cambio de la sala inicial, ya que ahora es bastante más intuitiva, clara y explicativa como podemos notar en las [Figuras 6a y 6b](#). Ahora muestra mejor las acciones, las teclas que están disponibles y además hay soporte para mandos de Xbox [7]. Además, cabe destacar que se han cambiado las etiquetas “C-R-M” por Sprites que representen las mismas ideas, pero de una forma más explícita.

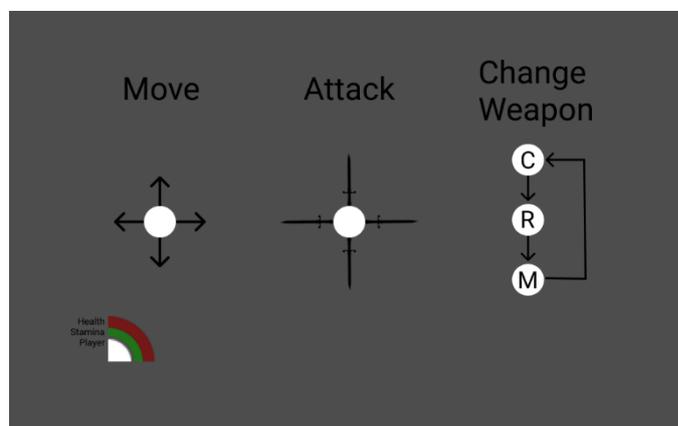


Figura 6a y b: Aspecto de la sala inicial original y sala inicial nueva con soporte para Xbox.

Sobre estos ataques, se ha cambiado el Sprite del ataque Rang para que sea una flecha blanca en lugar de una espada, algo bastante más claro para el jugador. Y por su lado, el ataque Mágic tiene un nuevo Sprite con aspecto de llama.

En esta misma línea, para diferenciar más estos dos tipos de ataques, se ha optado por cambiar su comportamiento y estadísticas. El ataque a distancia tiene rango infinito y una buena cadencia, mientras que el ataque mágico tiene una cadencia superior a cambio de tener menos rango y gastar el doble de energía por disparo.

Otro cambio visual importante ha sido la selección de colores. Como ya hemos indicado previamente, el azul se relaciona con la invulnerabilidad/energía, el rojo con la vulnerabilidad y estos son la mayoría de los colores que hay en *PolygonSouls 2.0*. Por otro lado, también está el blanco, el cual conforma las estructuras y algunos cuerpos, o el amarillo y negro que aparecen en algunos enemigos.

Cabe destacar que se ha intentado adaptar el juego para personas daltónicas [6], pero no se ha encontrado una solución sencilla aplicable en *Godot*. A raíz de esto, simplemente se ha evitado el uso del verde para facilitar la accesibilidad a las personas daltónicas de tipo "Protanopía", ya que el tutor del TFG tiene este tipo de daltonismo.

Como último cambio visual, se ha hecho que la interfaz sea bastante más entendible al cambiar cosas como "S" de "Souls" a "Almas" o la implementación de nuevos indicadores visuales basados en colores (Fig. 7). Como detalle, se ha traducido *PolygonSouls 2.0* para que todo esté en español de cara al jugador.



Figura 7: La interfaz es más explícita y cambia de color si sabemos cuál es el objetivo para entrar a la Cripta.

Respecto a las mecánicas de juego, el cambio más importante y notable es que al entrar a las salas, los enemigos están parados un breve momento. Esto ayuda a que el juego se sienta más justo, puesto que el jugador puede identificar más fácilmente a los enemigos antes de que estos empiecen a moverse y a dispararle.

Asimismo, el juego cuenta ahora con ataques en diagonal, una característica muy común en este tipo de títulos. Y para mejorar la sensación de juego, se ha reducido el tiempo que tenemos que esperar para que la energía empiece a regenerarse.

Por otro lado, se ha reducido el daño base de los ataques, ya que hacían veinte (20) puntos de daño, mientras que ahora hacen diecinueve (19). Esto provoca que muchas veces se generen sensaciones fuertes en el jugador al estar a poca vida, lo cual se traduce en momentos más memorables y potencialmente mayor diversión.

Un cambio muy sencillo, pero a la vez muy necesario ha sido la implementación de los botones de acción "F" y

"G". Esto se ha agregado para no tener que pulsar los botones de ataque en las hogueras y en la entrada al jefe final, ya que se podía crear otro botón.

Finalmente, la última de las mejoras a destacar son las dimensiones de las distintas hitbox, tanto del jugador como de enemigos. En líneas generales, el jugador y los proyectiles enemigos tienen una hitbox ligeramente más pequeña que sus Sprites para hacer que se eviten más golpes de los que debería y el "game feel" mejore. En la otra cara de la moneda, los proyectiles del jugador y los enemigos tienen una hitbox ligeramente más grande de lo normal para que sea más fácil conectar golpes que parecían que no iba a dar, haciendo así que el jugador se sienta mejor.

Algunas de estas mejoras no están intrínsecamente relacionadas con el proyecto, pero ayudan a mejorar a *PolygonSouls 2.0* como videojuego, algo que ha sido valorado positivamente.

3.5 Problemas/bugs encontrados

A continuación, se comentarán algunos de los problemas y bugs más importantes que se han descubierto a medida que se ha desarrollado *PolygonSouls 2.0*. Algunos se debieron a una mala codificación, mientras que otros eran errores ya existentes en la versión inicial.

Entre ellas destaca un error donde, pese a no poder lanzar ataques *Rang* o *Màgic* debido al limitador de tiempo, la energía se gastaba tantas veces como veces pulsásemos el botón de ataque. En otras palabras, podíamos pulsar muy rápidamente el botón de ataque para gastar cien (100) puntos de energía, pero solo se dispararían cuatro (4) proyectiles. Como se ha indicado arriba, esto se ha arreglado y se ha aprovechado para diferenciar aún más los ataques del jugador.

Otro problema heredado de la versión inicial era que los enemigos volvían a aparecer cada vez que entrábamos a una sala incluso si los habíamos matado previamente. No se sabe si era una característica intencionada, pero se estima que no, ya que había un estado de sala llamado "enemiesKilled", el cual no tendría utilidad si los enemigos revivían.

Los enemigos han sufrido otra serie de problemáticas derivadas de cómo se han implementado las ideas que se querían agregar al juego. Por ejemplo, durante un tiempo, el *SpinEnemy* se hacía daño a sí mismo. En esta misma línea, la *turretEnemy* cuenta con un problema por el que no puede girar bien en 360° debido a su nuevo comportamiento.

Por otro lado, al implementar el algoritmo de asignación de lejanía surgió un error debido a que el código original borraba las salas una vez les asignaba un rol especial. Esto era problemático si la sala del jefe final era la última de la

“lista”, puesto que nos devolvía un error al no existir en memoria.

4. RESULTADOS

Calcular las mejoras que se han aplicado a *PolygonSouls 2.0* de forma objetiva es algo imposible, sobre todo porque el objetivo principal era mejorar la I.A para que el “game feel” mejorase notablemente. Por ello, se ha recopilado la opinión de un grupo de jugadores de prueba, o “betatesters”.

Dichos jugadores han probado la versión original del juego, la intermedia 1.5 y la final 2.0, la cual es la que se ha entregado. Han jugado sin ninguna clase de ayuda o comentario por parte del autor y las sesiones han sido de alrededor de diez (10) minutos por cada versión. Después, han respondido dos formularios: el primero para comparar las dos primeras versiones y el segundo para comparar la versión final con las anteriores.

Las preguntas han sido divididas en varias secciones. La primera parte pide una opinión comparativa entre los enemigos en su versión inicial, en contraparte a su versión final. Mientras que la segunda sección está más enfocada a comentarios generales sobre el juego y las sensaciones que los testers han obtenido al jugar las diferentes iteraciones.

Si nos fijamos en los resultados de los enemigos mostrados, podemos ver cómo la mayoría de los usuarios han opinado que los cambios han sido bastante positivos.

En primer lugar, *spinEnemy* ha recibido una mejora notable, tal como se ve en la [Figura 8](#). Sin embargo, el cambio mejor valorado ha sido el del *kamikazeEnemy*, ya que en la [Figura 9](#) las valoraciones están totalmente invertidas. Por otro lado, en la [Figura 10](#) vemos que *turretEnemy* ha sido quien ha obtenido resultados más discretos, ya que su versión inicial ya era bastante satisfactorio y entendible.

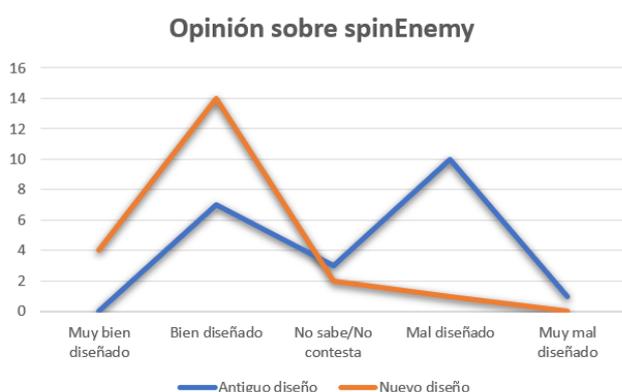


Figura 8: Opiniones sobre *spinEnemy* (más a la izquierda, mejor).

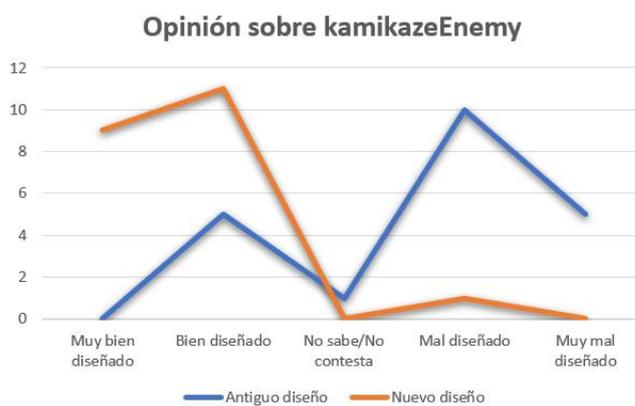


Figura 9: Opiniones sobre *kamikazeEnemy* (más a la izquierda, mejor).

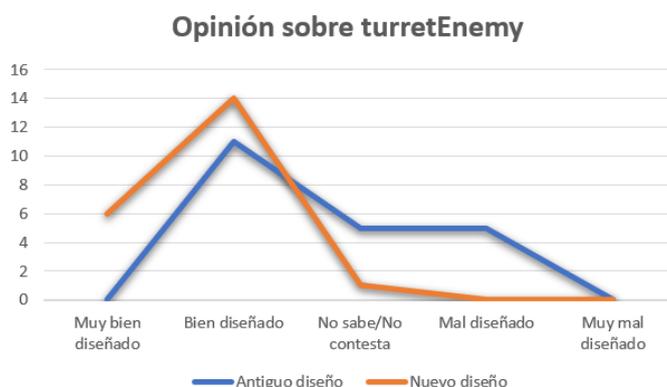


Figura 10: Opiniones sobre *turretEnemy* (más a la izquierda, mejor).

Finalmente, la opinión respecto al nuevo *bouncerEnemy* se puede ver en la [Figura 11](#) y es que es un enemigo bien implementado, sencillo y que aumenta la complejidad de *PolygonSouls* sin ser demasiado complicado.

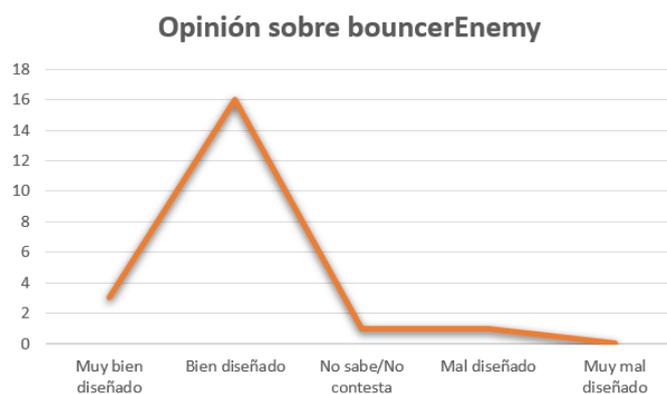


Figura 11: Opiniones sobre *bouncerEnemy* (más a la izquierda, mejor).

Sea como fuere, parece que los usuarios han sentido que, teniendo en cuenta todos los cambios y nuevas mecánicas, *PolygonSouls 2.0* es un juego considerablemente mejor. Esto ha repercutido directamente en la puntuación que los testers le han puesto al título, pasando de una media de “4.95” en la versión 1.0 a una media de “7.38” en la versión 1.5 como podemos ver en la [Figura 12](#). Además, es de recalcar que dicha versión aún no contaba con algunas de las mejoras mecánicas y visuales más importantes que sí están presentes en la versión final 2.0.



Figura 12: Diversión general de ambas versiones (más a la izquierda, mejor).

Un punto importante ha sido el trabajo con la I.U., la selección de colores y la claridad del título. En este aspecto, los jugadores han afirmado que la versión 1.5 de *PolygonSouls* es notablemente más entendible (Fig. 13).



Figura 13: Comparativa respecto a la claridad de ambas versiones del juego (más a la izquierda, mejor).

Si tenemos en cuenta todos los apartados mencionados previamente, el aumento de la complejidad del comportamiento de los enemigos y la inclusión de nuevas mecánicas hace que *PolygonSouls* sea un juego objetivamente más difícil. No obstante, los cambios también han hecho que sea un juego más accesible y comprensible, por lo que el jugador medio se siente menos perdido y abrumado. Esto ha provocado que la dificultad se sienta más equilibrada y que los testers hayan estimado que el juego ahora está mejor diseñado y es más justo, tal como podemos ver en la Figura 14.

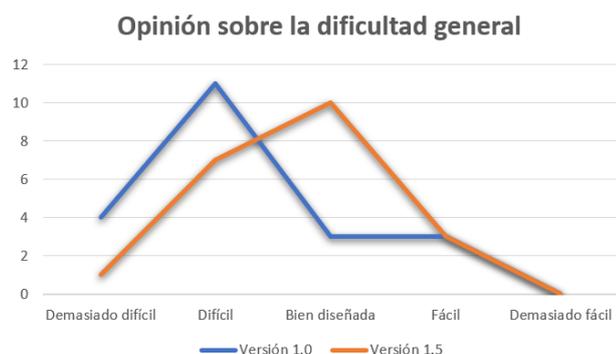


Figura 14: Opiniones sobre la dificultad general del juego (más céntrico, mejor).

En el formulario propuesto a los *betatesters* se han realizado algunas cuestiones extra sobre los cambios aplicados al juego, pero cuya información no es tan relevante, por lo que dichas opiniones se han incluido en el Anexo.

En líneas generales, los resultados son bastante positivos, aunque hay que destacar que la cantidad de usuarios que han dado *feedback* es baja, además de que es sobre una versión no final. Aun así, *PolygonSouls* ha seguido evolucionando después de que los usuarios probasen la versión 1.5 y muchos de sus comentarios han ayudado a desarrollar un mejor juego.

Para ver una versión final de *PolygonSouls* 2.0, se ha creado un vídeo donde se muestran un *gameplay* completo [13] con la mayoría de las mecánicas y características que nos puede ofrecer el juego. Y si quieres conocer el código en profundidad, se puede explorar en el enlace al GitHub [14].

Actualmente, no se ha podido recopilar información representativa de los cambios aplicados a la versión 2.0, pero hay que hacer un especial hincapié en que se han pulido aspectos estéticos, de calidad de vida y que la mayor parte del esfuerzo se ha ido al desarrollo de la batalla final contra el jefe.

Respecto al enfrentamiento contra el jefe final, los pocos comentarios recibidos han sido bastante positivos. La batalla original era poco satisfactoria, debido a que el jefe se sentía injusto y demasiado simple. En cambio, ahora es más carismático e interesante y tiene fases diferencias que nos obligan a utilizar todas las habilidades disponibles, por lo que realmente se siente como una pelea contra un enemigo exigente y peligroso.

5. CONCLUSIONES

Actualmente *PolygonSouls* se encuentra en un estado jugable, pero no se podría ver como un producto preparado para la venta. Podríamos considerarlo una versión Alpha o Beta, es decir, funcional, pero todavía requeriría de bastante trabajo por delante.

Pese a ello, los resultados que hemos obtenido del proyecto parecen ser muy buenos, ya que los jugadores que han probado las distintas versiones de *PolygonSouls* han dado opiniones muy positivas sobre las mejoras. La estadística objetiva más evidente que podemos coger es la nota media (Fig. 9), aunque todos los valores recopilados muestran unas conclusiones similares.

Dado que nuestro objetivo inicial era mejorar la experiencia de usuario mediante distintas estrategias (mejorar la I.A., el algoritmo de generación de niveles, entre otros), podemos concluir que el proyecto ha cumplido satisfactoriamente con su propósito. No obstante, cabe destacar que algunos de los cambios más

notados por los usuarios han sido los que tenían que ver con objetivos secundarios como el balance del juego, la mejora de la I.U. o los cambios de Sprites.

5.1 Mejoras y futuro del proyecto

PolygonSouls sigue siendo un juego bastante incompleto, por lo que, de cara al futuro, una de las primeras cosas que se buscaría implementar es una banda sonora y efectos de sonido, ya que el título carece de ellos en su totalidad. Es algo que haría que el juego mejorase bastante, pero que también requiere de un esfuerzo y conocimientos que actualmente están fuera de alcance.

Algo a destacar del proyecto es que actualmente ha sido programado pensando solamente en las funcionalidades, por lo que no se ha buscado ser eficiente a la hora de codificar. Hay funciones redundantes, variables que sobran, incontables warnings que aparecen en consola y otros problemas que requerirían de atención.

También cabe mencionar la ausencia de una coherencia artística que sirva como espina dorsal del juego. La mayoría de los enemigos tienen diseños simples y carecen de personalidad. Por su lado los fondos, paredes y proyectiles no siguen una misma temática, puesto que unas cosas tienen una estética poligonal simple, mientras que otras parecen tener aspecto medieval. Por ello, algo que se debería trabajar en el futuro sería establecer una temática que abarque todo el juego

Si pasamos a un nivel más personal, en mi opinión este proyecto ha sido realmente interesante. He ganado experiencia, ha sido intelectualmente nutritivo y me lo he pasado bien desarrollándolo. Sin embargo, también tengo que reconocer que me ha sido bastante complicado debido a que no sé trabajar bien con *Godot* y *GDScript*, razón por la que he estado continuamente buscando soluciones en la documentación y en los foros.

Actualmente, me desenvuelvo bastante mejor con las herramientas gracias a algunas habilidades que he descubierto y me gustaría realizar algunos cambios importantes, pero carezco de tiempo para implementarlas. Sea como fuere, me gustaría continuar desarrollando *PolygonSouls* o, en su defecto, comenzar con algún proyecto de videojuego propio, aunque este segundo posiblemente sería en Unity o Unreal Engine.

AGRADECIMIENTOS

En primer lugar, querría agradecer a A. J. P. Dedeu, el creador original de *PolygonSouls*. Su videojuego me ha permitido trabajar en el proyecto propuse.

En esta misma línea, querría dar mis más sinceros agradecimientos a Jorge Bernal, mi tutor del Trabajo de Fin de Grado, ya que ha sido quien me ofreció la oportunidad de trabajar en este proyecto, así como quien me dio el

acceso al código original. Además, pese a no poder ofrecerme ayuda directa con el entorno o el lenguaje, me ayudó con comentarios, guías de cómo afrontar los distintos retos y, sobre todo, sobre qué objetivos trabajar.

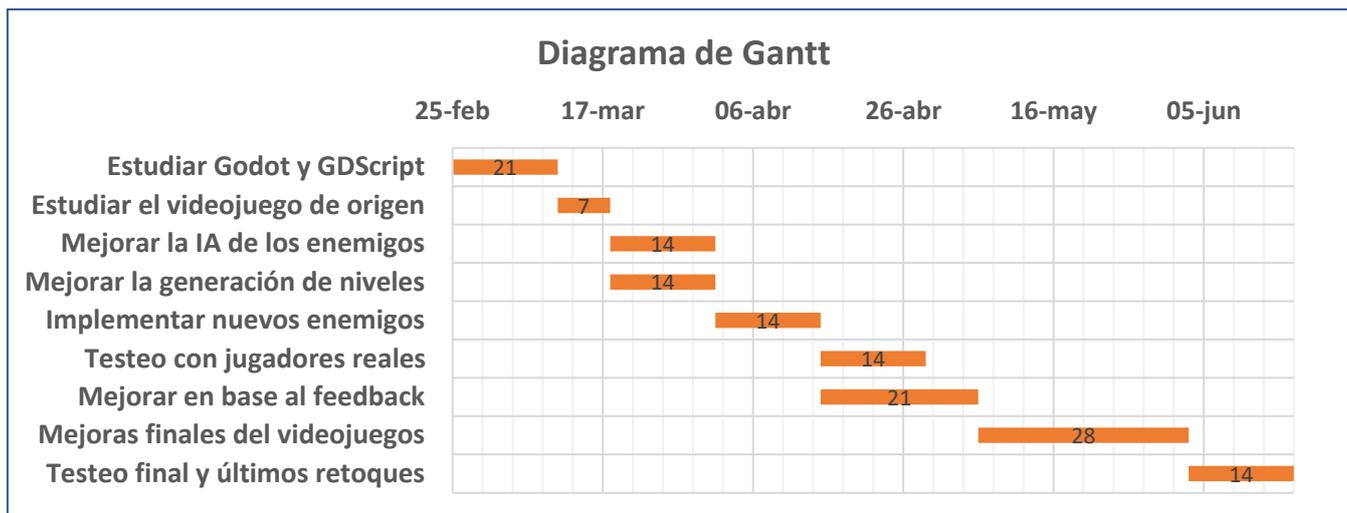
Por otro lado, me gustaría agradecerle a mi compañera de prácticas de *GIS (Garantia de l'Informació i Seguretat)*, Maria Tomàs, ya que este semestre solamente he cursado esta asignatura y ella ha sido muy flexible a la hora de acordar momentos para hacer los trabajos y estudiar.

También querría darle las gracias a todos los jugadores que han probado mi juego siendo mis *betatesters* al reportarme bugs y recomendarme cambios en el juego. Más de veinte (20) personas de distintas partes de España me han dado opinión respecto a las versiones 1.5 y 2.0 de *PolygonSouls* y me han permitido trabajar en ideas que a mí nunca se me habrían ocurrido.

6. BIBLIOGRAFÍA

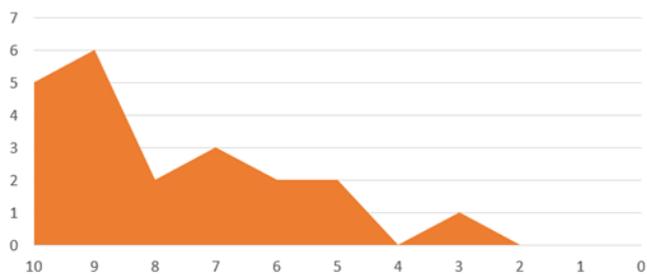
- [1] Godot Docs. (2022, Mar. 1). Documentación de Godot [Online]. Disponible: <https://docs.godotengine.org/es/stable/index.html>
- [2] A. Céspedes. (2018, Oct. 22). Como crear un juego simple 2D en Godot [Online]. Disponible: https://www.youtube.com/watch?v=vFbkF0JhSul&list=PL5K_XeiglfdJeaJ3-YgnzD711StmJkZh&index=1
- [3] GrandMasterTaco. (2021, Ene. 9). DIY Procedural Generation - Quick Python [Online]. Disponible: <https://www.youtube.com/watch?v=NUXHw3DBog8>
- [4] C. Bradfield. *Godot Engine Game Development Projects*. Los Ángeles: Packt Publishing, 2018
- [5] A. J. P. Dedeu, creador de *PolygonSouls* (comunicación privada), abril 2022.
- [6] Alan Zucconi (2022, May. 25). Accessibility Design: Color Blindness [Online]. Disponible: <https://www.alanzucconi.com/2015/12/16/color-blindness/>
- [7] Games From Scratch (2022, Jun. 2) Godot 3 Tutorial- Keyboard, Mouse and Joystick Input [Online]. Disponible: <https://gamefromscratch.com/godot-3-tutorial-keyboard-mouse-and-joystick-input/>
- [8] Desmos. Calculadora gráfica (2022, Jun. 4) Herramienta para probar los valores del algoritmo de creación de enemigos [Online]. Disponible: <https://www.desmos.com/calculator/rnzp7oh8pb>
- [9] shortcut (2022, May. 12) Link de invitación a Kanban. [Online] Disponible: <https://app.shortcut.com/invite-link/62448523-6eaf-45a0-ba85-ee5fa13734a4>
- [10] Listado de mejoras (2022, Abr. 25) Link de invitación a Google Docs [Online] Disponible: <https://docs.google.com/document/d/1Zr1YFTdgPEjxiVM4G2-4G-PA5UObzQkqv-6FuVDSvw/edit?usp=sharing>
- [11] Kanbanize (2022, Jun. 26) Qué es Kanban: Definición, Características y Ventajas. [Online] Disponible: <https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-kanban>
- [12] A. J. P. Dedeu – Dipòsit Digital de Documents de la UAB (2021) *PolygonSouls: Un joc d'acció amb vista zenital i esperit souls-like* [Online] Disponible: <https://ddd.uab.cat/record/238450?ln=ca>
- [13] Xiang Lin (2022, Jun. 27) *PolygonSouls 2.0 Gameplay* [Online] Disponible: <https://youtu.be/c0QkIJYZGZo>
- [14] Xiang Lin (2022, Jun. 26) Github de *PolygonSouls 2.0* [Online] Disponible: <https://github.com/Lolmaniaco/PolygonSouls-2.0-feature-and-AI-analysis-TFG>

Anexo



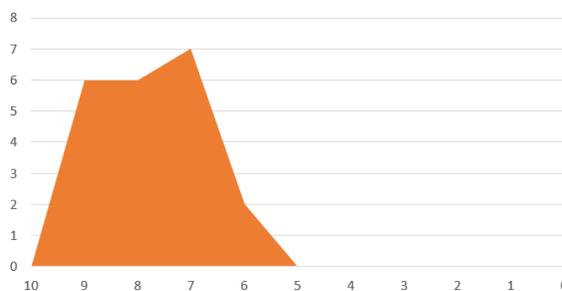
Anexo 1: Diagrama de Gantt

¿Cómo valorarías el cambio en la interfaz y en los colores del juego?



Anexo 2: Valoración de los usuarios a los cambios de la Interfaz de Usuario de la versión 1.5

¿Cuánto crees que ha mejorado el juego respecto a la versión original?



Anexo 4: Puntuación de los usuarios a los cambios aplicados en la versión 1.5

¿Crees que las nuevas mecánicas de los enemigos hacen al juego más divertido/entretenido?



- Sí, están bien implementadas
- Sí, pero habría que explicarlo por escrito
- No (con diferentes matices)

Anexo 3: Opinión de los usuarios respecto a los cambios aplicados en la versión 1.5