



**BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA**  
**ESCUELA DE ARTES PLÁSTICAS Y AUDIOVISUALES**  
**ARTE DIGITAL**



**Diseño y desarrollo del videojuego "The Fall of the last guardian"**  
**empleando como premisa el trastorno de ansiedad y pánico**

Abril 2023

**TESINA**

presentada para obtener el grado de  
**LICENCIADO EN ARTE DIGITAL**

**PRESENTA:**

Yoltic Seiki Camacho Figueroa

Director de Tesina: Mtro. Abner Quiroz Clemente  
Revisor de Tesina: Mtro. Hugo Enrique Rojas Cisneros  
Revisor de Tesina: Mtro. Héctor Adrián Díaz Furlong



Asunto: Aval del Jurado

Mtro. Héctor Adrián Díaz Furlong  
Secretario Académico  
Escuela de Artes Plásticas y Audiovisuales  
Benemérita Universidad Autónoma de Puebla  
P R E S E N T E

Quienes suscriben, integrantes del Jurado del proyecto de titulación por tesina "Diseño y desarrollo del videojuego "The Fall of the last guardian" empleando como premisa el trastorno de ansiedad y pánico" desarrollado por el pasante de la Licenciatura en Arte Digital, Yoltic Seiki Camacho Figueroa, con matrícula 201750095, después de haber revisado su trabajo final, emitimos un dictamen FAVORABLE para que presente su Examen Profesional el día 2 de junio del 2023 a las 9:30 horas, considerando los siguientes requerimientos: *Enlistar el equipamiento requerido, así como las características y capacidad del espacio, etc.*  
Espacio: Sala Chaplin

Equipo: Equipo de cómputo con teclado y mouse, proyector y bocinas.

Sin otro asunto que tratar, agradecemos su atención.

Atentamente  
"Pensar bien, para vivir mejor"  
H. Puebla de 24 de abril del 2023

Mtro. Hugo Enrique Rojas Cisneros  
Presidente  
hugo.rojascis@correo.buap.mx

Mtro. Héctor Adrián Díaz Furlong  
Secretario  
hector.diazfurlong@correo.buap.mx

Mtro. Abner Quiroz Clemente  
Vocal  
abner.quiroz@correo.buap.mx

1. c. c. p. Dr. Hugo Enrique Rojas Cisneros/ARPA/BUAP
2. c. c. p. Dra. Elsa Herrera Bautista, Coordinadora de Evaluación Académica /ARPA/BUAP
3. c. c. p. Yoltic Seiki Camacho Figueroa, pasante del Colegio de Arte Digital/ARPA/BUAP



2.29.55.99 ext 2379



www.arpa.buap.mx



Edificio Sur 1, Primer Piso Complejo Cultural  
Universitario Boulevard Atlixcáyotl 2499 Puebla,  
Puebla México

## Índice

|        |                                                                                         |    |
|--------|-----------------------------------------------------------------------------------------|----|
| 1.     | Introducción.....                                                                       | 6  |
| 2.     | Objetivos.....                                                                          | 9  |
| 2.1.   | Objetivo General.....                                                                   | 9  |
| 2.2.   | Objetivos Específicos.....                                                              | 9  |
| 3.     | Planteamiento del Problema.....                                                         | 10 |
| 4.     | Antecedentes .....                                                                      | 11 |
| 4.1.   | Personajes del Cine, Televisión, o Cómic que Padecen o Experimentaron Ataques de Pánico | 11 |
| 4.1.1. | Jessica Cruz.....                                                                       | 11 |
| 4.1.2. | Tony Stark, Iron Man 3 (2013) .....                                                     | 11 |
| 4.1.3. | Randall Pearson, This is Us (2016-2022).....                                            | 12 |
| 4.2.   | Videojuegos que Emplean Mecánicas de Juego Similares .....                              | 13 |
| 4.2.1. | Stick Fight: Shadow Warrior & Stickman Game.....                                        | 13 |
| 4.2.2. | Celeste .....                                                                           | 13 |
| 4.2.3. | Hollow Kinght .....                                                                     | 14 |
| 5.     | Marco Teórico .....                                                                     | 15 |
| 5.1.   | Arte Digital.....                                                                       | 15 |
| 5.2.   | Salud Mental.....                                                                       | 15 |
| 5.3.   | Trastornos de Ansiedad.....                                                             | 16 |
| 5.3.1. | Estrés y Ansiedad .....                                                                 | 17 |
| 5.3.2. | Trastorno de Pánico .....                                                               | 18 |
| 5.4.   | Diseño y Desarrollo del Videojuego.....                                                 | 20 |
| 5.4.1. | Documento de Diseño de Juego .....                                                      | 20 |
| 5.4.2. | Diseño de Videojuegos .....                                                             | 20 |
| 5.4.3. | Desarrollo de Videojuegos.....                                                          | 21 |
| 5.4.4. | Diseño de Niveles .....                                                                 | 21 |

|        |                                                 |    |
|--------|-------------------------------------------------|----|
| 5.4.5. | Videojuegos de Acción.....                      | 22 |
| 5.5.   | Programación.....                               | 22 |
| 5.5.1. | Máquina de Estados Finita (FSM).....            | 22 |
| 5.6.   | Narrativa (Storytelling).....                   | 23 |
| 5.6.1. | Narrativa.....                                  | 23 |
| 5.6.2. | Diseño de Narrativa.....                        | 24 |
| 5.7.   | Arte.....                                       | 24 |
| 5.7.1. | Diseño de Personajes.....                       | 24 |
| 5.7.2. | Teoría del Color.....                           | 25 |
| 5.7.3. | Arte de Pixel (Pixel art).....                  | 26 |
| 5.8.   | Herramientas de Desarrollo.....                 | 26 |
| 5.8.1. | Unity.....                                      | 26 |
| 5.8.2. | Visual Studio.....                              | 28 |
| 5.8.3. | Krita.....                                      | 28 |
| 5.8.4. | Aseprite.....                                   | 28 |
| 6.     | Marco Metodológico.....                         | 30 |
| 6.1.   | Preproducción.....                              | 30 |
| 6.2.   | Producción.....                                 | 30 |
| 7.     | Procesos.....                                   | 32 |
| 7.1.   | Preproducción.....                              | 32 |
| 7.1.1. | Diseño de Videojuego.....                       | 32 |
| 7.1.2. | Diseño de Nivel.....                            | 50 |
| 7.2.   | Producción.....                                 | 57 |
| 7.2.1. | Programación.....                               | 57 |
| 7.2.2. | Desarrollo del Apartado Visual y Animación..... | 80 |
| 7.2.3. | Música y Sonido.....                            | 98 |

|                                                                  |     |
|------------------------------------------------------------------|-----|
| 7.3. Publicación.....                                            | 99  |
| 8. Resultados y conclusiones .....                               | 102 |
| Bibliografía.....                                                | 104 |
| Anexos.....                                                      | 108 |
| Anexo A. Índice de figuras.....                                  | 108 |
| Anexo B. Códigos C# .....                                        | 110 |
| Mover un objeto de un punto a otro .....                         | 110 |
| Guardar la información de colisión de un Physics2D.Raycast ..... | 110 |
| Visualizar en el editor un Raycast.....                          | 110 |
| Clase base de los diferentes estados.....                        | 110 |
| Estructura base para definición de estados de jugador.....       | 110 |
| Clase PlayerStateManager .....                                   | 111 |
| Clase StateController .....                                      | 112 |
| Clase Action .....                                               | 112 |
| Clase Desicion.....                                              | 113 |
| Clase Transition.....                                            | 113 |
| Clase State .....                                                | 113 |

## 1. Introducción

A mediados de 2021, Bach (2021) publicó en el centro de noticias de Microsoft un artículo, acerca de la importancia que tiene los juegos en la salud mental, donde se registraron vivencias personales, experiencias de profesionales en el área de la salud y de desarrolladores de videojuegos, también se habla acerca del valor que tienen los videojuegos para ayudar a las personas a sobrellevar los problemas de la vida en general, para su uso clínico en el tratamiento de enfermedades de salud mental, o para brindar experiencias.

Retomando el último punto, uno de los temas más interesantes en la publicación fue el proceso de los creativos, y las consideraciones que se deben tener en cuenta al desarrollar videojuegos que aborden temas delicados como lo son los de la salud mental, ya que una representación estereotipada o mal dirigida hacia el público puede ocasionar un daño no solo en la percepción general de estas enfermedades, sino que puede repercutir directamente en la salud de alguien que las padece. Por ello, los desarrolladores detrás de estos videojuegos tienen y asumen la responsabilidad de investigar, recopilar información, trabajar con especialistas en el área y comunicarse directamente con personas que padecen estas enfermedades, para así brindar desde las mecánicas de juego, la narrativa o el apartado visual de un videojuego, una experiencia real y consciente de los problemas que estas personas afrontan, en su lucha diaria para sobrellevar y mejorar su salud mental.

En los últimos años la industria de los videojuegos ha evolucionado a un ritmo acelerado, desde las mejoras de software y hardware para desarrolladores como consumidores, hasta una gran oferta de videojuegos en múltiples plataformas. Y dentro de este gran crecimiento, los videojuegos independientes han logrado posicionarse como una alternativa llamativa e interesante frente a los videojuegos AAA, los cuales son producidos y distribuidos por una importante distribuidora.

Con base en la publicación de Graeme Bruce, periodista de datos empresariales de YouGov:

*Todos los géneros de juegos experimentaron una disminución de jugadores año tras año, excepto uno. Nuestros datos muestran que un 31 % más de jugadores de PC y consolas jugaron juegos independientes el año pasado, en comparación con 2021, ya que los jugadores miraron más allá de las ofertas típicas de AAA.*

*Los desarrolladores independientes tienen la capacidad de crear juegos de nicho que atiendan a audiencias más diversas. Títulos como Death's Door, Stardew Valley, Celeste y Among Us han ayudado a amplificar el género indie en los últimos años. (Bruce, 2022)*

Los videojuegos independientes tienen un papel muy importante dentro de la industria, ya que el incremento en la producción e interés del público en estos, es debido a que son los desarrolladores detrás de estos proyectos los que deciden tomar riesgos para innovar tanto en narrativas, estilos visuales y mecánicas de juego; volviendo estas características distintivas de lo que se conoce como el género Indie (Independiente), en contraposición con los videojuegos AAA de estudios grandes, que ofrecen también experiencias de juego llamativas e interesantes. Estos difícilmente salen de una formula preestablecida que les garantiza seguridad financiera al momento de lanzar al mercado sus videojuegos, bajo el riesgo de mermar no solo la calidad del producto, sino de abusar de la confianza que los jugadores depositan al comprar sus videojuegos.

Laura Parker publicó en 2019 para The New York Times lo siguiente:

*El año pasado, un juego llamado Celeste exploraba la depresión y la ansiedad a través de un protagonista que tenía que sortear obstáculos físicos y emocionales. En el videojuego de fantasía, acción y aventuras Hellblade: Senua's Sacrifice de 2017, una joven guerrera celta lidia con la psicosis.*

*Otros juegos en los últimos años, incluidos Night in the Woods y Pry, han profundizado en la identidad propia, los problemas de ira y el trastorno de estrés postraumático.*

*Hasta la fecha, la mayoría de los juegos que abordan la salud mental provienen de creadores independientes, que generalmente están más dispuestos y son más capaces de asumir riesgos al explorar temas inusuales. (Parker, 2019)*

Will Harris (2021) enlista quince videojuegos que abordan las enfermedades mentales: Aether, Lost Words: Beyond The Page, We Happy Few, Life Is Strange: Before The Storm, Gris, Child Of Light, Celeste, Night In The Woods, The Last Of Us, Wolfenstein 2: The New Colossus, Limbo, Persona 5, Dark Souls y Hellblade: Senua's Sacrifice. La mayoría de estos juegos fueron desarrollados

por pequeños estudios independientes, y muy pocos por grandes estudios, además de ser una cantidad baja de videojuegos en comparación con todos los que salen cada año en el mercado.

Algunos de estos videojuegos toman un problema de salud mental y se diseña y desarrolla el juego alrededor de este. No todos los videojuegos abordan su temática de la misma forma, y por ende, ninguno ofrece la misma experiencia a pesar de poder tomar el mismo tema; unos pueden ser un poco más directos y concisos como Hellblade al abordar la psicosis dentro de un mundo inspirado en la mitología nórdica y celta, pero otros no transmiten su mensaje de manera tan directa, sino que el videojuego incentiva a reflexionar y pensar sobre lo que se está jugando, como lo son juegos como Limbo, Celeste, Gris, Night In The Woods u Omori; pero a pesar de estas diferencias, la mayoría de estos videojuegos comparten un objetivo en común, y es el de transmitir los problemas y afecciones reales que viven las personas que padecen alguna enfermedad de salud mental.

Sin embargo, dentro de los videojuegos ya mencionados que abordan la salud mental, son pocos los que toman como tema central el trastorno de pánico, y si bien hay algunos juegos que retoman la ansiedad como tema central y logran reflejar desde el diseño y desarrollo del juego lo que es vivir con ella, no muchos se centran en los ataques de pánico. Celeste es uno de los pocos juegos que plasma en determinado punto de la historia como el personaje principal sufre de un ataque de pánico, y es mediante las mecánicas de juego y la narrativa que es posible vivir la experiencia del personaje. El trastorno de pánico como tema central en los videojuegos es un área que aún no ha sido explorada en su plenitud, y es una oportunidad que los desarrolladores aún no han aprovechado, por lo que es conveniente el desarrollo de videojuegos en los que se tome esta temática y se explore las posibilidades que los videojuegos pueden ofrecer para representar este tipo de trastornos en este medio

El presente proyecto pretende contribuir al catálogo de videojuegos existentes que abordan la salud mental, mediante el diseño, desarrollo y publicación del prototipo de un videojuego 2D (dos dimensiones) que tome el trastorno de pánico para definir las mecánicas de juego.



## **2. Objetivos**

### **2.1. Objetivo General**

- Diseñar, desarrollar y publicar el prototipo de un videojuego de acción en la página Itch.io, que tome como base el trastorno de pánico para definir las mecánicas de juego

### **2.2. Objetivos Específicos**

- Elaborar el documento de diseño de juegos.
- Realizar el prototipo del videojuego utilizando el motor de videojuegos Unity.
- Publicar el prototipo del videojuego en la página de distribución de videojuegos Itch.io.

### 3. Planteamiento del Problema

Con base en la página Our World Data *“Para 2017 este estudio estima que 792 millones de personas vivían con un trastorno de salud mental. Esto es un poco más de una de cada diez personas en todo el mundo (10,7%)”* (Dattani et al., 2021)

Los trastornos de ansiedad fueron uno de los problemas de salud mental que afectaron a más personas en 2017 con un total de 3,8% de la población mundial (284 millones de personas), seguido de la depresión con un 3,4% (264 millones de personas) (Dattani et al., 2021).

Actualmente el sector de los videojuegos se encuentra en un auge tanto de innovación y desarrollo, como de atención e interés, esto gracias a la mejora constante de motores de desarrollo como Unity y Unreal 5, eventos enfocados para desarrolladores como la Game Developers Conference, o The Game Awards, gala para premiar a los mejores juegos del año en diversas categorías, y el interés de las más grandes distribuidoras de videojuegos, Microsoft y Sony, para ofrecer productos y servicios que beneficien tanto a desarrolladores como consumidores y consolidar aún más la industria de los videojuegos.

Dentro de todo este mercado en crecimiento, los desarrolladores independientes o estudios en crecimiento son los que deciden tomar un camino diferente a los videojuegos AAA, y prefieren buscar nuevas narrativas que puedan aprovechar para transmitir un mensaje reflexivo, emocional o emocionante. Uno de los temas más importantes abordados por esta nueva ola de videojuegos, son los de la salud mental, en donde encontramos juegos que hablan sobre la depresión, la ansiedad, el duelo, traumas, entre otros; pero ninguno toma el trastorno de pánico como tema central. Además, este tipo de proyectos en general son escasos dentro del mercado por la misma razón de que son desarrollados por estudios independientes, y al existir una falta de videojuegos que aborden el tema de los trastornos de pánico, se está desaprovechando la oportunidad de crear un videojuego que visibilice a través de las mecánicas de juego esta enfermedad y así poder brindar una experiencia que sensibilice a las personas que desconozcan o vivan con este tipo de problema de salud.

Por estas razones, y empleando los conocimientos adquiridos durante la licenciatura en Arte Digital, se decidió diseñar, desarrollar y publicar el prototipo de un videojuego en Itch.io, que toma como base el trastorno de pánico para definir las mecánicas de juego, teniendo como público objetivo personas de entre dieciséis a veintitrés años.

## 4. Antecedentes

### 4.1. Personajes del Cine, Televisión, o Cómic que Padecen o Experimentaron Ataques de Pánico

#### 4.1.1. *Jessica Cruz*

Jessica Cruz es un personaje ficticio del universo de DC Comics, creado por Geoff Johns y Ethan Van Sciver, teniendo su primera aparición completa en 2014. Jessica Cruz es una linterna verde de la tierra, y tras un pasado traumático cuando era joven, ella desarrolla un trastorno de ansiedad, la agorafobia, miedo a lugares o situaciones que puedan generar pánico, miedo o impotencia; Jessica logra afrontar su pasado y es por ello que se vuelve una linterna verde, sin embargo la ansiedad no es algo que desaparece de la noche a la mañana, y es por ello que ella lucha todos los días con su ansiedad y ataques de pánico y sobrellevar sus responsabilidades como superheroína.

Jessica Cruz es uno de los pocos personajes, y en específico super héroes, que lidian con la ansiedad de manera responsable, ya que además de utilizar la ansiedad como un punto de desarrollo para la narrativa del personaje y la historia en general, logran también transmitir el sentir y pensar que sufren las personas que padecen de un trastorno de ansiedad y muestran posibles soluciones que se pueden utilizar para poder combatir la ansiedad.

Este proyecto toma como inspiración a este personaje por ser uno de los precursores en la narrativa ficcional de super héroes que abordan la ansiedad desde diferentes perspectivas, logrando plasmar un personaje ficticio basado en los superhéroes, pero con vulnerabilidades humanas reales, logrando así una empatía más sólida con el personaje.

#### 4.1.2. *Tony Stark, Iron Man 3 (2013)*

Tony Stark, alias Iron Man, es un personaje ficticio de Marvel Comics, y forma parte del universo cinematográfico de Marvel (UCM). Este personaje es uno de los protagonistas más importantes dentro del UCM y mantiene una actitud y personalidad muy definida, como una persona extrovertida e ingeniosa, pero también a veces egoísta y egocéntrica; sin embargo, a través de los años su actitud cambia, volviéndolo un personaje más interesante y dinámico.

Iron Man 3 es una película lanzada en 2013, en donde la personalidad de Tony Stark cambia de manera ya que se exploran los estragos de los acontecimientos de la película Los Vengadores (2011) en la vida personal del personaje, provocándole una ansiedad constante, y desembocando en

ataques de ansiedad; además de derrotar a un enemigo que amenaza con destruir todo lo que ama. Durante la película se pudo observar una actitud de Tony por querer controlar y tener todo bajo control, ya que esto le genera cierta paz, y de esta forma puede proteger a sus seres queridos, sin embargo, es evidente que su salud mental no se encuentra bien, y empieza a presentar síntomas de ansiedad, y al intentar lidiar solo con esta situación y no permitir recibir ayuda, esto desencadena en que experimente ataques de pánico; conforme avanza la historia, Tony acepta que no puede hacer todo solo y se permite aceptar ayuda de personas cercanas a él, y es así como logra controlar sus ataques de ansiedad.

Se tomó como inspiración a este personaje por el trasfondo de las situaciones que pueden llegar a generar ataques de ansiedad, además de predisponer al individuo y hacerlo crear y creer escenarios irreales e imaginarios, limitando varios aspectos de su vida por el miedo propiamente generado.

#### **4.1.3. *Randall Pearson, This is Us (2016-2022)***

Randall Pearson es un personaje ficticio de la serie *This is us*, él es hijo de Jack y Rebeca Pearson, y hermano de Kevin y Kate. La serie es un drama familiar que aborda y entrelaza las vidas de los integrantes de la familia Pearson en diferentes puntos de su vida, tanto en el presente siendo adultos, como en el pasado mediante saltos de tiempo, la serie aborda temas, situaciones y problemas de la vida cotidiana que afrontan cada uno de los hermanos tanto personalmente como en su vida social y familiar.

Randall es una persona perfeccionista y exigente consigo mismo, lo cual desde su juventud le trajo problemas de estrés que desembocaron en problemas de ansiedad y ataques de pánico, pero fue gracias a su familia, en especial a su padre, que pudo afrontar las situaciones que le provocaban ansiedad. Sin embargo, tras la muerte de su padre, siendo Randall aún muy joven, hizo que este perdiera la capacidad de controlarse ante ciertas situaciones y decidido lidiar él solo con su ansiedad, pero no fue hasta una edad adulta que sus crisis de ansiedad se agravaron, interfiriendo en su vida profesional, personal y familiar, siendo ahora el padre de familia; pero gracias a la ayuda de sus hermanos, su esposa e hijas, decide buscar ayuda profesional y trabajar en mejorar su salud mental.

El proyecto se inspira en el personaje por plasmar a través del tiempo como es sobrellevar ciertas situaciones que provocan ansiedad, afrontarlas y seguir adelante; ya que la ansiedad puede

presentarse en diversos puntos de la vida, y pueden prevalecer por varios años, y si bien no se puede predecir la ansiedad, es posible prevenirse gracias a diversas técnicas, tratamientos y rutinas que ayudaran a reducir los niveles altos de ansiedad.

## **4.2. Videojuegos que Emplean Mecánicas de Juego Similares**

### **4.2.1. *Stick Fight: Shadow Warrior & Stickman Game***

Stick Fight: SW es un videojuego de acción para dispositivos móviles que se encuentra actualizado hasta 2022 y desarrollado por el estudio Homecooked Games. Dentro del videojuego se controla a un personaje que se ubica en el centro de la pantalla y el objetivo es derrotar al mayor número de enemigos que irán apareciendo en los extremos de la pantalla. Para controlar al personaje es necesario realizar toques en la pantalla para indicar la dirección en la que se quiere desplazar el enemigo para atacar a los enemigos, si se falla, el personaje queda indefenso y puede ser atacado por los enemigos y pierde la partida.

El proyecto retoma las mecánicas principales de este videojuego debido a dos razones; la primera por implementar mecánicas de juego poco conocidas y aún sin ser aprovechadas en su totalidad, y la segunda, por ser un videojuego dinámico que requiere concentración y coordinación para poder alcanzar un objetivo.

### **4.2.2. *Celeste***

Celeste es un videojuego multiplataforma del género de plataformas, desarrollado por el estudio Extremely OK Games y lanzado en 2018. Celeste es un videojuego en el cual controlamos a una joven llamada Madeline y mediante saltos, escalar paredes, y desplazarnos en medio del aire, se deben superar varios niveles hasta llegar a la cima de una montaña.

La historia gira en torno a Madeline, una joven con algunos problemas de la salud mental, como depresión y la ansiedad, que decide aventurarse a subir la montaña Celeste, la cual tiene propiedades místicas y espirituales que la ayudaran a sanar, sin embargo, el trayecto hasta la cima será duro y complicado, ya que deberá luchar consigo misma. Durante su trayecto a la montaña conocerá a un par de personajes que la guiarán y ayudarán para que pueda continuar su camino.

El proyecto a desarrollar se inspira en Celeste por el cómo se deben de abordar los problemas de salud mental en los videojuegos y como este tipo de temáticas pueden utilizarse en mecánicas de

juego; por ejemplo, en Celeste, al llegar a un punto específico de la historia Madeline sufre de un ataque de pánico y comienza a perder el control, sin embargo gracias a que un personaje que la acompaña le enseña una técnica de relajación basada en la respiración imaginando una pluma, ella logra calmarse; y gracias a la interactividad que brindan los videojuegos esta técnica de relajación se pudo reflejar en una mecánica de juego basada en mantener en un sector de la pantalla dicha pluma y simular con un botón la respiración.

#### **4.2.3. *Hollow Knight***

Hollow knight es un videojuego del género metroidvania desarrollado por Team Cherry y lanzado en 2017 para diversas plataformas como Xbox, Playstation y Nintendo Switch. El género metroidvania es la combinación de los géneros de juego de aventura, acción y plataformas, en donde la obtención de habilidades para superar retos o explorar secciones del nivel es una de sus características más importantes y distinguibles de otros géneros de juego. Hollow Knight le da al jugador el control de un misterioso caballero, el cual deberá adentrarse en la ciudad de Hallownest y aventurarse en las profundidades de la ciudad para descubrir los misterios que le aguardan ahí. El juego cuenta con diversas mecánicas de juego que enriquecen no solo la jugabilidad, sino la propia narrativa del mismo, creando una sinergia que vuelve al juego en una experiencia sumamente completa.

Hollow Knight implementa una dinámica de juego llamada alma, el alma dentro del videojuego es un tipo de energía que tienen la mayoría de los enemigos y estructuras especiales, el alma es extraída por el protagonista al momento de atacar estos elementos y está llena un medidor visible en la interfaz de usuario (HUD) con forma de máscara, el alma que se absorbe puede ser utilizada para curarse, realizar ataques especiales o movimientos de desplazamiento especiales.

Para el proyecto presentado aquí, se tomaron ciertos aspectos de esta dinámica de juego, en específico el medidor y su funcionamiento, y se adaptaron a las mecánicas y narrativa del trabajo a desarrollar; ya que dentro del proyecto un medidor de este tipo puede ser una representación visual del estado mental del personaje.

## 5. Marco Teórico

### 5.1. Arte Digital

*Es producto de la creatividad artística y la mezcla de tecnologías digitales, ciberespacio y vida artificial, constituyéndose en un universo original de estética, espacio y tiempo. A su vez, las actuales manifestaciones artísticas experimentan y amplían tanto las posibilidades expresivas y de divulgación como las de la participación y apreciación de los espectadores. (Regil Vargas, 2005, 8)*

El arte digital es una forma de expresión artística, en donde el artista requiere o hace uso de dispositivos o herramientas digitales durante el proceso de desarrollo o exhibición de la obra de arte. Existen diversos tipos de arte digital, como el modelado y escultura 3D (tres dimensiones), la animación digital, el dibujo y fotografía digital, y los videojuegos; cada uno destaca por sus técnicas, procesos, herramientas y posibilidades de desarrollo y proyección en diversos medios digitales y tradicionales.

Sin embargo, Yoon Kim, Hyung y Tae Kim distinguen el arte digital (digital art) y el arte de juegos digitales (digital game art), definiendo este último como el uso de elementos propios del desarrollo y diseño de videojuegos para expresar y transmitir un mensaje, creando así una nueva forma de arte que atiende a la interactividad del espectador con la obra (Kim et al., 2013)

Dicha distinción no pretende separar uno del otro, sino que ayuda a profundizar y consolidar mejor la definición del arte digital, ya que el arte de juegos digitales son un tipo de arte digital que engloba en sí mismo otros tipos de el mismo, convirtiéndola en una disciplina que utiliza diferentes técnicas artísticas digitales y el uso de herramientas de desarrollo de videojuegos; como motores de videojuegos, diseño de mecánicas y niveles, inteligencia artificial y arquitectura de videojuegos; para así crear obras artísticas interactivas, donde la obra le brinda al espectador la posibilidad de ser partícipe de la misma, brindándole experiencias inmersivas que expresen ideas, sentimientos, emociones o pensamientos.

### 5.2. Salud Mental

*La salud mental es un estado de bienestar en el que un individuo se da cuenta de sus propias capacidades, puede hacer frente a las tensiones normales de la vida, puede trabajar productivamente y puede hacer una contribución a su comunidad.*

*La salud mental es fundamental para nuestra capacidad colectiva e individual como seres humanos para pensar, emocionarse, interactuar entre nosotros, ganarnos la vida y disfrutarla. Sobre esta base, la promoción, protección y restauración de la salud mental puede considerarse una preocupación vital de las personas, las comunidades y las sociedades de todo el mundo. (World Health Organization, 2018)*

Sin embargo, el no cuidar y atender de manera oportuna la salud mental puede provocar un trastorno mental. La Organización Mundial de la Salud (OMS) menciona que “Hay una gran variedad de trastornos mentales, cada uno de ellos con manifestaciones distintas. En general, se caracterizan por una combinación de alteraciones del pensamiento, la percepción, las emociones, la conducta y las relaciones con los demás” (Organización Mundial de la Salud, 2019).

El Manual diagnóstico y estadístico de los trastornos mentales (2013) enlista los tipos de trastornos mentales, entre los más conocidos son: “espectro de la esquizofrenia y otros trastornos psicóticos”, “trastorno bipolar y trastornos relacionados”, “trastornos depresivos” y “trastornos de ansiedad”.

Se seleccionó esta temática general por lo importante que es conocer y atender la salud mental, y por un desconocimiento general de esta, lo que provoca que se generen prejuicios y estereotipos hacia las personas que padecen alguna enfermedad asociada a una mala salud mental.

### **5.3. Trastornos de Ansiedad**

*La ansiedad ocasional es una parte esperada de la vida. Puede sentirse ansioso cuando se enfrenta a un problema en el trabajo, antes de realizar un examen o antes de tomar una decisión importante. Pero los trastornos de ansiedad involucran más que preocupaciones o miedos temporales. Para una persona con un trastorno de ansiedad, la ansiedad no desaparece y puede empeorar con el tiempo. (National Institute of Mental Health, 2018)*

Dentro de los trastornos de ansiedad existen diversos tipos como el trastorno de ansiedad generalizada, separación o social, fobias específicas, o el trastorno de pánico.



El Manual diagnóstico y estadístico de los trastornos mentales define el trastorno de pánico como “una oleada abrupta de miedo intenso o malestar intenso que alcanza su punto álgido en cuestión de minutos, y durante la cual se producen cuatro o más de una lista de 13 síntomas físicos y cognitivos” (American Psychiatric Association, 2013, 208)

El trastorno de pánico se caracteriza por ser un episodio repentino de miedo o terror, ya sea por una causa real o proyectada por el individuo, el cual escala de forma abrupta y manifiesta reacciones físicas como ritmo cardíaco acelerado, tensión muscular, dolor en el pecho o una sensación de pérdida de control que generan un miedo a morir. El INSM (Instituto Nacional de Salud Mental) dice que “Muchas personas con trastorno de pánico se preocupan y temen la posibilidad de tener otro ataque. Es posible que una persona que tiene el trastorno de pánico puede desanimarse o avergonzarse” (National Institute of Mental Health, 2017).

### **5.3.1. Estrés y Ansiedad**

*El estrés es la respuesta física o mental a una causa externa, como tener muchas tareas o padecer una enfermedad. Un estresor o factor estresante puede ser algo que ocurre una sola vez o a corto plazo, o puede suceder repetidamente durante mucho tiempo.*

*La ansiedad es la reacción del cuerpo al estrés y puede ocurrir incluso si no existe una amenaza real. (National Institute of Mental Health, n.d.)*

Según Healthline (2022) ambos comparten sintomatologías similares como: “problemas para dormir”, “problemas digestivos”, “dificultad para concentrarse”, “tensión muscular”, “irritabilidad o ira”.

*Los trastornos de ansiedad son problemas de salud mental que se relacionan con experimentar en exceso ansiedad, miedo, nerviosismo, preocupación o terror ... se encuentran entre los problemas de salud mental más comunes. Afectan a personas de todas las edades, incluidos adultos, niños y adolescentes. Hay muchos diferentes tipos de trastornos de ansiedad, con síntomas diferentes. (Nemours KidsHealth, 2013)*

Mayo Clinic (2021) enlista los trastornos de ansiedad más comunes: “agorafobia”, “trastorno de ansiedad debido a una enfermedad”, “trastorno de ansiedad generalizada”, “trastorno de pánico”,

“mutismo selectivo”, “trastorno de ansiedad por separación”, “trastorno de ansiedad social (fobia social)”, “fobias específicas” y “trastorno de ansiedad inducido por sustancias”.

### **5.3.2. Trastorno de Pánico**

*El ataque de pánico se define como un episodio de ansiedad repentino e intenso que se acompaña de sensaciones físicas desagradables (palpitaciones, sensación de ahogo, mareos...) y de pensamientos que suelen ser catastróficos (miedo a perder el control o a morir, entre otros). (Forcadell López et al., 2019)*

Sin embargo, Mayo Clinic menciona que “si tienes ataques de pánico inesperados y recurrentes, y pasas mucho tiempo con miedo constante de sufrir otro ataque, es probable que tengas una afección llamada «trastorno de pánico».” (Mayo Clinic, 2018)

De acuerdo con el manual diagnóstico y estadístico de los trastornos mentales (2013), la sintomatología que se presenta durante un ataque de pánico es: palpitaciones, golpeteo del corazón o aceleración de la frecuencia cardiaca, sudoración, temblor o sacudidas, sensación de dificultad para respirar o de asfixia ... sensación de mareo, inestabilidad, aturdimiento o desmayo ... desrealización (sensación de irrealdad) o despersonalización (separarse de uno mismo), miedo a perder el control o de “volverse loco.””, miedo a morir.

Según Burke y Collins (2022) la causas de este trastorno aún se encuentran en investigación, pero existen factores específicos que pueden provocarlo: la genética, que actualmente divide a la comunidad científica debido a la falta de pruebas sólidas que relacionan el trastorno con este factor; sin embargo, varios estudios coinciden en dos características, el primero es sobre los cambios en el gen "COMT", el cual impacta en las habilidades cognitivas y el comportamiento; y el segundo, que habla sobre una estrecha correlación genética entre otros problemas mentales como el trastorno depresivo mayor o el neuroticismo. Otras causas para este trastorno son los cambios en la vida, que se relaciona con el inicio de nuevas etapas, por eventos o situaciones estresantes, o retos que generen estrés, algunos ejemplos son cambiar de escuela, la muerte de alguien cercano, iniciar en un nuevo trabajo, tener un hijo o mudarse. Y finalmente la ansiedad, la cual es normal y natural en los seres humanos, ya que algunas veces uno se siente preocupado, nervioso o abrumado, pero el que estos sentimientos se prolonguen e interfieran con nuestras actividades cotidianas es cuando se podrían dar indicios de padecer algún trastorno de ansiedad.

Dentro de los posibles tratamientos para este trastorno:

*La psicoterapia desempeña un papel fundamental, quedando la psicofarmacología relegada a un segundo plano, cuya principal indicación es la complementariedad a la primera; sin embargo, no debe olvidarse que la opción farmacológica también resulta útil cuando no es posible aplicar un tratamiento psicoterapéutico, o bien, el paciente rechaza esta opción.*

*Se ha comprobado la especial utilidad de llevar a cabo una buena labor educativa del enfermo y de la familia, para lo que puede recurrirse al empleo de libros o panfletos divulgativos; también ha funcionado la incorporación del afectado en grupos de autoayuda. (Gómez Ayala, 2012, 37)*

Diversas páginas web enfocadas en la medicina concuerdan que existen diversas formas de tratar los trastornos de pánico, como, por ejemplo: ejercicios de relajación y respiración, realizar ejercicio de manera constante, meditación o ejercicios de atención plena y reconocer y aceptar la experiencia.

*El trastorno de pánico es a menudo una condición a largo plazo que puede ser difícil de tratar. Es posible que tenga un trastorno de pánico si ha experimentado múltiples ataques de pánico y tiene un temor persistente de que vuelvan a ocurrir.*

*Puede que no sea posible "curar" completamente su trastorno de pánico, pero el tratamiento puede ayudar. La terapia, incluida la TCC (Terapia Cognitiva-Conductual), trabajar para disminuir los factores estresantes de la vida y tomar medicamentos son opciones posibles. (Burke & Collins, 2022)*

Los trastornos de ansiedad son un tema complejo y delicado de abordar, y que merecen tener visibilidad en medios digitales de manera responsable y proactiva, ya que en los últimos años ha existido una tendencia de romantizar o no dirigir de manera adecuada las enfermedades mentales en diversos medios de consumo, provocando una desinformación y una distorsión de la realidad.

## 5.4. Diseño y Desarrollo del Videojuego

### 5.4.1. Documento de Diseño de Juego

*El documento de diseño de juego o Game Design Document (GDD) es el equivalente al esqueleto del videojuego.*

*Podría decirse que el documento de diseño o Game Design Document es una síntesis de lo que va a ser el videojuego en sí (concepto, historia, género, número de plataformas, equipo de producción...) y la base de su futuro desarrollo. (Tokio School, 2019)*

El GDD no tiene una estructura definida, ya que este se adapta a las necesidades de cada proyecto, por lo que no existe una plantilla sobre la cual se trabaja, sin embargo, existen características que comparten algunos de estos documentos, como el diseño de juego (game design), diseño de nivel (level design), descripción del juego, jugabilidad, mecánicas, narrativa, arte y diseño, entre muchos otros aspectos. El uso de un GDD facilita la organización del proyecto, además de marcar una dirección general de este, creando así un documento que mejora el flujo de trabajo dentro del equipo, como para futuros desarrolladores que se unan al desarrollo, manteniendo al equipo bajo una misma dirección.

### 5.4.2. Diseño de Videojuegos

*El diseño del juego es el acto de decidir cómo debería ser un juego ... el diseñador suele estar involucrado en el desarrollo de un juego desde el principio hasta el final, tomando decisiones sobre cómo debería ser el juego a lo largo del camino. (Schell, 2015, 38)*

Schell (2015) enlista las habilidades o conocimientos que necesita tener un diseñador de juegos (game designer): animador, business, cinematografía, comunicación, escritura creativa, economía, ingeniería, gestión, matemáticas, música, hablar en público, sound design, visual arts. Un game designer debe conocer y comprender las diversas fases del diseño y desarrollo de videojuegos, ya que de esta forma sabrá cómo abordar y tomar las decisiones más adecuadas del proyecto.

Entre los aspectos que se abordan al momento de desarrollar el diseño del juego están la descripción técnica, tema, género público meta, jugabilidad, mecánicas, dinámicas, reglas y objetivos, personajes y enemigos, interfaz de usuario, entre otros elementos.

#### **5.4.3. Desarrollo de Videojuegos**

*El desarrollo de videojuegos es el proceso de dar vida al juego. Esto incluye la codificación, la programación de software, los efectos de sonido, la ingeniería, el renderizado, las pruebas y varios otros procesos para poner en marcha el videojuego requerido. (New England Institute of Technology, 2020)*

*Un desarrollador de juegos es cualquiera que tenga alguna participación en la creación del juego. Los ingenieros, animadores, modeladores, músicos, escritores, productores y diseñadores que trabajan en juegos son todos desarrolladores de juegos. Los diseñadores de juegos son solo una especie de desarrolladores de juegos. (Schell, 2015, 39)*

Las etapas del desarrollo de videojuegos (game development) pueden dividirse en tres; preproducción, etapa donde se planifica y se desarrolla conceptualmente el juego y los alcances del mismo, aspectos propios del GDD, además de desarrollar el apartado narrativo y los primeros bocetos de personajes, entornos entre otros elementos; la producción, donde se desarrollan todas las ideas de diseño, arte, sonido y programación, además de realizar pruebas conocidas como pruebas de juego (game testing), donde se identifican errores en el videojuego; y finalmente la postproducción, donde el mantenimiento del juego y el desarrollo de nuevo contenido son las tareas principales.

#### **5.4.4. Diseño de Niveles**

*El diseño de niveles es la fase del desarrollo del juego que se ocupa de crear los escenarios del juego. A diferencia del diseño del entorno, que se centra en la composición del escenario y el fondo, el diseño de niveles incorpora las capacidades del jugador, la mecánica del juego, los obstáculos y los elementos reconocibles que crean una experiencia de usuario positiva.*

*Un diseñador de niveles crea los escenarios (o mapas) y las misiones en un videojuego. Los diseñadores de niveles se inspiran en el arte conceptual y GDD para crear un entorno creíble, establecer los límites del juego, configurar el ritmo y la composición de cada nivel, así como mantener un estilo coherente con los objetivos del juego. (MasterClass, 2020)*

Como señala Christopher Totten en su libro *An Architectural Approach to Level Design* (2014) las características que deben abordarse en un level design son; diseño espacial, emociones a través del espacio de juego, teoría arquitectónica, requerimientos funcionales, usabilidad”, deleite, arreglos espaciales, estructura del espacio de juego, tipos de tamaño espacial, consideraciones de la cámara, comunicación visual, diseño de molécula, diseño de plano, dibujo de secciones, entre otras.

Entre estos puntos, resalta el tipo de tamaño espacial, el cual se compone de tipos de espacio; el estrecho, en el cual se le brinda al jugador una sensación de vulnerabilidad o limitación en sus acciones; el íntimo, donde el diseño general del nivel en cuanto a métrica del personaje y el espacio, las interacciones con el nivel y el alcance de estas para el jugador, y la adaptabilidad del espacio para las mecánicas del juego sean las adecuadas; y el espacio de introspección, en el cual el jugador puede sacar ventajas sobre ciertas situaciones, o verse expuesto ante una amenaza, además compartir ciertas características con los espacios estrechos, pero con un enfoque diferente.

#### **5.4.5. Videojuegos de Acción**

*Los juegos de acción se centran en desafíos físicos que requieren coordinación ojo-mano y reflejos rápidos. Los juegos de disparos en primera persona, los juegos de plataformas, los juegos de lucha, los juegos de supervivencia y los juegos de ritmo son parte del género de acción. (MasterClass, 2020)*

Jugar juegos de acción no se trata solo de completar misiones y superar los límites de tiempo. También estás ayudando a tu personaje a sobrevivir a toda la prueba. Esta presión adicional es una característica clave que distingue a los juegos de acción de los juegos de aventura. El desafío de mantenerse con vida también se suma a la emoción. La dificultad de los videojuegos aumenta a medida que avanzas, y algunos títulos exigen toda tu atención para superar ciertas batallas y enemigos. (Nanou, 2021)

### **5.5. Programación**

#### **5.5.1. Máquina de Estados Finita (FSM)**

*Una máquina de estados finitos es un dispositivo, o un modelo de un dispositivo, que tiene un número finito de estados en los que puede estar en un momento dado y puede operar en la entrada para hacer transiciones de un estado a otro o para hacer que se*

*lleve a cabo una salida o acción. Una máquina de estados finitos solo puede estar en un estado en cualquier momento. (Buckland, 2004, p44)*

Las FSM se utilizan en una gran cantidad de videojuegos, y se usan de diferentes formas, como por ejemplo en la programación y control de las inteligencias artificiales de ciertos agentes dentro de un videojuego.

## **5.6. Narrativa (Storytelling)**

### **5.6.1. Narrativa**

*La narrativa (storytelling) es el arte interactivo de usar palabras y acciones para revelar los elementos e imágenes de una historia mientras fomenta la imaginación del oyente.*

*El storytelling implica una interacción bidireccional entre un narrador y uno o más oyentes. Las respuestas de los oyentes influyen en la narración de la historia. De hecho, la narración surge de la interacción y los esfuerzos cooperativos y coordinados del narrador y la audiencia. (National Storytelling Network, n.d.)*

En los videojuegos de rol, la historia es central y los jugadores deben progresar a medida que la historia continúa. Un ejemplo de juego de rol es The Legend of Zelda o Fallout. En los juegos de aventuras, el jugador puede explorar más y crear sus “propias historias” mientras intenta completar misiones para progresar. (Starloop Studios, 2021)

Desde hace varios años el storytelling ha tenido un papel muy importante en el mundo de los videojuegos, desde las sagas clásicas como The Legend Of Zelda o Final Fantasy, hasta títulos más actuales como The Last of Us o Life is Strange 2, y si bien existen un gran número de videojuegos que prefieren desarrollar o experimentar nuevas mecánicas de juego, también existen otros títulos que deciden desarrollar una historia y personajes únicos y así poder brindarle al jugador nuevas experiencias narrativas dentro de los videojuegos.

### **5.6.2. Diseño de Narrativa**

*Se enfoca en cómo ramificar una historia. Una historia se puede contar de forma lineal, pero el diseñador narrativo la dividirá en muchas partes más pequeñas, de modo que se convierta en una conversación con el jugador en lugar de un discurso dirigido a él.*

*En Ubisoft, el diseñador narrativo a menudo trabaja en colaboración con los diseñadores de juegos y los equipos de audio, producción y diseño de misiones. Son básicamente una interfaz entre la mecánica del juego y la narrativa. (Manileve, 2021)*

El diseñador narrativo es un puesto de trabajo en la industria de los videojuegos que es relativamente nuevo, pero retoma muchas características del storytelling y el perfil está encaminado a su implementación en un videojuego.

En lo videojuegos las historias que comúnmente se abordan son sobre fantasías de poder, guerra, salvar al mundo o de competición; sin embargo, la particularidad de los videojuegos es la participación activa del jugador, y es este el que moldea se cuenta esta historia, y esto se logra a través de los diálogos interactivos, las mecánicas de juego, el apartado gráfico como arte o interfaces de usuario y sobre todo las decisiones que puede tomar el propio jugador durante el videojuego. Es por esto que los videojuegos se diferencian de otros medios, por la capacidad que tiene el usuario de poder interactuar y ser partícipe de la obra; crear una conexión con los personajes a través de sus historias y mecánicas de juego, adentrarnos en la mitología y descubrir secretos en estos mundos ficticios o permitirnos crear nuestros propios mundos o historias dentro del juego y gozar de experiencias únicas y memorables.

## **5.7. Arte**

### **5.7.1. Diseño de Personajes**

*El diseño de personajes es el proceso de generar ideas, esbozar, delinear y dibujar personas ficticias o seres no humanos, a menudo desde cero. Los diseñadores imbuyen a estos seres con personalidades y características únicas, como su apariencia física, patrones de habla, movimientos y personalidades. (Jones, n.d.)*

Desde el punto de vista de DeGuzman en la página StudioBinder (2021) existen tres componentes clave al momento de diseñar personajes; la silueta, que consiste en eliminar cualquier



color, detalle, contorno del diseño del personaje y quedar únicamente con la forma del personaje en color negro, el objetivo de esto es lograr que el personaje sea fácilmente reconocible y esto únicamente a partir de la silueta, por lo que es necesario un buen uso del lenguaje de las formas; la paleta, el cual se refiere al uso adecuado de los colores en el diseño del personaje, por lo que es necesario tener un buen conocimiento acerca de la teoría del color y como los colores interactúan entre sí; y finalmente la exageración, el cual deforma o exagera ciertas características físicas del personaje para provocar respuestas emocionales específicas, y de esta forma el personaje se perciba de diferentes maneras, ya sea como un héroe o villano.

El diseño de personajes no solamente abarca el apartado visual, y para que exista un buen personaje, debe tener un perfil psicológico y una historia, MasterClass (2020) enlista 6 características sobre cómo diseñar un personaje de videojuego, las cuales son, “hazte una idea general”, “establecer una historia de fondo”, “averiguar su arco”, “añadir rasgos de carácter”, “definir relaciones”, “proporcione una estética que encaje”.

### **5.7.2. Teoría del Color**

*La teoría del color trata de comprender cómo funcionan las combinaciones de colores y cómo los colores no siempre son como los percibimos.*

*Cuando se colocan juntos, los colores se comunican entre sí. Algunos son complementarios, algunos hacen que otros colores resalten, mientras que algunos colores no van bien juntos, lo que hace que ambos parezcan aburridos como resultado.*  
(Clip Studio paint, n.d.)

Para entender las combinaciones de colores es necesario conocer la rueda de color, Bretaña (2020) explica que “Las ruedas de color permiten representar geométricamente las relaciones cromáticas y muestran la relación entre los colores primarios, los colores secundarios y los colores terciarios”. Estos colores se dividen en dos grupos, los cálidos; como amarillo, rojo, naranja, y expresan calidez, energía, pasión, alegría o dinamismo; y los fríos, como azul, morado y verde, que evocan serenidad, tristeza, calma, o soledad.

Otro punto importante a tomar en cuenta son los esquemas de color, los cuales se utilizan para definir el ambiente o estado de ánimo tanto de personajes como paisajes, los esquemas más destacables son los monocromáticos, análogos, complementarios y tríadas, estos se agrupan y

definen gracias al esquema de la rueda de color y cada uno de estos grupos transmite sentimientos y emociones diferentes acordes a la situación en donde se utilicen.

### **5.7.3. Arte de Pixel (Pixel art)**

*El estilo visual de Atari 2600, Nintendo Entertainment System, Sega Genesis y los primeros gráficos por computadora influyeron, y siguen influyendo, en millones de mentes creativas. El pixel art se nutre de esa influencia y nostalgia.*

Los artistas que hacían videojuegos en las décadas de 1970 y 1980 tenían que innovar y reducir las imágenes a su esencia. Unos pocos píxeles rojos tendrían que sugerir el sombrero de Mario, y uno o dos píxeles tendrían que entenderse como sus manos o su cara. (Adobe, n.d.)

Se tomó este estilo visual debido a varias razones, la primera fue la nostalgia; esto debido a remembranzas por los antiguos videojuegos de la década de los 80s y 90s y el poder volver a experimentar esa sensación visual en dispositivos modernos y con una mayor potencia gráfica; la segunda es la eficiencia técnica, ya que en comparación con juegos AAA 3D o 2D, los requerimientos técnicos para el desarrollo artístico son elevados debido a la cantidad de detalles que estos requieren, sin embargo, el pixel art es un estilo visual que no requiere muchos detalles para poder comunicar un mensaje efectivo visualmente, además que los procesos de animación y arte son relativamente más cortos, pero esto dependerá de la complejidad del proyecto; y finalmente su implementación con software y hardware moderno, ya que esto permite mezclar el estilo pixel art con los nuevos avances tecnológicos realizados en los motores de videojuegos, lo que permite crear un estilo de pixel art totalmente diferente y adaptado a los estilos visuales de nuestra época.

## **5.8. Herramientas de Desarrollo**

### **5.8.1. Unity**

*El término "motor de juego" surgió a mediados de la década de 1990 en referencia a los juegos de disparos en primera persona (FPS) como el increíblemente popular Doom de id Software ... los desarrolladores comenzaron a otorgar licencias de juegos y los convirtieron en nuevos productos mediante la creación de nuevo arte, diseños del mundo, armas, personajes, vehículos y reglas del juego con sólo cambios mínimos en el software del "motor" ... Los motores se hicieron altamente personalizables a través de lenguajes de secuencias de comandos como Quake C de id, y las licencias de motores*

*comenzaron a ser una fuente de ingresos secundaria viable para los desarrolladores que los crearon. (Gregory, 2017, 11)*

Citando a la página de Microsoft “Unity es una plataforma de desarrollo 3D en tiempo real para crear aplicaciones 2D y 3D, como juegos y simulaciones, utilizando .NET y el lenguaje de programación C#.” (Microsoft .NET, n.d.)

Unity es un motor de juegos que ofrece un espacio de trabajo completo en el cual se pueden crear proyectos interactivos de un alto nivel, desde videojuegos independientes o prototipos, hasta experiencias en realidad aumentada o videojuegos AAA.

Entre las características más importantes se encuentran; su entorno de trabajo, ya que gracias al ser un motor de desarrollo en tiempo real, es posible modificar y editar el proyecto al mismo tiempo que se ejecuta, permitiendo una fácil edición y depuración tanto de código como de otros elementos que se encuentran dentro de la escena; desarrollo de proyectos 2D y 3D, al tener herramientas específicas que permiten desarrollar en uno o ambos formatos, además de brindar plantillas predefinidas para diversos tipos de proyectos; arquitectura de desarrollo, cada proyecto se compone de escenas, las cuales contienen los elementos organizados en una jerarquía y los componentes que otorgan diversos comportamientos a estos objetos, y gracias al motor de física integrado en Unity, es como se logra la interacción de todos los elementos dentro de una escena y crear una experiencia, además de la implementación del lenguaje de C Sharp (C#) el cual cuenta con una fácil codificación para implementar comportamientos específicos dentro de los proyectos; multiplataforma, este es uno de sus atractivos más interesantes, debido a que Unity permite desarrollar y compilar los proyectos en diferentes plataformas, como PC, Xbox One, Xbox Series X|S, Playstation 4 y 5, Nintendo Switch y HTML 5, y con la facilidad de poder cambiar entre plataformas sin necesidad de realizar cambios significativos en nuestro proyecto; Unity Personal, la opción más accesible y gratuita para futuros desarrolladores o estudiantes, ya que brinda acceso a todas las funcionalidades del motor, actualizaciones constantes del mismo, y sin cobro de regalías para videojuegos lanzados en el mercado, siempre que los ingresos de la empresa sean inferiores a los 100 mil dólares estadounidense en los últimos 12 meses; entre otras características como la Asset Store, Unity Learn y sus propias certificaciones.

Entre los videojuegos más populares desarrollados en Unity encontramos a Hollow Knight, Cuphead, Fall Guys: Ultimate Knockout, Ori and the Blind Forest, Subnautica, Rust, Marvel Strike Force, Gris, entre muchos otros juegos.

### **5.8.2. Visual Studio**

*Un entorno de desarrollo integrado (IDE) es un programa rico en funciones que admite muchos aspectos del desarrollo de software. El IDE de Visual Studio es una plataforma de lanzamiento creativa que se puede usar para editar, depurar y compilar código, y luego publicar una aplicación. Más allá del editor y depurador estándar que proporciona la mayoría de los IDE, Visual Studio incluye compiladores, herramientas de finalización de código, diseñadores gráficos y muchas más funciones para mejorar el proceso de desarrollo de software. (Microsoft, 2022)*

Visual Studio es la opción más popular entre los desarrolladores que utilizan Unity por contar con una integración en sus herramientas para este motor de juego, como funciones de inspección y depuración, o Attach and Play; y en conjunto con la instalación de la documentación de Unity, Visual Studio se vuelve un editor versátil y flexible al momento de escribir y editar códigos (scripts) en C#.

### **5.8.3. Krita**

Tal como se expresa en la propia página de Krita (Krita, n.d.) “Krita es una aplicación para la creación y manipulación de imágenes. Nos enfocamos en pintura, ilustración, arte conceptual y otros trabajos creativos.” (Krita, 2009)

Uno de los atractivos más llamativos de Krita es su desarrollo con código libre, por lo que es gratuito y está en constante desarrollo tanto por los creadores como por la comunidad, además de contar con herramientas similares a su contraparte de pago Photoshop, como su amplia variedad de herramientas de pintura y vectores, el uso de filtros tanto artísticos como de retoque y la posibilidad de trabajar y exportar archivos en diversos formatos.

### **5.8.4. Aseprite**

Aseprite es un editor de gráficos de pago, y como se expresa en su página oficial “te permite crear animaciones 2D para videojuegos. Desde sprites hasta pixel-art, gráficos de estilo retro y lo que quieras de la era de los 8 y 16 bits” (Aseprite, n.d.)

Entre las características del software encontramos herramientas de animación como organizador de capas, modos de capa y onion skin; herramientas de color y edición como paletas de color; rotar, escalar mover capas, y pinceles personalizados; y el soporte de importación y exportación en diferentes formatos.

## **6. Marco Metodológico**

Dentro del área del desarrollo de videojuegos existen flujos de trabajos diversos, esto debido a que no existen reglas estrictas, permitiendo adaptar las necesidades y recursos con los que se cuentan para desarrollar el proyecto. Sin embargo, muchos coinciden en tres fases principales: preproducción, producción y post producción; para el desarrollo de este proyecto se desarrollaron las dos primeras fases y omitiendo temporalmente la última, ya que esta fase se centra en el desarrollo, mantenimiento y lanzamiento de nuevo contenido para el videojuego una vez que este es lanzado al mercado y el público comienza a interactuar con él.

### **6.1. Preproducción**

Durante la preproducción se desarrolla de forma conceptual el videojuego, delimitando los requisitos de diseño y desarrollo, los procesos desarrollados en esta etapa fueron:

- Diseño de videojuego: Descripción del videojuego, tema, género, público, jugabilidad, mecánicas, experiencia de juego, reglas, objetivos, estética, interfaz de usuario, entre otros elementos.
- Diseño de nivel: Diseño de la usabilidad del nivel, diseño espacial, requerimientos funcionales, estructura del espacio, recompensas, consideraciones de la cámara, entre otros.
- Narrativa: Desarrollo de la historia del videojuego, realizando una investigación sobre el tema, la premisa, logline, storyline, argumento, diseño de personaje y su contexto.

### **6.2. Producción**

Durante la producción se desarrolla y transforma el apartado conceptual el videojuego en un prototipo jugable, los procesos desarrollados en esta etapa fueron:

- Programación: Programación de las acciones del jugador y enemigos, animaciones, menús, escenas del juego, entre otros.

- Arte: Desarrollo del apartado visual tanto de personajes y enemigos, como de fondos, animaciones, interfaces, menús y escenas.
- Sonido / VFX: Fase donde se grabó y compuso la música del videojuego como sonidos de personajes, enemigos e interfaces.

## 7. Procesos

### 7.1. Preproducción

La fase de preproducción se dividió en dos etapas, el game design y el level design, ambas fueron desarrolladas de forma simultánea y posteriormente fueron unificadas en la documentación del videojuego. El formato del game design está basado en diferentes apartados de diversos game design documents para el mejor desarrollo y entendimiento de la documentación del videojuego. El level design está mayormente basado en el libro de Christopher W. Totten, *An architectural approach to level design*, retomando capítulos específicos del libro para poder diseñar y definir el tipo de experiencia dentro del nivel que se buscaba ofrecer al jugador.

#### 7.1.1. Diseño de Videojuego

**Título.** *The fall of the last guardian*.

**Descripción.** *The fall of the last guardian* es un videojuego perteneciente al género de acción, desarrollado en el motor de videojuegos Unity y para la plataforma Microsoft Windows. La historia del videojuego gira en torno a Jacobo, un joven con trastorno de pánico que descubre que posee superpoderes, y deberá aprender a dominar sus poderes y controlar sus emociones para poder proteger a la ciudad de un naciente grupo delictivo con superpoderes.

El videojuego consiste en derrotar a varias oleadas de enemigos que aparecerán en los bordes de la pantalla y perseguirán a nuestro personaje ubicado en el centro de la pantalla, el objetivo es derrotar a la mayor cantidad de enemigos sin que logren derrotar al jugador, al mismo tiempo que se debe controlar una barra de estado mental que refleja el estrés e irá subiendo cada vez que se derroten enemigos.

**Género.** Acción.

**Propósito y Público Objetivo.** El público objetivo son jóvenes, y adultos, de entre 15 y 23 años de edad.

**Plataformas de Distribución.** Inicialmente el videojuego será distribuido en la página web Itch.io.



**Categoría.** Dentro del mercado existen un par de juegos que utilizan la mayoría de las mecánicas implementadas en el videojuego; como por ejemplo *Stick Fight: Shadow Warrior & Stickman Game* y *Stick Fight*, ambos para dispositivos móviles.

Cabe destacar que estos juegos sirvieron como inspiración directa para el desarrollo de las mecánicas de este proyecto. Sin embargo, *The fall of the last guardian* se diferencia de estos juegos por dos razones; la primera es la jugabilidad, ya que existen mecánicas como la barra de estado mental o estado de bloqueo mental, que cambian por completo la forma en cómo se juegan este tipo de juego y brindando una experiencia de juego diferente; y la segunda es la narrativa del videojuego, ya que ambos juegos que se tomaron como inspiración no cuentan ninguna historia, y se limitan exclusivamente a brindar una experiencia de juego, y al ver este vacío en este tipo de juegos, se decidió ajustar y crear una narrativa a las mecánicas de juego.

**Tecnología.** El software utilizado para desarrollar el videojuego fue el siguiente:

- Motor de videojuegos: Unity 2020.3.24f1.
- Programación: Visual Studio Community 2019.
- Lenguaje programación: C Sharp (C#).
- Arte: Krita 5.0.2 y Aseprite v1.2.34.1.

**Jugabilidad.** Antes de iniciar una nueva partida, dentro del menú principal se encuentra el tutorial, el cual muestra gráficamente mediante cuadros de texto e imágenes el objetivo del juego, los controles, y las mecánicas de juego principales.

Una vez iniciada una partida, el jugador visualizara al personaje ubicado en la parte central de la pantalla, también un contador en la parte superior central de la pantalla, la cual refleja la cantidad de enemigos derrotados hasta el momento y en la esquina superior izquierda encontrará su barra de estado mental, la cual aumentará cada vez que el jugador elimine a cada enemigo. Esta barra al llegar al punto máximo hará que el personaje entre en un estado de pánico que le impedirá realizar cualquier acción hasta que la barra descienda gradualmente dado unos segundos y le permita volver a jugar. Es posible controlar dicha barra en cualquier momento, y el jugador deberá mantener presionada una tecla para hacer que el personaje entre en un estado de relajación o tranquilidad y

pueda reducir su barra de estado mental, sin embargo, al estar en este estado el personaje no podrá realizar ninguna acción hasta que se deje de presionar dicha tecla.

En los extremos laterales de la pantalla aparecerán los enemigos y perseguirán al personaje hasta reducir sus puntos de vida. El jugador únicamente tiene un punto de vida, por lo que un único ataque de los enemigos pondrá el fin del juego.

El jugador podrá presionar las teclas A o D para poder atacar a los enemigos que se aproximen, este ataque tiene un rango de alcance donde podrá atacar consecutivamente a los enemigos, pero si ataca y no hay un enemigo, el jugador quedara noqueado por unos segundos y quedará indefenso.

La partida termina cuando un enemigo ataca al personaje. Se despliega una pantalla que marca el fin del nivel y se muestra la puntuación alcanzada, además de un botón que redireccionará al menú principal para volver a iniciar una nueva partida.

### **Reglas del Videojuego**

- *The fall of the last guardian* es un videojuego para un solo jugador.
- Se necesita una computadora con un sistema operativo (SO) Microsoft Windows, un teclado, mouse, bocinas, y una conexión a internet.
- El objetivo del videojuego es derrotar al mayor número de enemigos, y para ello el jugador deberá controlar la barra de estado mental del personaje, e impedir que ataquen al personaje, de lo contrario se pierde la partida.
  - Los enemigos aparecerán en los extremos de la pantalla y perseguirán al jugador hasta derrotarlo o hasta que estos sean eliminados.
  - El jugador se ubica en el centro de la pantalla, por lo que deberá atacar a los enemigos que estén cerca de él para aumentar su puntuación y evitar ser derrotado.
  - La barra de estado mental aumentará según las acciones del personaje. Al llegar al máximo esta barra, el jugador experimentará un ataque de pánico

que le impedirá moverse o atacar durante un tiempo breve, al final el jugador volverá al estado normal y podrá seguir realizando sus acciones. La barra de estado mental puede ser reducida durante batalla, por lo que el personaje pasara a un estado de tranquilidad, sin embargo, no se podrá realizar ninguna acción mientras esté en ese estado.

- Al ser derrotado el jugador se guardará la puntuación más alta alcanzada y se podrá volver a ingresar al nivel para poder registrar y batir una nueva puntuación más alta con cada nueva sesión de juego.
- Dentro del videojuego existe un tutorial que explica las acciones que el jugador puede realizar.
  - Para desplazarse y atacar tanto hacia la izquierda como a la derecha se utilizan las teclas “A” y “D” respectivamente.
  - Para controlar la barra de estado mental se utiliza la tecla “SPACE”.
  - Para desplegar el menú de pausa durante la partida se utiliza la tecla “ESC”, donde se despliegan los botones de “Continue” o “Main Menu”.
- Existen menús con los cuales se podrá interactuar usando el clic izquierdo del mouse.

**Mecánicas, Dinámicas y Estéticas del Juego.** Con base en Tokio School (2020) “Una mecánica describe los componentes particulares del juego, al nivel de representación de datos y algoritmos; una dinámica es el comportamiento que emerge de los inputs y outputs entre el jugador y las mecánicas de juego.”

En un artículo sobre diseño y la investigación de juegos, se señala que la estética dentro de las mecánicas y dinámicas “describe las respuestas emocionales deseables evocadas en el jugador, cuando interactúa con el juego sistema.” (Hunicke et al., n.d.)

Las mecánicas, dinámicas y estéticas dentro del nivel son las siguientes:

- Mecánica: Desplazarse de izquierda a derecha.

- **Dinámica:** Desplazarse desde la posición inicial hasta la distancia máxima de desplazamiento, sin haber atacado a ningún enemigo, provoca entrar en un estado de noqueo por un breve periodo de tiempo, impidiendo realizar cualquier otra acción.
  - **Estética:** Los jugadores se sienten indefensos por un breve periodo de tiempo, ya que no pueden realizar ninguna acción y se vuelven vulnerables a ataques de los enemigos, provocando el fin del juego.
- **Mecánica:** Atacar a un enemigo que se encuentre del lado izquierdo o derecho.
    - **Dinámica:** Desplazarse desde la posición inicial hasta la posición de un enemigo dentro de la distancia máxima de desplazamiento y derrotar a un enemigo, no penaliza al jugador con un estado de noqueo, sino que aumenta su puntuación, y le permite volver a realizar una acción.
    - **Estética:** Los jugadores experimentan una sensación de realización, ya que logran coordinar su vista y la pulsación de los botones para derrotar a un enemigo y aumentar la puntuación total. Además, se crea un desafío, porque a la vez que se generan nuevos enemigos y se realizan ataques exitosos contra enemigos, estas dos sensaciones aumentan por la coordinación y concentración del jugador. Sin embargo, existe un aumento constante de la barra de estado mental con cada ataque acertado, comenzando a generar preocupación o atención a este elemento para reducirlo y poder continuar realizando acciones.
  - **Mecánica:** Disminuir la barra de estado mental.
    - **Dinámica:** Mantener la barra de estado mental en niveles bajos, permite seguir realizando acciones como desplazarse o atacar.
    - **Estética:** Los jugadores experimentan una sensación de control y estabilidad, que les permite seguir jugando de manera normal, sin embargo, al estar la barra en el punto máximo y siendo esta una representación simbólica del estrés del personaje, el jugador experimenta una sensación de descontrol y debilidad, provocando en el personaje entrar en un estado de inestabilidad, impidiendo realizar cualquier acción y volviéndolo vulnerable a ataques

enemigos, pudiendo provocar el final del juego. Este estado dura un breve periodo de tiempo, y al mismo tiempo la barra de estado mental disminuye. Además, el aumento o disminución de la barra de estado mental se ve reforzado por ciertos efectos visuales que reflejan ciertos síntomas de las personas que pueden experimentar un ataque de pánico.

**Cámara y Fondo del Nivel.** Al ser un proyecto totalmente en 2D y al utilizar sprites y no modelos 3D, se optó por elegir una cámara 2D ortográfica, ya que no se trabajará con elementos con perspectiva.

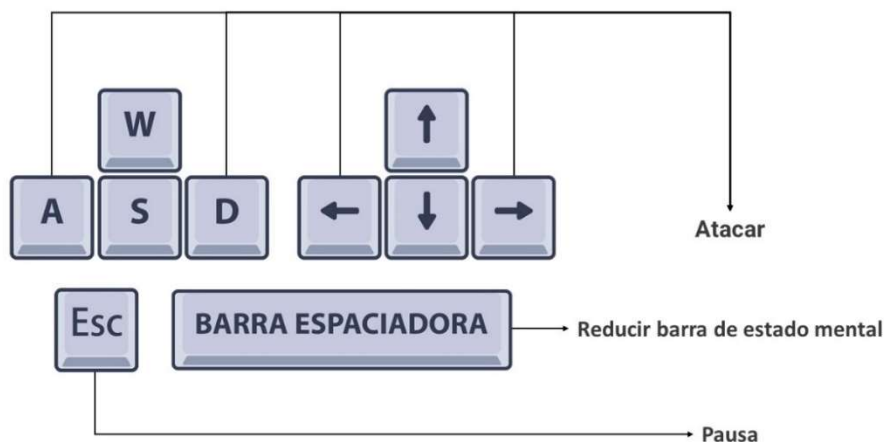
En el nivel, la cámara está ubicada ligeramente alejada del personaje, brindando un campo visual amplio, capaz de visualizar a los enemigos con cierta antelación para poder atacarlos, además de ofrecer una vista general del panorama del nivel. La cámara sigue constantemente al personaje principal durante sus ataques o desplazamientos, y cuenta con un ligero desfase de movimiento, logrando así un efecto de retraso, lo que ayuda visualmente al movimiento del personaje a que se percibe como un movimiento más rápido de lo que es.

El background (fondo del nivel) está diseñado pensando en el efecto parallax (paralaje), en donde las partes del background están dibujadas en capas diferentes, desde los elementos más cercanos a la cámara como arbusto o plantas, después edificios o árboles, montañas, hasta el cielo y las nubes; y una vez teniendo cada elemento en capas separadas, cada una se moverá a diferentes velocidades, siendo las capas más lejanas a la cámara las que se moverán a una velocidad menor, y las más cercanas a una velocidad mayor, logrando crear un efecto de profundidad. Este efecto es muy efectivo en videojuegos basados en desplazamientos laterales y logran brindar al videojuego una estética más interesante y agradable a la vista.

La cámara cuenta con ciertos efectos visuales que refuerzan el cambio de la barra de estado mental, logrando generar en el jugador una mejor inmersión dentro de la experiencia, estos efectos son el de una viñeta oscura que limita la visibilidad de la pantalla aumentando la dificultad del nivel, un ligero tambaleo en la cámara representando la desorientación, y un cambio en la paleta de color general del nivel, a una escala de grises, para reforzar el cambio de estado que está sufriendo el personaje por el aumento del estrés.

**Controles.** Al tener mecánicas relativamente sencillas que requieren de tres botones para ejecutarse y fáciles de implementar en cualquier plataforma de distribución, los controles se pensaron para ser fáciles de entender y adaptar, por lo que, para la plataforma elegida, los controles se muestran en la Figura 1.

*Figura 1 Esquema de controles para Windows*



**Puntuación.** Al ser un juego que se enfoca en un solo nivel, la rejugabilidad es un punto importante a tomar en cuenta para que el jugador siga interesado por continuar jugando tras terminar una partida.

Para incentivar la rejugabilidad existe un registro de la puntuación más alta alcanzada en una partida, la cual simboliza el número de enemigos derrotados. Si bien no existe una tabla de posiciones que registre el nombre y su puntuación, al final del nivel aparece la puntuación más alta registrada en todas las sesiones de juego y de la puntuación alcanzada en esa sesión, y de ser superada, esta se actualiza.

**Niveles.** El videojuego únicamente cuenta con un nivel y está diseñado para incentivar la rejugabilidad.

**Nivel Infinito.** El diseño básico del nivel es el siguiente:

- Título del Nivel: Modo infinito.

- **Encuentro:** El nivel es un modo infinito ambientado en un mirador a las afueras de una ciudad, en el cual aparecen constantemente enemigos. El objetivo del nivel es obtener la puntuación más alta. El nivel termina cuando somos derrotados. El acceso al nivel es desde el menú principal y no tiene restricción de acceso, por lo que se podrá entrar a este un número ilimitado de veces.
- **Objetivos:** Derrotar al mayor número de enemigos.
- **Progreso:** Al terminar el nivel se le informa al jugador la puntuación alcanzada en la sesión de juego y la puntuación máxima registrada de todas las sesiones previamente jugadas, de ser superada esta puntuación, se actualiza.
- **Personajes:** El único personaje jugable dentro del nivel es Jacobo.
- **Enemigos:** Únicamente existe un tipo de enemigo, cuyo objetivo es seguir al jugador y derrotarlo.
- **Música y Efectos de Sonido:** La música del nivel es del género 8 bits, música característica de los antiguos videojuegos de los años 80s. La música está compuesta en forma de loop, lo que permite reproducirse varias veces y logra dar un efecto de continuidad a la composición. Tanto el personaje principal como los enemigos cuentan con efectos de sonido al golpear.

**Estados del Juego e Interfaz de Usuario (IU).** El videojuego cuenta con un total de siete estados de juego, cada uno con un diseño de interfaz de usuario. Los estados e interfaces son los siguientes:

- **Menú principal:** En este estado se encuentran tres botones, el primero es para entrar al nivel, el segundo ejecuta el tutorial donde se explica el objetivo del juego y los controles, y el ultimo botón es el de los créditos, en el que se encuentran las atribuciones correspondientes del proyecto. Para pasar a cualquiera de estos estados es necesario pulsar con el botón izquierdo del mouse los botones que direccionen al estado seleccionado.

- IU: El fondo es una ilustración de unas montañas, un bosque y un cielo azul. En la parte superior se encuentra el título del videojuego. En la parte central se encuentran los tres botones organizados de izquierda a derecha y con una separación equidistante.

*Figura 2 Boceto del menú principal*

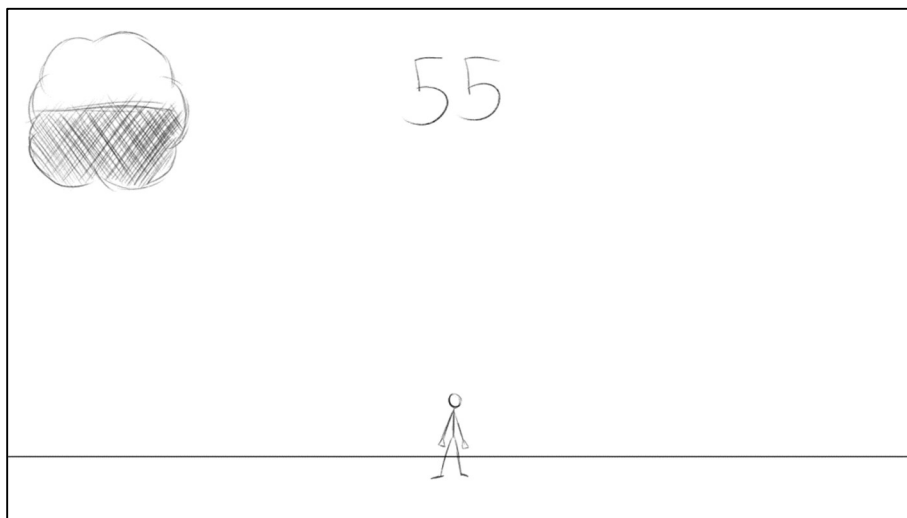


- Nivel: Es el estado en donde el jugador aplica los conocimientos aprendidos en el tutorial del juego e intenta cumplir con el objetivo del nivel. Durante la partida de juego es posible cambiar al estado de menú de pausa presionando la tecla ESC en cualquier momento de la partida.
- IU: La interfaz dentro del nivel está compuesta por un medidor m, el cual representa el estado mental del personaje, el diseño tiene un contorno negro y el interior es oscuro, el medidor que indica el nivel de estrés es de color blanco. Todos los elementos esta pensados para poder diferenciarse más rápido del fondo. Esta barra de estado mental está ubicada en la esquina superior izquierda y es ligeramente más grande que el diseño del personaje, esto para que sea visible de forma más cómoda y rápida.



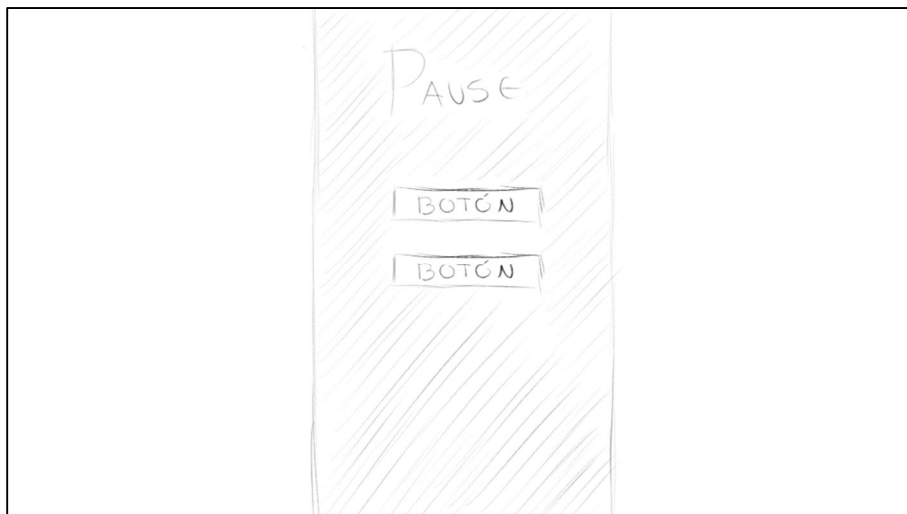
En la parte superior central se encuentra el contador de enemigos derrotados, es ligeramente más pequeña que la barra de estado mental. La fuente es de color blanco con una pequeña sombra paralela para que sea más fácil distinguirla.

*Figura 3 Boceto del nivel*



- **Menú de pausa:** El menú de pausa congela el tiempo de juego y despliega un panel con dos botones para poder cambiar de estado, el primero permite continuar con la partida de juego, y el segundo botón cambia al menú principal, saliendo del nivel. También es posible retirar el menú de pausa para volver al nivel presionando nuevamente la tecla ESC.
- **IU:** El menú aparece en medio de la pantalla, en un panel ligeramente transparente, estrecho de los laterales y su tamaño ocupa únicamente la parte central de la pantalla. Dentro del panel se encuentra el nombre del menú, y en la parte central los dos botones separados.

Figura 4 Boceto del menú de pausa



- Pantalla de game over (fin del nivel): Al finalizar el nivel se mostrará la puntuación alcanzada y la puntuación máxima lograda hasta ese momento. También hay dos botones para cambiar de estados, el primero es el de reintentar el nivel, y el segundo lleva al menú principal.
- IU: La pantalla aparece como un panel sobre el nivel, con el mismo formato que el estado de menú de pausa, en la parte central aparece la puntuación más alta, y debajo de esta la puntuación lograda en la partida jugada. La puntuación más alta esta se actualiza si es superada. En la parte inferior central encontramos dos botones, el primero es el de reintentar el nivel, y el segundo redirige al menú principal.

Figura 5 Boceto de pantalla de game over

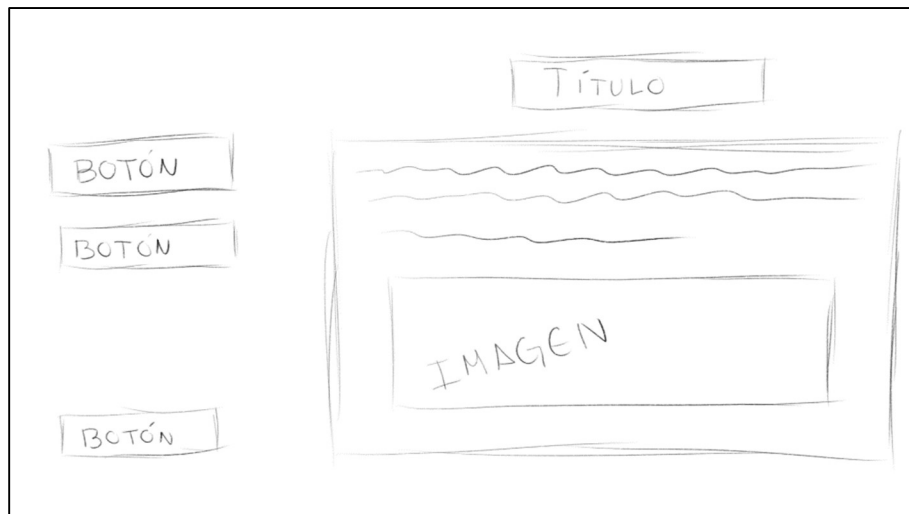


- Tutorial: El tutorial es una guía visual en la cual se le enseña al jugador mediante una ilustración y texto explicativo, los controles y objetivo del juego. Existen tres botones ubicados en el lateral izquierdo de la pantalla, el primero despliega el objetivo del juego, el segundo muestra los controles para poder jugar y el tercero manda al menú principal.

■ IU: El tutorial tiene el mismo fondo que el menú principal. Es una plantilla que consta de tres botones organizados de abajo a arriba en la parte de la izquierda, y dos paneles, el primero es para mostrar el título del tema del tutorial, y debajo de este va el segundo que contiene la parte que explica el tema.

Sobre esta plantilla se explican el objetivo del juego y los controles, ambos acompañados de una imagen, ya sea una captura del nivel o los diagramas de los controles.

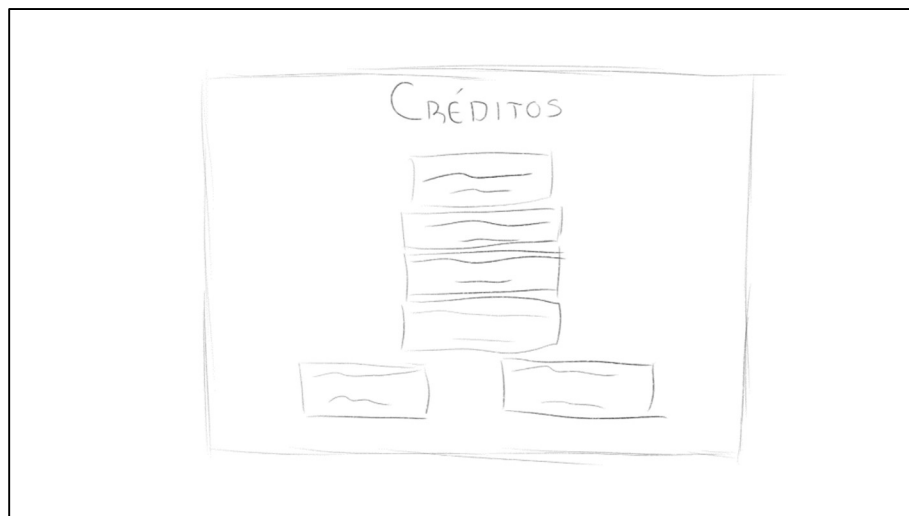
Figura 6 Boceto de tutorial



- Pantalla de créditos: En este estado aparecen los datos técnicos del desarrollo del videojuego como el nombre del desarrollador, atribuciones de material y créditos de música.

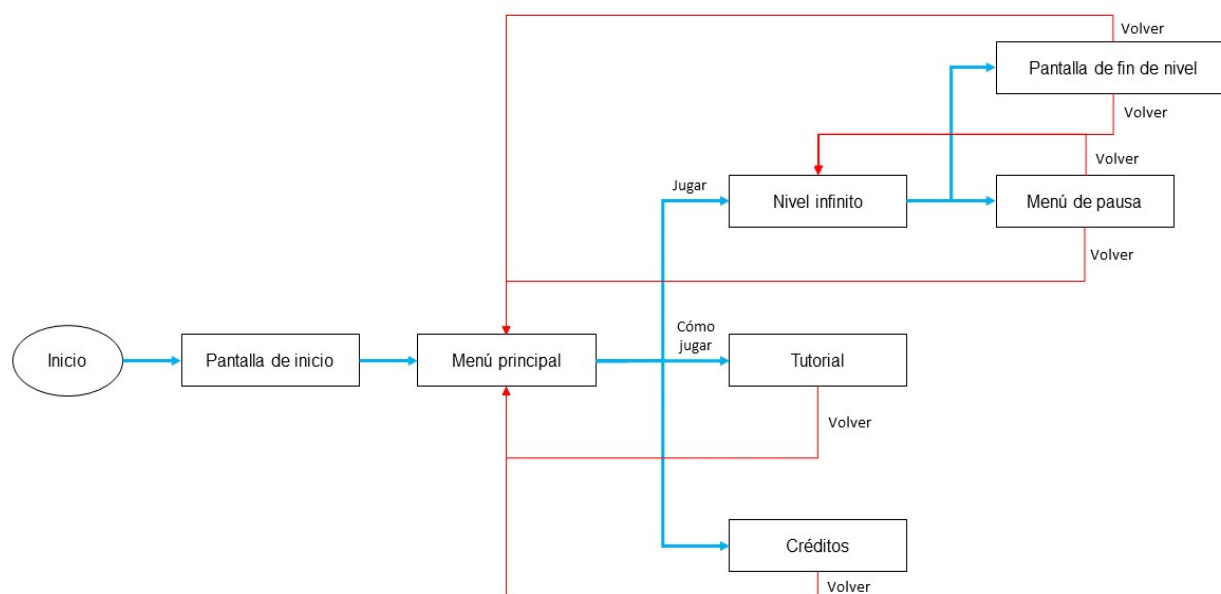
■ IU: Los créditos utilizan el mismo fondo del menú principal. En la parte central de la pantalla se encuentra un panel cuadrado de color morado ligeramente transparente y en él la información de los créditos.

Figura 7 Boceto de pantalla de créditos



**Progreso del Juego.** En este diagrama se representa el flujo de interacciones que tiene el jugador con los estados del videojuego.

*Figura 8 Diagrama del flujo de interacciones de los estados del videojuego*



**Personajes.** El videojuego cuenta únicamente con un personaje jugable y un enemigo. Las imágenes de los personajes pertenecen a un proceso desarrollado posteriormente en el proyecto, pero se anexan en este apartado para acompañar la descripción de cada personaje respectivamente. El desarrollo visual de los personajes se aborda en el apartado de “Desarrollo del Apartado Visual y Animación”.

***Personaje Principal.*** Diseño del personaje principal.

- Nombre del personaje
  - Jacobo.
- Descripción física
  - Edad entre 15 - 16 años.
  - Estatura: 1.60 m. - 1.65 m.
  - Complexión Ectomorfa y delgada. Con una postura ligeramente encorvada, casi imperceptible.
  - Tez de piel clara.
  - Forma del rostro entre redondo y ovalado.
  - Rasgos finos.
  - Cabello lacio, de color negro y corto. Un poco alborotado, peinado de lado y con uno(s) mechón(es) que cubren su frente.
  - Ojos son de color café oscuro (Casi negros), grandes y ligeramente rasgados.
- Vestimenta
  - Camisa blanca tipo neru de manga larga; tanto el cuello como las mangas siempre están abotonadas y ajustadas hasta el último botón. Ligeramente holgada y fajada.
  - Encima de la camisa usa el uniforme (Películas: Duna 1984 y 2021) de tela tipo ripstop y gabardina, con un cuello alto y un grabado del grupo al que pertenece. Tipo charreteras integradas al uniforme, que definen la parte superior de los trapecios y hombro. Las mangas cuentan con ligeras decoraciones en color plateado. El uniforme llega un poco debajo de la mitad de la pelvis y usa un cinturón de color negro con chapa plateada que ajusta el uniforme a la cintura. El color del uniforme es un azul oscuro. No cuenta con ningún distintivo, medallas o condecoraciones.

- Pantalón de la misma tela del uniforme. Ligeramente holgado y de color gris metálico. La parte superior no es visible por la altura del corte del uniforme, y la altura del pantalón llega un poco debajo del tobillo, pero esta parte no es visible debido a las botas.
- Botas militares negras lisas que llegan hasta por debajo de la rodilla. Estas cubren la parte inferior del pantalón. Tienen una punta fina y con un pequeño tacón.
- Descripción psicológica
  - Arquetipos de Jung
    - El cuidador: Es colaborativo, procura brindar ayuda tanto a personas cercanas a él como a desconocidos.
    - El sabio: Sobre piensa las situaciones para evitar panoramas que pongan en riesgo a las personas, sin embargo, le es difícil tomar decisiones debido al miedo. Además, es una persona curiosa y con ánimos de aprender cosas nuevas todos los días.
    - El héroe: Los miedos de su presente y pasado lo limitan a explotar su potencial como persona, y no le permiten ver en lo que puede llegar a convertirse.
- Habilidades
  - Jacobo posee una energía oscura que puede materializarse fuera de su cuerpo y esta energía no puede controlarla, y por eso al momento de realizar algún ataque hacia un enemigo, él proyecta en sus puños parte esa energía.

*Figura 9 Diseño visual del personaje principal*



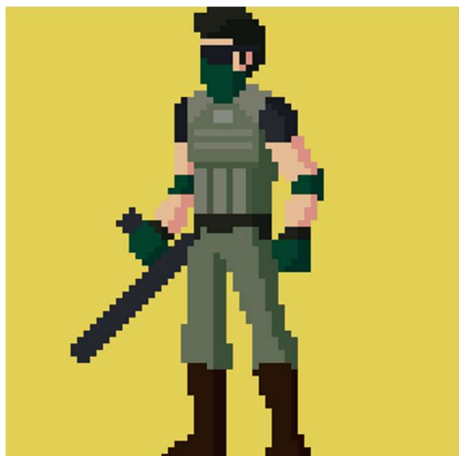
***Enemigo.*** Diseño de personaje del enemigo.

- Nombre
  - Mercenario.
- Descripción física
  - Edad entre 35 - 40 años.
  - Estatura: 1.70 m. - 1.75 m.
  - Complexión Mesomorfo.
  - Tez de piel clara.
  - Forma del rostro entre cuadrado o diamante. Tiene rasgos marcados.
  - Cabello quebrado, de color café oscuro y corto. Peinado hacia atrás con un ligero copete que mantiene despejada la frente.
  - Ojos de color café obscuro y pequeños.
- Vestimenta
  - Gafas tácticas negras sencillas, con un armazón de color café obscuro.
  - Playera negra de cuello redondo de manga corta; ajustada y fajada.
  - Pantalón color beige, tipo militar. Ligeramente holgado. Utiliza un cinturón táctico, del cual sobresale una tonfa negra ubicada en el lateral del cinturón.
  - Guantes cafés oscuros tácticos militares.



- Utiliza botas militares negras robustas que llegan hasta la mitad de la pantorrilla.
- Descripción psicológica
  - Personas de carácter fuerte, intimidantes y fieles a sus propias convicciones. Poco devotos a la causa y se limitan únicamente a cumplir órdenes, por lo que difícilmente toman decisiones por sí mismos.
- Encuentro.
  - Los enemigos son una plantilla, sobre la cual se van creando nuevos enemigos durante el nivel.
- Habilidades
  - Los enemigos no tienen habilidades, y su única forma de atacar al personaje es mediante ataques cuerpo a cuerpo con su arma.
- Armas
  - Tonfa.

*Figura 10 Diseño visual del enemigo*



**Narrativa.** Cuando descubre que tiene poderes sobrenaturales, un adolescente con trastorno de pánico deberá luchar contra una asociación delictiva que amenaza con capturar y explotar a la nueva generación de superhumanos; al mismo tiempo que lucha por sobrellevar su ansiedad y mejorar su salud mental.

**Estética y Estilo Artístico.** El desarrollo del apartado estético se basó en dos apartados importantes, el primero fue cierta sintomatología de las personas que viven con el trastorno de pánico; la sensación de peligro, mareos, dolor en el pecho y temblores o palpitaciones, estos síntomas se quisieron plasmar y adaptar a las mecánicas de juego, como la barra de estado o ciertas animaciones específicas, esto para poder transmitir al jugador ciertas sensaciones que experimentan las personas que padecen este tipo de aflicciones y sensibilizar al jugador mediante su interacción con el personaje.

Para el apartado visual se tomó como inspiración el pixel art, estilo que vio su origen en las primeras consolas de videojuegos de los años 80 y 90, donde las limitaciones del software y hardware obligaban a los departamentos de arte a dibujar sprites (mapa de bits) con un gran nivel de abstracción para poder plasmar a los personajes y entornos; se eligió el pixel art debido a su fácil producción a nivel de animación y escenarios, por generar visualmente un sentimiento de nostalgia o añoranza por las antiguas consolas de videojuegos, y por ser un estilo que logra evocar ideas y emociones con pocos recursos artísticos.

**Música y Sonido.** Utilizar tanto música como sonidos libres de derechos de autor.

**Miembros del Equipo.** Proyecto desarrollado por Yoltic S. Camacho Figueroa. Uso de material de terceros con respectivas atribuciones dentro del videojuego.

### **7.1.2.    *Diseño de Nivel***

El nivel comienza con el jugador ubicado en la parte central de la pantalla, y de los laterales comenzarán a llegar oleadas de enemigos, y el jugador deberá derrotar al mayor número posible de estos.

## Elementos de Diseño de Niveles

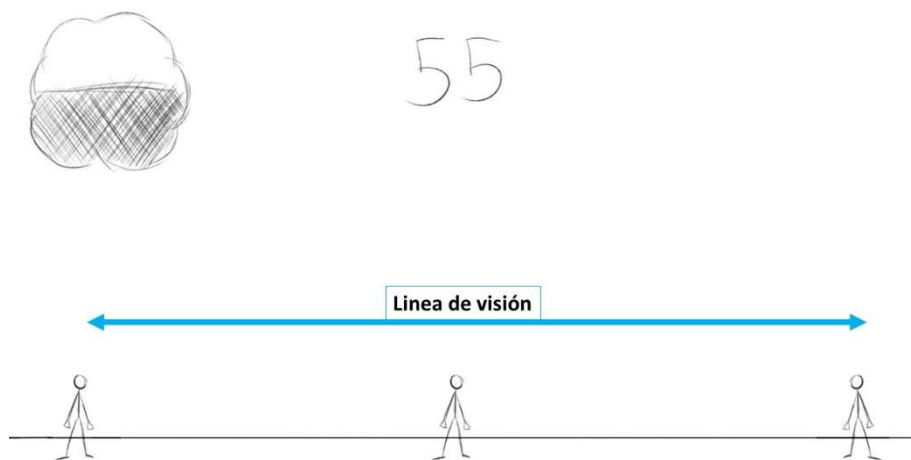
- **Requerimientos Funcionales**
  - El videojuego cuenta con un movimiento ilimitado hacia el lado izquierdo y derecho, esto por tres elementos, el primero es la distancia máxima de desplazamiento que tiene el jugador para poder atacar; el segundo son los estados de recuperación, en donde el jugador necesita estar quieto por unos segundos para poder volver a realizar alguna acción; y la tercera son los enemigos, los cuales aparecerán en forma de oleadas, limitando así el movimiento en ambas direcciones.
- **Usabilidad**
  - El nivel guía al jugador a ser proactivo dentro de la partida constantemente, ya que al derrotar a cada enemigo se irá llenando la barra de estado mental del personaje, y el jugador deberá estar atento de controlar dicha barra para poder continuar jugando; además, de estar al pendiente de los enemigos que aparecerán en las dos direcciones, en este caso, los indicadores sobre los enemigos ayudaran a definir el alcance al que estos están y si es posible atacarlos o no.
- **Deleite**
  - El diseño del nivel destaca gracias al combate y la experiencia de juego, ya que al efectuar coordinadamente y correctamente una secuencia de golpes que derroten a los enemigos, el jugador sentirá satisfacción gracias a su coordinación y además podrá seguir realizando otras acciones. Esta situación atiende en parte a la temática de los superhéroes, por las habilidades de combate; y la experiencia de juego que se logra gracias a la unión de las mecánicas de combate con la de reducir la barra de estado mental, logrando de esta manera una conexión mecánica y visual con el jugador sobre cómo es experimentar un ataque de pánico cuando no se controla el estrés, esto debido al constante uso de las mecánicas de combate, las cuales desgastan mentalmente al personaje y terminan por debilitarlo, volviéndolo vulnerable ante ataques de enemigos, sin embargo con el paso del tiempo el estrés se reduce y es posible volver a jugar de manera normal, es por ello que todas las

mecánicas deben coordinarse para lograr una estabilidad durante la partida de juego.

### Elementos Históricos de Arquitectura

- Líneas de visión
  - El nivel tiene una línea horizontal invisible sobre la cual está diseñado todo el escenario del nivel y en ella se mueven los enemigos, el entorno y el personaje.

*Figura 11 Línea de visión del nivel*



- Sistema de representación
  - La zona de campo no está llena de pasto, hay un camino de tierra sobre el cual el nivel se desarrollará, próximo a este camino se encuentran bastantes árboles altos, impidiendo la visibilidad de cercanías. En la mitad del escenario se encuentra un gran río que divide la zona, y a lo lejos es posible ver más árboles, montañas, y nubes. El nivel pretende transmitir al jugador que se encuentra en una zona lejana y donde no se encuentran civiles cerca, un lugar ideal para los enemigos por no haber testigos cerca y para acorralar al personaje.
- Hacer una declaración
  - El significado del nivel es la dualidad, ya que puede transmitir seguridad y libertad al ser una zona despejada y sin civiles cerca, en donde el personaje principal se siente con la seguridad y confianza de que no lastimara a nadie

inocentes; pero también el nivel transmite inseguridad y vulnerabilidad por conforme se desarrolla el juego y el personaje se enfrenta a diversos retos y amenazas, generando sentimientos de persecución y acorralamiento.

- Relaciones espaciales y simbólicas
  - El entorno es un lugar abierto que no cambia de estado, por lo que la jugabilidad no se ve tampoco alterada por el entorno.

### **Arreglos Espaciales**

- Figure ground
  - El espacio positivo es todo el camino de tierra horizontal, ya que sobre él es donde podemos desplazarnos, atacar, quedar noqueados, reducir nuestra barra de estado mental, entre otras cosas. Los espacios negativos no existen propiamente en el videojuego, ya que no existe límite en el juego que impida moverse sobre el espacio positivo, sin embargo los enemigos pueden considerarse espacios negativos en cierto momento, ya que al aparecer en los laterales de la pantalla y no poder atacarlos debido a la lejanía del personaje, se vuelven obstáculos dentro del espacio, pero al acercarse al enemigo estos enemigos pueden ser destruidos y dejan de representar una barrera en el espacio para poder continuar derrotando más enemigos.
- Form void
  - Para el diseño del nivel no se ocupó form void, dado que al ser un videojuego en 2D al momento de estructurar el nivel no se ocuparon elementos 3D para su maquetación, sino que se realizaron bocetos digitales de la disposición de los elementos dentro del nivel.
- Arrivals
  - Al inicio del nivel el jugador se encontrará en medio del terreno del escenario y tendrá gran visibilidad de los laterales para poder prepararse a los enemigos que aparecerán.

- Genius loci

- La experiencia que se busca generar al jugador es la de inseguridad, por una parte, por los enemigos que aparecen constantemente y que buscan atacarnos, pero la razón principal es por la atención constante que se debe prestar a la generación constante de estrés al derrotar enemigos y que se va acumulando en la barra de estado mental, y que, de no controlar, desencadena en un ataque de pánico, incapacitando al jugador de poder realizar cualquier acción hasta que se reduzca la barra de estrés,

**Estructura del Espacio de Juego.** El nivel tiene una estructura de laberinto sumamente sencilla, debido a que, si bien el objetivo de la partida es derrotar a la mayor cantidad de enemigos sin ser derrotado, existen mecánicas que puede usar el jugador de diferentes formas para lograr dicho objetivo, logrando una jugabilidad diferente en cada sesión de juego, convirtiendo el objetivo del nivel de algo lineal, en algo complejo cuando se le dan las mecánicas adecuadas al jugador para explorar nuevas formas de jugar un nivel.

### **Tipos de Tamaño Espacial**

- Espacio estrecho

- El espacio estrecho es implementa al momento de que el jugador no controla la barra de estado mental y deja que suba, provocando un cambio visual en la pantalla en donde aparece una viñeta de color negro que va cubriendo poco a poco la zona de juego, creando un cambio en la jugabilidad del jugador ya que esta viñeta limita visualmente la pantalla, provocando que no se puedan antelar la aparición de los enemigos y es más difícil coordinar los ataques y evitar ser derrotado.

- Espacio íntimo

- El nivel desarrolla este tipo espacio cuando el jugador se encuentra estable para poder pelear, el campo de visión es amplio, proporcionando un control del terreno y del combate, ya que puede prepararse con antelación para realizar cualquier acción que el jugador quiera realizar.

- **Espacio de prospección**

- Este tipo de espacio se desarrolla en la mayor parte del nivel, porque al estar el jugador en medio del campo de pelea, se vuelve vulnerable para ser acorralado por los enemigos en ambas direcciones; también cuando se ejecuta un ataque y no logra conectar con un enemigo, él no puede realizar ninguna acción porque el personaje quedó noqueado y el jugador no puede hacer nada; y finalmente cuando la barra de estado mental llega a su punto máximo y el personaje experimenta un ataque de ansiedad y no puede hacer absolutamente nada hasta que se reduzca el estrés.

**Consideraciones de la Cámara.** La cámara es de tipo ortográfica y no es controlable por el jugador, pero esta sigue en todo momento al personaje. Esta cuenta con un pequeño desfase de movimiento cuando el personaje se desplaza a otra posición, generando un efecto que logra aparentar una mayor velocidad y potencia del desplazamiento.

Los elementos del fondo al estar en capas diferentes y ser diseñados pensados para usarse con el efecto parallax, es posible crear un efecto de profundidad en el nivel, a pesar de no ser una cámara que capte la perspectiva.

**Recompensas.** El nivel recompensa al jugador de tres formas, la primera es al ejecutar ataques que conecten con los enemigos, proporcionando al jugador la oportunidad de seguir realizando más ataques sin ninguna penalización, en comparación cuando no se conectan los ataques y el jugador es penalizado con un tiempo de espera por estar el personaje noqueado, la segunda es cuando el jugador es consciente y atento de la barra de estado mental y la reduce cada cierto tiempo para no alcanzar el punto máximo, dándole la oportunidad al jugador de poder seguir realizando ataques, y la última y la más visible, es mediante un puntaje que se va actualizando cada vez que el jugador derrota a un enemigo y que al terminar la partida, si es que el puntaje de la sesión de juego previa fue superado, este se actualiza con el puntaje logrado en la partida actual, brindando al jugador una experiencia de satisfacción por superar las puntuaciones anteriores.

## Narrativas

- Mecánicas como generador de diseño
  - Tanto el nivel como el videojuego se construyeron a partir de mecánicas de juego ya existentes, y sobre ellas se desarrolló y adaptó la narrativa del videojuego y el nivel.
  
- Narrativa Incrustada
  - La historia gira en torno a Jacobo, un joven que posee poderes sobrenaturales y sufre de trastorno de pánico y él debe aprender a controlarlos ambos para poder proteger la ciudad de un grupo de criminales que buscan destruirla.

El nivel se basa en un determinado punto de la historia, donde Jacobo es acorralado en una zona alejada de la ciudad por un gran número de criminales que buscan capturarlo, sin embargo, Jacobo opone resistencia y lucha hasta derrotar al último enemigo que intente lastimarlo.
  
- Espacios Evocadores
  - El nivel se desarrolla en un mirador ubicado en el bosque, la zona está dividida por un gran río, árboles de gran altura cubren gran parte del lugar, y hay grandes montañas al fondo. Este es un lugar acogedor y pacífico, donde impera cierta seguridad y tranquilidad por ser un lugar apartado de la civilización.
  
- Espacios de escena
  - Tanto las montañas como, el bosque y el río son espacios que destacan sobre todo el escenario, ya que, desde la ubicación del personaje, se puede observar todo el paisaje de la zona, brindando una gran vista contemplativa.
  
- Espacios incrustados
  - El escenario está inspirado en las inmediaciones del volcán Popocatepetl y La Malinche.



## 7.2. Producción

La fase de producción se dividió en tres etapas, programación, desarrollo del apartado visual y animación, y música y sonido; el desarrollo de estas etapas no se hizo de manera simultánea, cada una de estas tenía un tiempo de desarrollo aproximado de tres a cuatro meses, y al término de las actividades correspondientes de la etapa, se procedió a trabajar en la siguiente etapa.

### 7.2.1. Programación

El videojuego está desarrollado en el motor de videojuegos Unity en su versión 2020.3.24f1, el lenguaje de programación utilizado fue C# y fue desarrollado en Visual studio 2019. Se utilizó la plantilla predefinida para proyectos 2D de Unity.

El apartado de programación fue dividido en 5 secciones, cada una con un desglose de las mecánicas o funcionalidades a programar. Cada sección se desarrolló individualmente, y al término de una se proseguía a trabajar en la siguiente:

- Escenas
- Manager y controladores
- Jugador
- Enemigos
- Fondo y Efecto Paralaje (Parallax)
- Animaciones
- Interfaz de usuario
- Condición de derrota y clasificación

**Escenas.** El videojuego cuenta con un total de seis escenas, las cuales son:

- MainMenu: Es la escena inicial del videojuego, en ella se encuentra el logotipo del juego, y tres botones, uno abre una partida de juego, otro un pequeño tutorial, y el último abre un panel sobre el cual se muestran los créditos correspondientes al desarrollo del videojuego.
- SceneManager: Es una escena sin cámara, que tiene un objeto vacío llamado SceneManager, esta cuenta varios scripts que se encargan de controlar el

cambio de escenas dentro del juego, logrando transiciones y cambios de escenas fluidos, además de crear un sistema modular que permite mantener separado comportamientos específicos dentro del juego en diferentes escenas y que pueden ser ejecutadas de forma simultánea.

- **Gameplay:** Nuevamente es una escena sin cámara, y cuenta con un UIManager que se encarga del funcionamiento del contador de enemigos derrotados en la partida. Además, cuenta con un Canvas que contiene el HUD, el menú de pausa, y la pantalla de game over.
- **Level:** Es la escena donde el juego se desarrolla y se encuentra el funcionamiento general de las mecánicas de juego del jugador y de los enemigos. Esta escena a diferencia de las anteriores si cuenta con una cámara que renderiza los elementos que se encuentren dentro de la escena.
- **Tutorial:** Es una escena que cuenta con tres botones, los dos primeros muestran en pantalla pequeñas explicaciones sobre el objetivo y controles del videojuego, y el tercer botón permite regresar a la escena MainMenu.
- **Level2:** Es la misma escena que Level, solo que al reintentar una partida cuando se es derrotado en Level, se carga esta escena exactamente igual, y viceversa cuando se pierde la partida en esta escena.

**Manager y Controladores.** Se implemento en el proyecto el asset de la Asset Store llamando Scriptable Object Architecture, el cual permitió crear un sistema de comunicación entre diferentes escenas mediante scriptable objects y sistema de eventos personalizables, logrando tener un proyecto modular y fácil de editar.

Los scripts encargados de guardar datos o realizar alguna función en específico dentro de diferentes escenas son los managers y controladores, para este proyecto se desarrollaron los siguientes:

- **GameManagerSO:** Es un scriptable object que se encarga de guardar los datos de los enemigos derrotados en partida, saber si el jugador ha sido derrotado, y contar con un canal de eventos personalizados sobre el cual es posible avisar a los scripts suscritos al evento en diferentes escenas. Además, cuenta con funciones para cambiar los estados del juego y cargar datos guardado.
- **GameStateSO:** Es un scriptable object que únicamente cuenta con una variable publica string llamada stateName, y una vez que se crea un asset a partir de este script, se le debe asignar el nombre de un estado de juego (game over, main menu, pause, playing) para poder funcionar con los otros scripts.
- **GameStateListener:** Es un script que está suscrito al canal de eventos personalizados, y este reacciona cuando se cambia de estado. Su funcionamiento consiste en activar o desactivar componentes MonoBehaviour o llamar eventos de unity, según el estado en que se encuentre el juego.
- **SceneInitializer:** Es un script que se encarga de cargar las escenas necesarias para el funcionamiento de otra escena.
- **SceneSO:** Es un scriptable object que únicamente cuenta con una variable publica string llamada sceneName, y una vez que se crea un asset a partir de este script, se le debe asignar el nombre de una escena cargada en build settings para poder funcionar con los otros scripts.
- **SceneLoaderManager:** Es el script que administra la carga y descarga de escenas dentro del proyecto, y que responde a un sistema de eventos personalizados.

**Jugador.** Las mecánicas y funcionalidades del jugador fueron las primeras en ser desarrolladas, las cuales fueron las siguientes:

- **GameObject**
- **Input**
- **Desplazamiento y ataque**

- Cámara de jugador
- Patrón de diseño State
- Barra estado mental

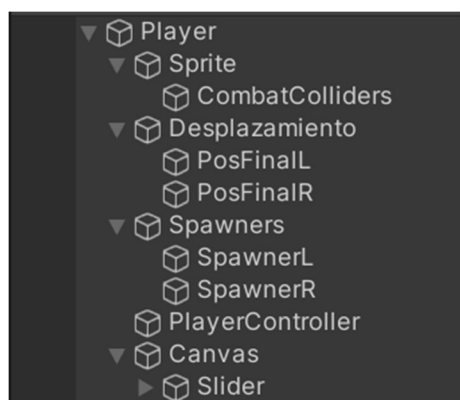
Los códigos de cada proceso están en el Anexo B. Códigos C#.

**GameObject.** El GameObject es el objeto que contiene el comportamiento general del jugador, la estructura de este GameObject fue la siguiente:

- Player
  - Este objeto contiene los componentes encargados del comportamiento general del enemigo, como el PlayerStateManager, GameStateListener, entre otros scripts.
- Sprite
  - Se encarga de renderizar los sprites del jugador en la escena, además de contar con el componente Animator que permitirá animar dichos sprites. También este objeto cuenta con un objeto hijo llamado CombatColliders, el cual cumple con la función de detectar las colisiones de combate realizadas por las animaciones del personaje.
- Desplazamiento
  - Es un objeto vacío que tiene dos objetos hijos, los cuales también son objetos vacíos, y su funcionalidad es definir en el espacio de juego la distancia máxima en la que el objeto Player se puede desplazar.
- Spawners
  - Es un objeto vacío que tiene dos objetos hijos, y su funcionalidad es definir en el espacio del punto sobre el cual se estarán generando los enemigos a lo largo de la partida.
- PlayerContoller

- Contiene un componente que se encarga de revisar los valores de la barra de estado mental, y avisar mediante un evento cuando la barra llega a su valor máximo.
- Canvas
  - Es el objeto sobre el cual los elementos UI deben estar, para ellos estos elementos deben ser objetos hijos de Canvas. En este objeto se encuentra un Slider el cual sirve para indicar el estado mental del personaje dentro de la partida.

*Figura 12 Estructura del GameObject Player*



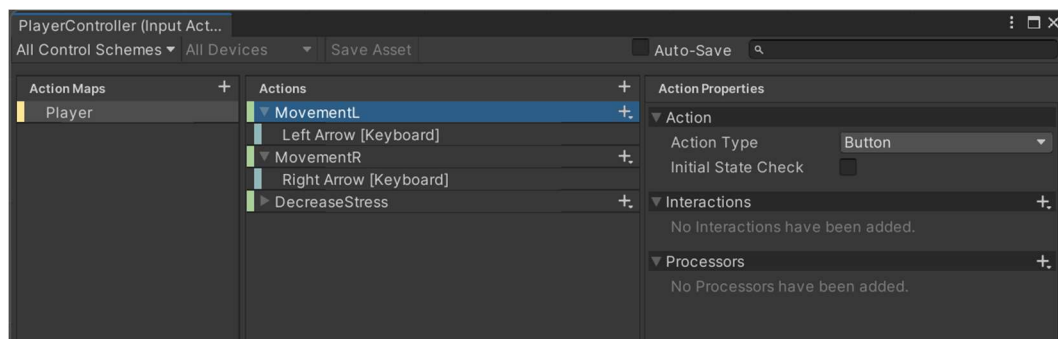
**Input.** Para la creación de los controles del videojuego se utilizó el nuevo sistema de input de Unity (new input system) ya que este ofrecía ciertas ventajas de optimización en relación a la antigua forma de recibir los inputs desde el Update de un código aparte.

Una de las ventajas del nuevo sistema de input es la creación de Input Actions, los cuales crean mapas de acciones específicos para diferentes comportamientos, dentro de estos mapas se crean las acciones a realizar, las cuales están a la espera de un input. Otro elemento importante del nuevo sistema de input es que funciona como un componente dentro del inspector, el cual se asigna al objeto que reaccionara a los inputs, y dentro del componente se desglosan todas las acciones de los Input Actions creados en forma de Unity Events.

Para el videojuego se creó un mapa de acciones llamado “Player”, entre las acciones a contener fueron “MovementL”, que tenía como input la tecla “A”, “MovementR” con la tecla “D”, y

“DecreaseStress” con la tecla “Space; posteriormente se les asignan comportamientos a los eventos de estas acciones.

*Figura 13 Mapa de acciones de Player*

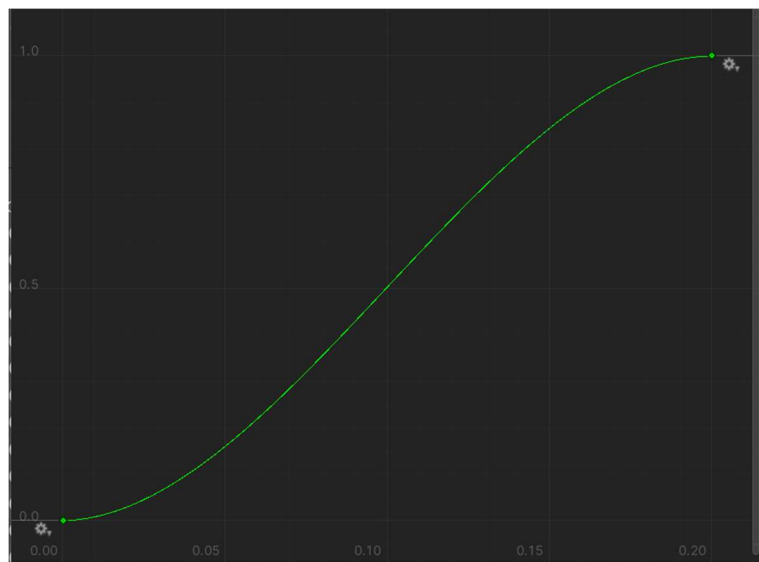


**Desplazamiento y Ataque.** Descripción y objetivo de la mecánica: Desplazar al personaje de su posición inicial a un punto predefinido sobre una línea recta horizontal con una aceleración variada.

Para mover un objeto de un punto a otro se decidió utilizar la función Lerp (interpolación lineal), la cual devuelve un valor entre otros dos valores (a & b), y el valor resultante está definido por otro valor (t), el cual es el encargado de interpolar los dos valores. Dentro del videojuego la función de Lerp se adaptó, donde “a” era la posición inicial del personaje, “b” la posición final, y el valor “t”, que representaría la aceleración, se definió usando una AnimationCurve; la cual es una colección de fotogramas clave, que puede almacenar valores y evaluarlos a través del tiempo. La gran ventaja de la AnimationCurve es que permite editar desde el Inspector de Unity los valores de los keyframes de la curva y su forma, facilitando la configuración de la aceleración fuera de código.

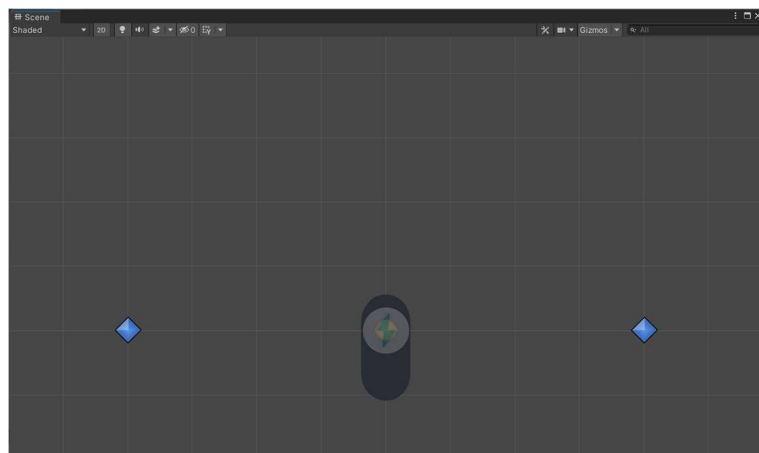
**(Anexo B)**

*Figura 14 Animation curve que describe la aceleración del personaje principal*



Dentro de la escena en Unity, se creó un objeto vacío llamado y etiquetado como “Player”, en este objeto se asignaron los scripts de lógica y componentes como Rigidbody2D y un Collider2D para el uso del motor de física de Unity. Dentro de este objeto se creó un objeto hijo llamado “PuntosDeDesplazamiento” y en él se crearon objetos vacíos, que sirvieron para definir el alcance máximo de desplazamiento del personaje, en 4 unidades.

*Figura 15 Alcance máximo de desplazamiento*



Descripción y objetivo de la mecánica: Desplazar al personaje de su posición inicial a un punto no predefinido sobre una línea recta horizontal con una aceleración variada.

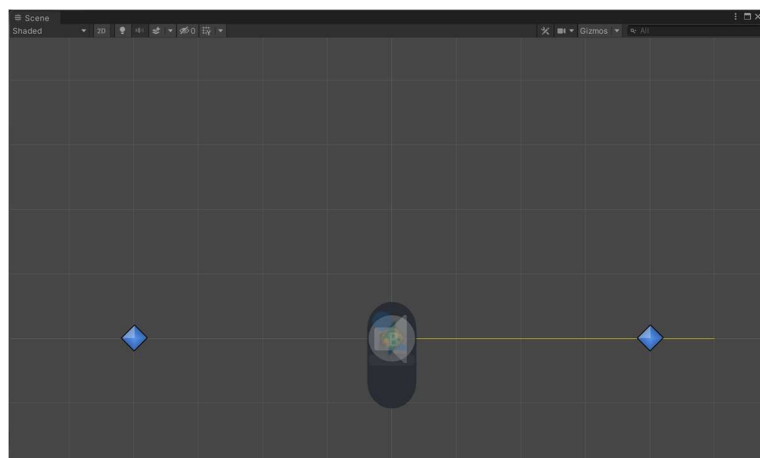
La mecánica de ataque comparte el mismo funcionamiento que la de desplazamiento, sin embargo, para esta nueva mecánica es necesario saber si hay un enemigo dentro del rango de movimiento.

Para programar el ataque del jugador se decidió utilizar un Raycast, el cual lanza un rayo desde el punto inicial, con una dirección y una distancia, que choca contra los colisionadores en la escena; la dirección del rayo se define por el input del botón presionado, "A" como un Vector2.Left y "D" como un Vector2.Right, y la distancia del rayo se definió en 5 unidades, esto para brindarle al jugador más alcance del que realmente tiene para brindarle un poco de ventaja. Una vez que se crea el rayo y choca con algún colisionador es necesario guardar la información del objeto de ese colisionador, para ello es necesario utilizar un RaycastHit2D, el cual regresa la información del objeto colisionado dentro del motor de física 2D.

La creación del Raycast no es visible en la escena y de esta forma es difícil editar el videojuego para detectar posibles errores, por lo que fue necesario dibujar un rayo con Debug.DrawRay.

**(Anexo B).**

*Figura 16 Alcance de desplazamiento definido por un Raycast*



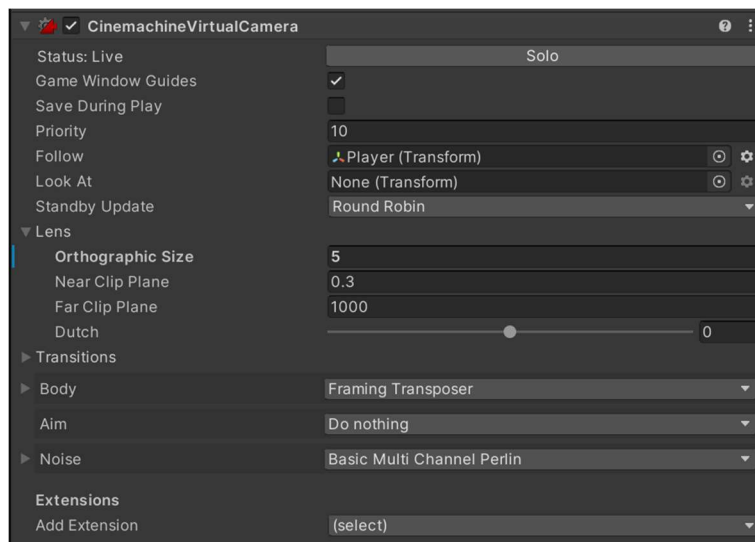
**Cámara de Jugador.** Descripción y objetivo de la mecánica: Desplazar en todo momento la cámara a la posición del personaje.



Para el comportamiento de la cámara del jugador se decidió utilizar Cinemachine, el cual es una suite con múltiples herramientas para operar cámaras dentro de Unity, facilitando su uso gracias los diversos componentes y comportamientos ya preconfigurados. Se creó un CinemachineBrain, que es el encargado de monitorear todas las cámaras en escena, y un CinemachineVirtualCamera, sobre la cual se editará y programará el comportamiento deseado; ambos son componentes editables en el Inspector y es posible editar ciertas propiedades desde código.

Para desplazar la cámara a la posición del personaje, se editaron diversas propiedades del componente CinemachineVirtualCamera, la primera fue el Framing Transposer, propiedad que le permite a la cámara moverse por el espacio de la pantalla y está optimizado para videojuegos 2D, y dentro su configuración se modificó el X Damping, lo que cambia la velocidad de respuesta de movimiento de la cámara, logrando así un efecto visual de potencia de ataque debido al retraso de la cámara al moverse; y la propiedad Follow, que recibe como parámetro un Transform, en este caso el GameObject Player, y se encargaría de seguir a este en todo momento del juego.

*Figura 17 Configuración de la CinemachineVirtualCamera*



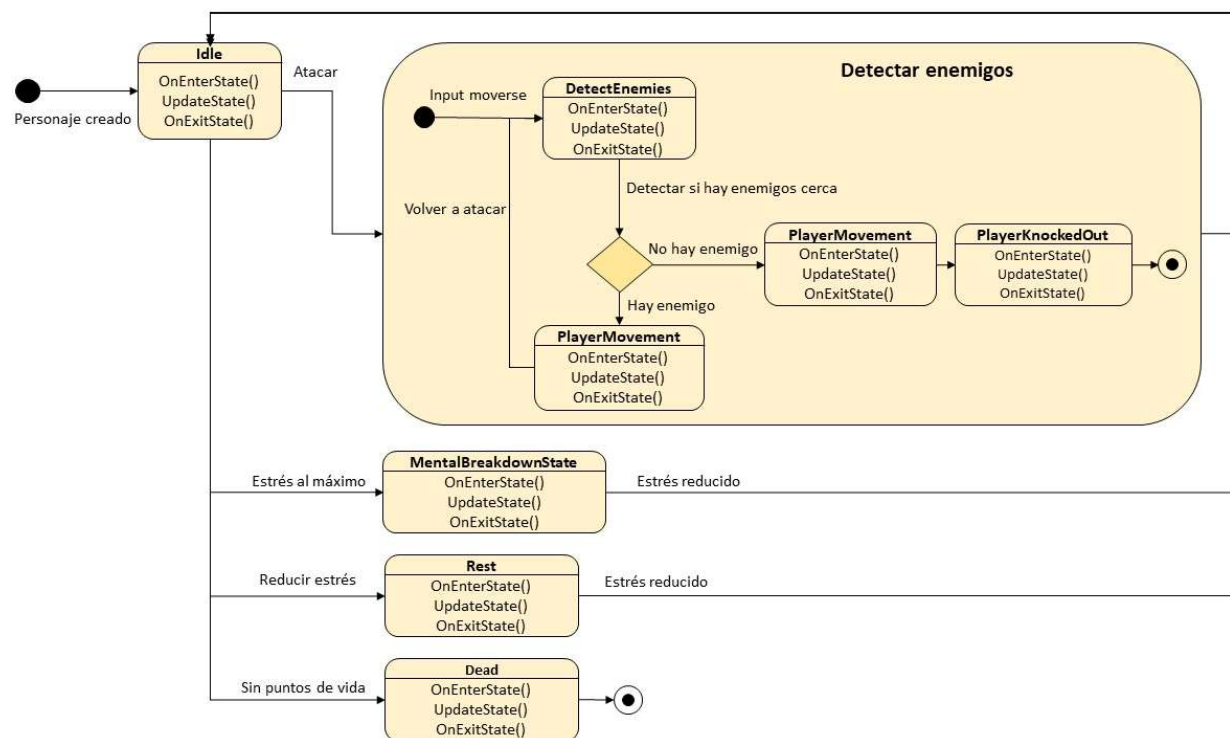
**Máquina de Estados Finita.** Descripción y objetivo de la mecánica: Crear una máquina de estados finita para los diferentes estados del personaje.

El patrón de diseño State (State pattern) es un patrón de diseño el cual es un modelo general el cual puede ser reutilizado para solucionar problemas de diseño presentes en diferentes situaciones.

Para llevar a cabo dicho patrón de diseño se implementó una máquina de estados finita, esto por dos razones, la primera fue por la facilidad de brindar este tipo de diseño cambiar el comportamiento del personaje de una forma sencilla en función del estado actual; la segunda fue debido a la necesidad propia de la jugabilidad, ya que el jugador debe tener la capacidad de presionar los inputs de movimiento y que estos se ejecuten en el orden en que fueron presionados, logrando así una secuencia de golpes similar a un combo.

Antes de programar la FSM se realizó un diagrama de estados, y su interacción entre sí, y una vez teniendo la idea general del funcionamiento de la FSM, se procedió a programarla.

Figura 18 Diagrama de máquina de estados finita



Lo primero que se realizó fue la creación de una clase abstracta llamada `PlayerBaseState`, clase que sirvió como plantilla para que de esta heredarán los scripts de los estados a programar.

Esta clase tiene 3 métodos OnEnterState, el cual se ejecuta al inicio cuando un estado transiciona a otro, UpdateState, que se ejecuta en cada frame, y OnExitState, que se ejecuta al final de un estado antes de transicionar a otro. Cada uno de estos métodos debe recibir una variable de tipo PlayerStateManager, el cual hereda de MonoBehaviour y se encarga de controlar todo el funcionamiento de la FSM (**Anexo B**).

Posteriormente se crearon los scripts de los estados, los cuales no heredan de MonoBehaviour, sino de PlayerBaseState (**Anexo B**).

El encargado de controlar la lógica y funcionamiento de la FSM es el script de PlayerStateManager, dentro de este script se creó una variable de tipo PlayerBaseState para guardar una referencia al estado actual, las instancias de los estados que tendrá el personaje, la lógica para ejecutar los métodos de OnEnterState y UpdateState, y un método propio para cambiar de estado.

Dentro del script de PlayerStateManager se crearon las variables y métodos necesarios para todo el funcionamiento de los estados, ya que estos al no heredar de MonoBehaviour, no tienen acceso a ciertas propiedades y métodos, sin embargo, PlayerStateManager si tiene acceso a estos y al ser padre de los estados, ellos tienen acceso a estos.

Una vez programada la FSM, se trasladó y adaptó todo el comportamiento del movimiento del personaje al estado MovementState, para que de esta forma cualquier comportamiento que tenga el personaje se controle desde la FSM.

Los estados en la FSM son 6, y cada uno tiene el siguiente comportamiento:

- IdleState
  - Es el estado predeterminado y al que vuelven todos los estados cumplidas ciertas condiciones. Checa si no cuenta con vida o si quedó noqueado, y así transicionar al estado correspondiente. Además, este revisa en cada momento los inputs que se registran y guardan en una lista, la cual almacena variables booleanas para saber si está en desplazamiento el jugador, la dirección del desplazamiento y si está activa esa solicitud.

- **DetectEnemiesState**
  - Este estado se encarga de lanzar un rayo en la dirección que tiene registrada en ese momento la lista de inputs, y detecta si existe un enemigo dentro del rango del rayo creado y de ser así, se guarda la información del enemigo. El estado siempre transiciona a MovementState.
- **MovementState**
  - El estado se encarga de trasladar desde la posición en que se encuentre el jugador hacia un punto en específico, ya sea un enemigo o el rango máximo hasta donde puede desplazarse. Este estado puede transicionar a varios estados, a MentalBreakDownState en el caso de que el estrés generado haya superado la barra de estado mental, a RestState cuando se desplaza y no golpea a ningún enemigo, a DetectEnemiesState si el personaje si golpeo a un enemigo y aún tiene elementos pendientes en la lista de inputs, o simplemente a IdleState cuando ninguna de las otras condiciones de transiciones se cumple.
- **RestState**
  - El estado se encarga de esperar un determinado de tiempo, en donde no es posible realizar ninguna acción. Se entra en este estado cuando se mueve el personaje y no hay enemigo. Al salir de este estado se borran los elementos de la lista de inputs y transiciona a IdleState.
- **DeadState**
  - Este estado avisa a un GameManager, script que controla lógica de toda la escena del videojuego, que el jugador ha muerto y pone la escala de tiempo del juego a 0, para pausar todo dentro de la escena.
- **MentalBreakdownState**
  - Este estado funciona igual que RestState, espera un determinado de tiempo en donde no se puede realizar ninguna acción, con la diferencia que este determina si el jugador deja de estar en dicho estado. Una vez terminado el tiempo de espera, transiciona a IdleState.

Cada uno de los scripts de los estados tiene comentado pseudocódigo para las animaciones del personaje, para ser sustituido por la implementación adecuada.

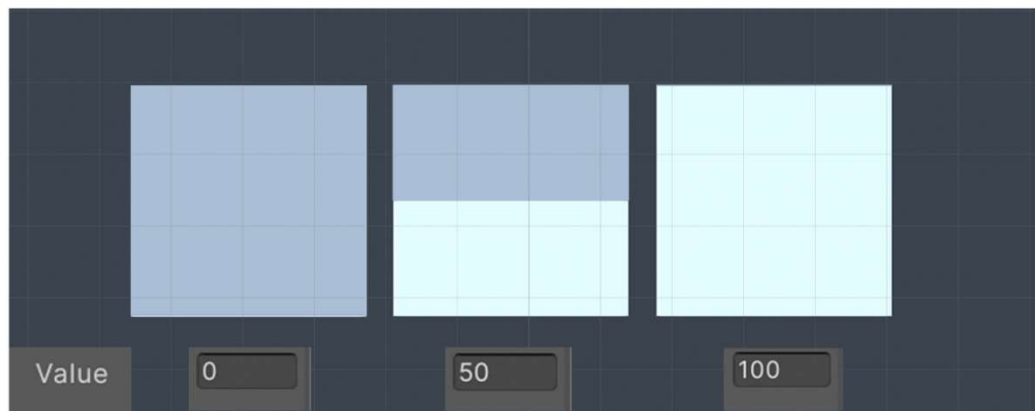
**Barra de Estado Mental.** Descripción y objetivo no.1: Crear un indicador que señale el estado de salud mental (estrés) del personaje.

Para programar la barra de estado mental se utilizó y modifiqué un Slider, el cual es un componente deslizante que puede moverse entre un valor mínimo y máximo, y gracias a su variable On Value Changed fue posible crear eventos que reaccionaran cuando el slider cambiara de valor.

El comportamiento del Slider se programó en un script diferente a la máquina de estado llamado “MentalBarController”, y este nuevo script se encarga de aumentar el estrés del personaje mediante una variable y definir si el personaje se encuentra en un estado de calma o inestabilidad; además de tener un método que se llama desde el Input Action de Player Controller, en la acción “DecreaseStress”, cuya función es la de reducir la barra de estado mental.

El primer evento creado para el Slider fue uno que se encargaba de modificar el acercamiento de la cámara conforme al valor del slider, limitando la visibilidad del jugador si no reducía la barra de estado mental de manera constante.

*Figura 19 Progreso del Slider del prototipo de la barra de estado mental*



**Enemigo.** Una vez programadas las mecánicas y funcionalidades básicas del personaje, se procedió a crear las del enemigo, las cuales fueron las siguientes:

- GameObject
- Inteligencia artificial conectable
- Sistema de oleadas

**GameObject.** Al igual que con el personaje principal, el GameObject es el objeto que contiene el comportamiento general del enemigo y su estructura fue la siguiente:

- Enemy
  - Este objeto contiene los componentes encargados del comportamiento general del enemigo, como el state controller y el enemy controller.
- Sprite
  - Se encarga de renderizar los sprites del enemigo en la escena, además de contar con el componente Animator que permitirá animar dichos sprites. También este objeto cuenta con un objeto hijo llamado CombatColliders, el cual cumple con la función de detectar las colisiones de combate realizadas por las animaciones del personaje.
- Player detector
  - Es un objeto que contiene un box collider 2D y un script que se encarga de detectar al jugador una vez que este entra en contacto con el colisionador 2D. El collider se ajusta a la máxima distancia en que el jugador puede atacar a un enemigo, y su principal función es activar un indicador en el enemigo que permite saber si es posible atacarlo.
- Object near
  - Este objeto también cuenta con un box collider 2D, y se encarga de detectar al jugador, con la diferencia que este se activa cuando este entra en contacto con él, pero a una distancia mucho menor.

- Canvas
  - Es el objeto sobre el cual los elementos UI deben estar, para ellos estos elementos deben ser objetos hijos de Canvas. En este objeto se encuentra un Image el cual se usa para indicar que el enemigo se encuentra al alcance del jugador para ser atacado.

Figura 20 Estructura del GameObject Enemy



**Inteligencia artificial conectable (Pluggable AI).** La inteligencia artificial que se decidió desarrollar fue una inteligencia conectable con objetos programables utilizando una máquina de estados. La ventaja de esta IA fue que aplicaba tres de los pilares de la ingeniería de juegos, la primera por tener un diseño modular, al no crear sistemas que se volvieran dependientes directamente de otros; la segunda por facilitar la edición de cada una de los elementos que conformaban la IA gracias a su diseño por componentes; y la tercera por facilitar la depuración, ya que permitía probar y editar el juego de una manera más fácil gracias al monitoreo de cada una de las piezas de la IA en la consola.

Los scripts que conforman la estructura de la IA son los siguientes:

Los códigos de cada sección están en el Anexo B. Códigos C#.

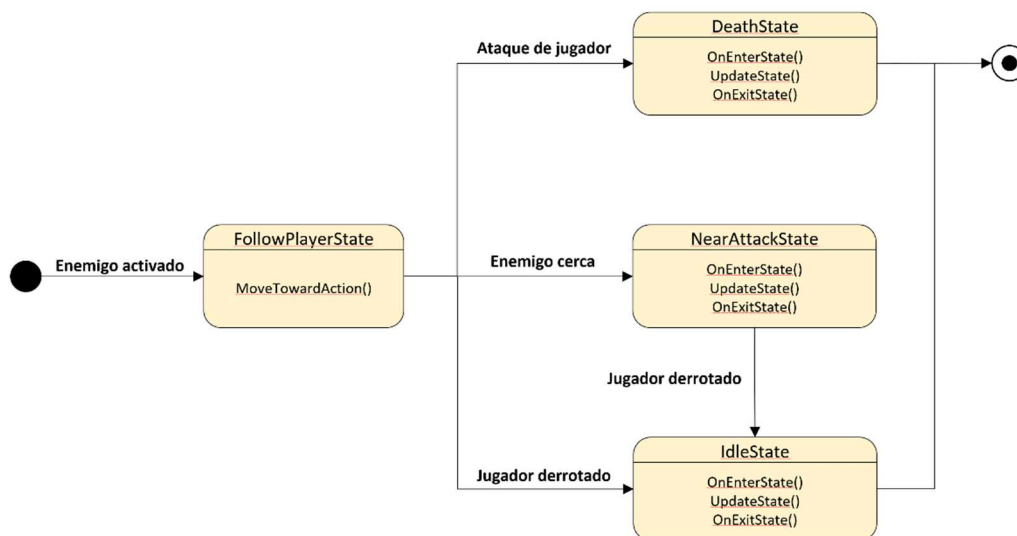
- StateController
  - Este script funciona como el cerebro de una máquina de estados, ya que este se encarga de controlar el cambio de estados, acciones, decisiones y transiciones de la IA, además de contener variables y métodos que pueden compartir diferentes estados.
- Action

- Es un scriptable object de clase abstract y que hereda de ScriptableObject permitiendo crear scripts que hereden de este, además de permitir crear assets de estos scripts. Action tiene un método llamado Act, y sobre este los scripts que hereden de él, deberán contener el comportamiento que quieren que tenga el agente.
- Decision
  - Este scriptableObject únicamente contiene un método abstract de tipo bool, sobre el cual los scripts que hereden de él deberán crear las condiciones que permitan cambiar de un estado a otro.
- Transition
  - Es una clase que contiene una variable decision, y dos variables state, y la función de la clase es brindar la estructura para el script decisión para que, dependiendo del valor devuelto de su método bool, pueda transicionar a uno de los dos estados.
- State
  - Se encarga de almacenar y ejecutar el comportamiento de actions, y de revisar constantemente las transitions, para poder cambiar de un estado a otro.

Antes de programar la máquina de estados se realizó un diagrama de estados conforme al funcionamiento de la misma, y una vez teniendo la idea general del funcionamiento de la máquina de estados, se procedió a programarla.



Figura 21 Diagrama de máquina de estados finita del enemigo



Los estados y su comportamiento son los siguientes:

- FollowPlayerState
  - Es el estado inicial al ser activado el enemigo. Este estado tiene dos actions, la primera se encarga de mover al enemigo hacia la posición del jugador, y el segundo únicamente activa la animación correspondiente a la acción. Y cuenta con 3 desicions, las cuales verifican constantemente si el jugador esta derrotado, si el jugador está cerca para atacar o si el jugador está atacando.
- DeathState
  - Este estado únicamente tiene una action, y se encarga de deshabilitar el collider del enemigo y de activar la animación correspondiente.
- NearAttackState
  - Cuenta con una action que habilita el collider de combate del enemigo, además de también ejecutar la animación de ataque. Cuenta con dos desicion, y esta verifica si el jugador fue derrotado o si el jugador lo atacó.
- IdleState
  - Este estado únicamente se encarga de activar la animación correspondiente.

**Sistema de oleadas.** El funcionamiento del sistema de oleadas consiste en crear una cierta cantidad de enemigos en relación al tiempo transcurrido del juego en una de las dos posiciones ubicadas fuera de los extremos de la zona de juego.

Para lograrlo se creó un scriptable object llamado TypeWave, que contenía toda la información necesaria de cada tipo de oleada, siendo estas de uno, dos tres, cuatro o hasta cinco enemigos, además de almacenar variables que definían los tiempos en que estas oleadas debían comportarse al momento de ser creadas.

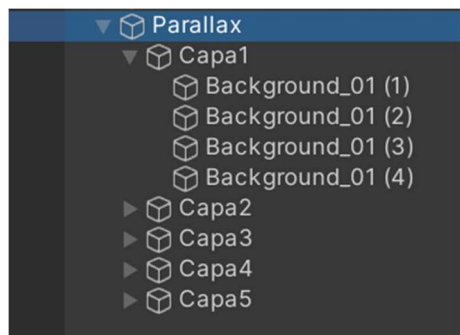
El script encargado del funcionamiento general del sistema de oleadas fue el EnemySpawnController, el cual, mediante los datos de cada tipo de oleada, y una animation curve encargada de definir una cantidad máxima de enemigos en relación al tiempo transcurrido en la partida, fue posible crear un sistema que adaptara el número de enemigos por oleada en relación al tiempo transcurrido en partida para así aumentar la dificultad del nivel.

### **Fondo y Efecto Paralaje (Parallax)**

Para realizar el efecto parallax se creó un game object con la siguiente estructura:

- Parallax
  - Es un objeto vacío que contiene otros gameobjects.
- Capa1
  - Es un gameobject que cuenta con un componente Sprite Renderer, y con el script ParallaxEffect. A su vez, este objeto contiene otros cuatro objetos, y cada uno de ellos tienen un SpriteRenderer.
- Capa 2, 3, 4, 5
  - Tienen los mismos elementos a la capa 1.

Figura 22 Estructura del objeto Parallax



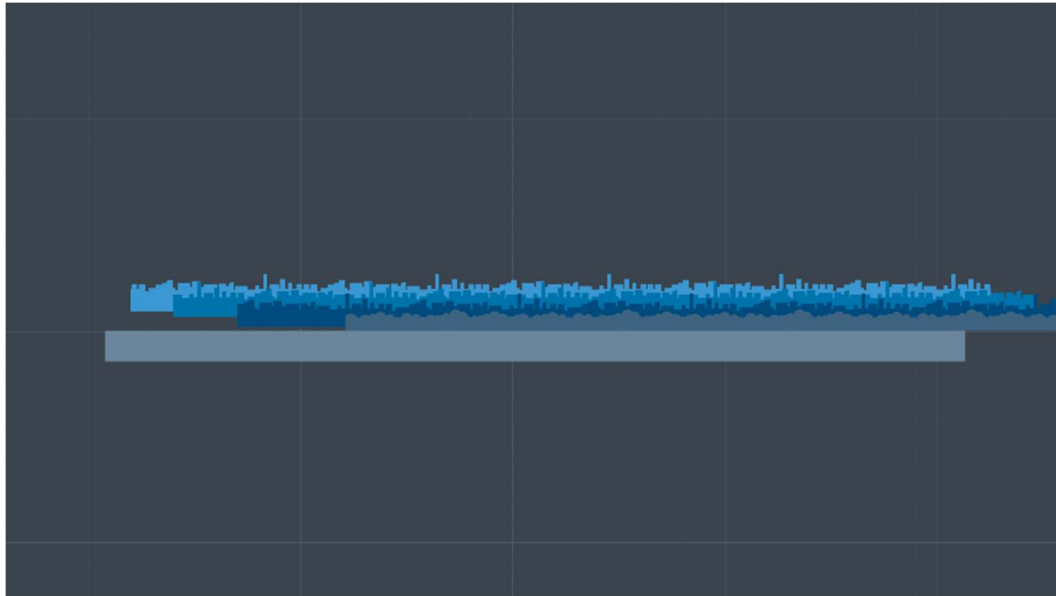
Para que el efecto funcione, es necesario que el fondo este dividido en capas o niveles acorde a su proximidad con la cámara. Por ejemplo, el suelo en una capa, arbustos en otra, montañas, y así con todos los elementos que conformen el fondo del nivel.

La capa 1 tiene en su Sprite Renderer un sprite que conforma el fondo del nivel. Los objetos hijos de este también tienen el mismo sprite, pero con la diferencia que cada uno de estos están ubicado de manera que se cree un panel horizontal de cinco objetos con el mismo sprite, teniendo dos objetos hijos del lado izquierdo, y los otros dos del lado derecho; generando así parte del fondo del nivel. Este proceso se repite con las demás capas, cada una mostrando partes separadas del nivel.

El script ParallaxEffect se encarga de mover el objeto en el que esta asignado el script acorde a la posición de la cámara; y cuando se supera el tamaño del sprite asignado al Sprite Renderer de este mismo objeto, se reubica el objeto a una velocidad imperceptible en donde aún sea posible ver el sprite. El valor que define a qué velocidad se mueven los objetos se llama ParallaxSpeed, el valor debe oscilar entre 0.9 para aquellos objetos que no se tienen que mover, hasta 0.1 para los que necesitan moverse a una mayor distancia.

Para probar el funcionamiento del efecto de paralaje se crearon cinco sprites que no reflejan el diseño de nivel planteado, sino que únicamente cumplen con una función de demostración, y se asignaron a cada una de las capas y sus objetos hijos. Logrando mover cada una de las capas a una velocidad diferente.

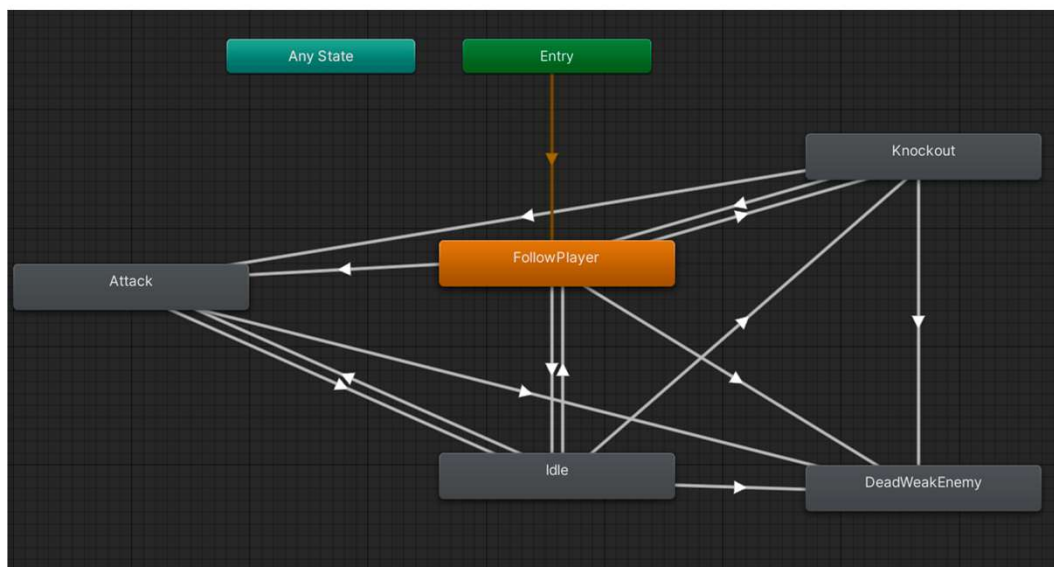
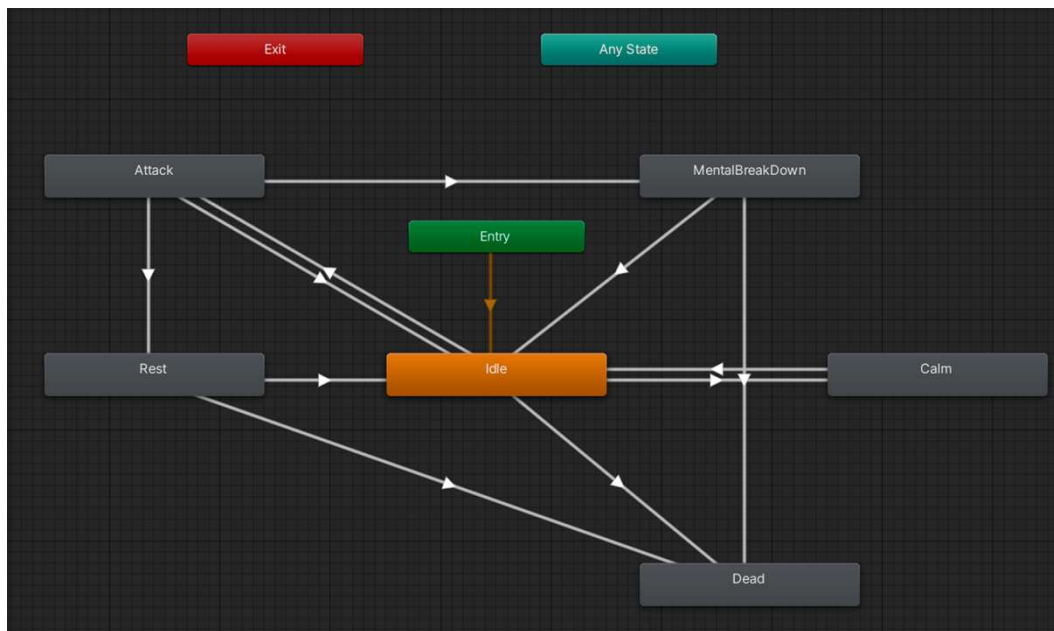
*Figura 23 Implementación del efecto paralaje (Parallax)*



### **Animaciones**

Las animaciones 2D del jugador como del enemigo se desarrollaron posteriormente de haber realizado la programación, sin embargo, dentro de los propios estados de sus respectivas máquinas de estados se crearon las líneas de código necesarias para activar sus correspondientes clips de animación en sus respectivos Animator Controller,

Figura 24 Animator de jugador y enemigo (de arriba a abajo)



### Interfaz de Usuario

La interfaz de usuario de la partida consta dos grupos de elementos, el primer grupo consto únicamente de un marcador ubicado en la zona superior central de la pantalla en la cual se reflejaba la cantidad total de enemigos derrotados por el jugador; y el segundo grupo consto de la pantalla de pausa y derrota.

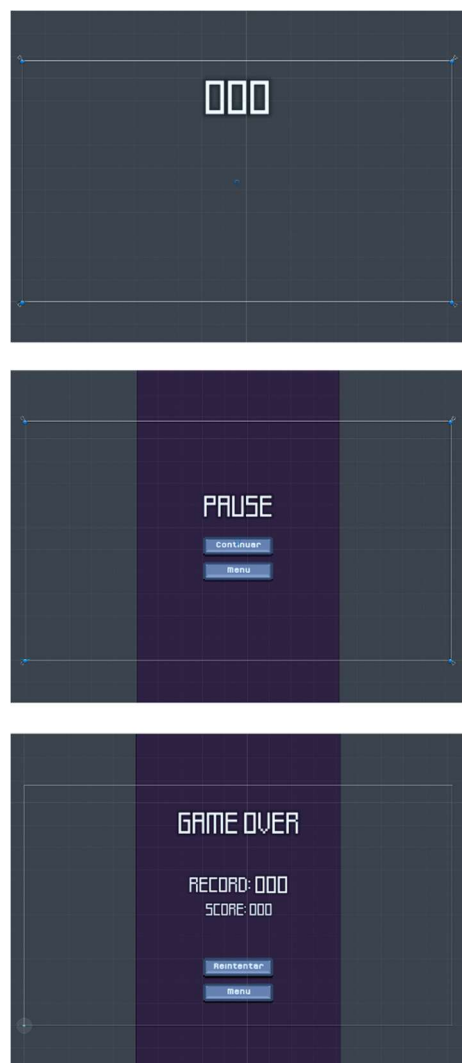
## Grupo 1

- **Marcador:** Es un elemento dentro del Canvas de tipo TextMeshPro, y su funcionamiento está ligado a un controlador llamado UIController, el cual mantiene una referencia al TextMeshPro y lo actualiza de acuerdo al puntaje guardado en el GameManagerSO.

## Grupo 2

- **Pantalla de pausa:** Esta conformado por un panel, un TextMeshPro y un botón que permite regresar a la partida. Para acceder a la pantalla de pausa, se creó una nueva Input Action ligada al botón "Escape", y al ser presionada la tecla se despliega la pantalla de pausa, y se cambia la escala de tiempo a 0 para simular la pausa dentro del juego.
- **Pantalla de derrota:** Cuenta con un panel, tres TextMeshPro, dos de ellos para mostrar el puntaje logrado en partida, y otro para mostrar el puntaje más alto registrado de todas las partidas jugadas; y dos botones, uno para volver a intentar otra partida, y otro para regresar al menú principal.

Figura 25 Interfaces de Usuario del nivel



### Condición de Derrota y Clasificación

La partida de juego termina cuando el collider del objeto CombatCollider del enemigo entra en contacto con el collider principal del jugador, cambia el estado de juego a GameOver y se desencadenan una serie de acciones para mostrar que la partida ha terminado.

Para poder comprar y guardar los datos de puntuación del jugador, se utilizó un sistema de guardado con serialización binaria, la cual consiste en convertir los datos a guardar en valores binarios y guardarlos en un archivo específico, y viceversa, para cargar los datos, se convierten de binario a los datos originales. Con este sistema de guardado se pudo crear el sistema de clasificación, donde se

registra la puntuación más alta alcanzada hasta ahora, y se actualiza y guarda una nueva puntuación si se supera.

### **7.2.2. Desarrollo del Apartado Visual y Animación**

El trabajo de esta sección se dividió en tres áreas, la primera a realizar fue el apartado visual del videojuego, la segunda la animación y la tercera fue efectos visuales. Para el apartado visual se realizaron los siguientes puntos:

- Diseño de personaje principal
- Diseño de enemigo
- Fondo (Background)
- Imagotipo
- Barra de estado mental
- Botones, paneles y tipografía

Para el apartado de la animación se realizaron:

- Animaciones del personaje principal y del enemigo
- Animación de la barra de estado mental

Y finalmente para los efectos visuales se desarrolló:

- Postprocesado y efectos visuales en Unity

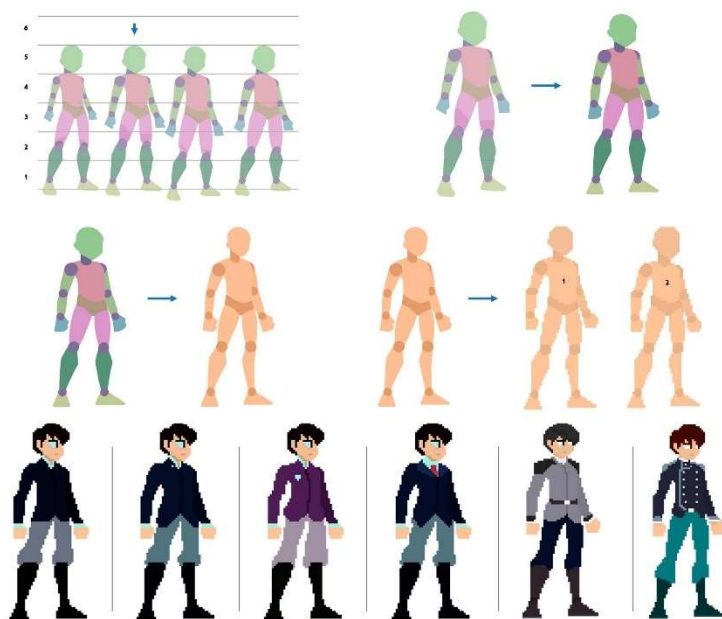
Se decido utilizar el estilo pixel art para el diseño de los personajes y el fondo del videojuego debido a que es un estilo que no requiere de un gran tiempo de producción y que mantiene una estética “retro” que recuerda a los videojuegos de los años 90s. Además, este estilo tiene una característica interesante a la vista, y es la abstracción, ya que, al ser un estilo limitado a un espacio de trabajo de unos cuantos pixeles, requiere que el diseño sea sencillo y conciso, pero manteniendo ciertos elementos en su composición que lo vuelven interesante y de fácil lectura para el espectador.

**Diseño del Personaje Principal.** Al tener un desarrollo escrito del apartado visual y psicológico del personaje principal se comenzó a diseñar al personaje. El proceso del diseño paso por



varias iteraciones para poder encontrar el diseño final, primero se realizaron varios bocetos de forma y silueta, esto para poder tener una idea general de la estructura y proporciones del personaje, de los cuales se eligió el segundo por tener un diseño más dinámico en relación a la longitud de las piernas y los brazos; posteriormente se pulió el boceto dos veces hasta tener dos bases más estilizadas, y se adaptaron al diseño pixel art en un canvas de 103 X 103 pixeles, y sobre esas base creada, se empezaron a trabajar nuevos bocetos con vestimenta y color.

*Figura 26 Bocetos del personaje principal*



Sin embargo, el canvas elegido era demasiado grande teniendo en cuenta el tiempo de producción de la animación y los detalles que estos iban a requerir, por lo que se redujo a un canvas menor de 64 X 64 pixeles, por lo que se tuvo que rediseñar al personaje para poder mantener los elementos más representativos de la base estilizada, que eran las piernas, brazos y rostro. Finalmente, de los bocetos realizados, se decidió trabajar y mejorar el último, ya que su paleta de color y vestimenta reflejaban mejor las características del personaje.

*Figura 27 Propuesta final de diseño de personaje principal*

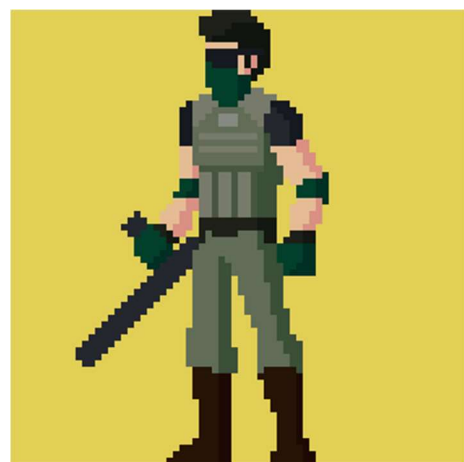


**Diseño de Enemigo.** Al igual que el personaje principal, se diseñó al enemigo teniendo en cuenta el desarrollo escrito del personaje. El proceso de diseño fue ligeramente diferente al del personaje principal, ya que, al tener ya una base sólida de este personaje, se trabajaron los bocetos iniciales sobre esas proporciones utilizando siluetas que fueran claras y de fácil lectura. Además, se decidió trabajar directamente sobre el estilo pixel art para ahorrar pasos en el desarrollo. Se realizaron 5 bocetos, y de estas, se decidió trabajar sobre las opciones uno y cuatro, ya que ambas iban más acordes en proporciones y estilo que el personaje principal. Finalmente, después de 3 iteraciones sobre estas siluetas se llegó al diseño final del personaje.

*Figura 28 Bocetos de enemigo*



*Figura 29 Propuesta final de diseño de enemigo*



**Fondo (Background).** Para desarrollar el fondo del nivel se tomaron en consideración dos aspectos, el primero fue el desarrollo escrito previo del diseño que debería de tener el fondo, y en segundo, su uso dentro del juego, ya que al ser un fondo con un efecto de paralaje (parallax) este debía ser dividido en varias capas para que estas fueran desplazadas a diferentes velocidades y generar un efecto de profundidad en el juego. Se realizaron 7 bocetos para definir cuál sería el que mejor aprovecharía el efecto de paralaje, además de no requerir un desarrollo tan complejo. De todas las opciones se decidió trabajar sobre la número seis, ya que brinda más espacio visual en la zona de juego (suelo) y no satura otras zonas de la pantalla, además de contar con elementos que pueden

aprovechar bien el efecto de paralaje como árboles, montañas y nubes. Posteriormente se creó el canvas con medidas de 569 X 320 pixeles y se activó una herramienta disponible en Aseprite llamada Tiled in X axis, la cual replica el canvas en los laterales izquierdo y derecho del mismo, simulando un mosaico, permitiendo diseñar el fondo teniendo en consideración que este pueda repetirse de forma indefinida hacia la izquierda o derecha. Tanto la composición, las proporciones de los elementos y la paleta de color del fondo se diseñaron teniendo en cuenta las proporciones del personaje principal como del enemigo. Para funcionar con el sistema de paralaje, se dividió el fondo en varias secciones según su proximidad hipotética con la cámara, y estas se trabajaron en capas separadas, con doce capas que formaron el fondo. Cada una de estas capas fue exportada individualmente en formato PNG para su implementación en unity.

Figura 30 Bocetos del diseño de fondo

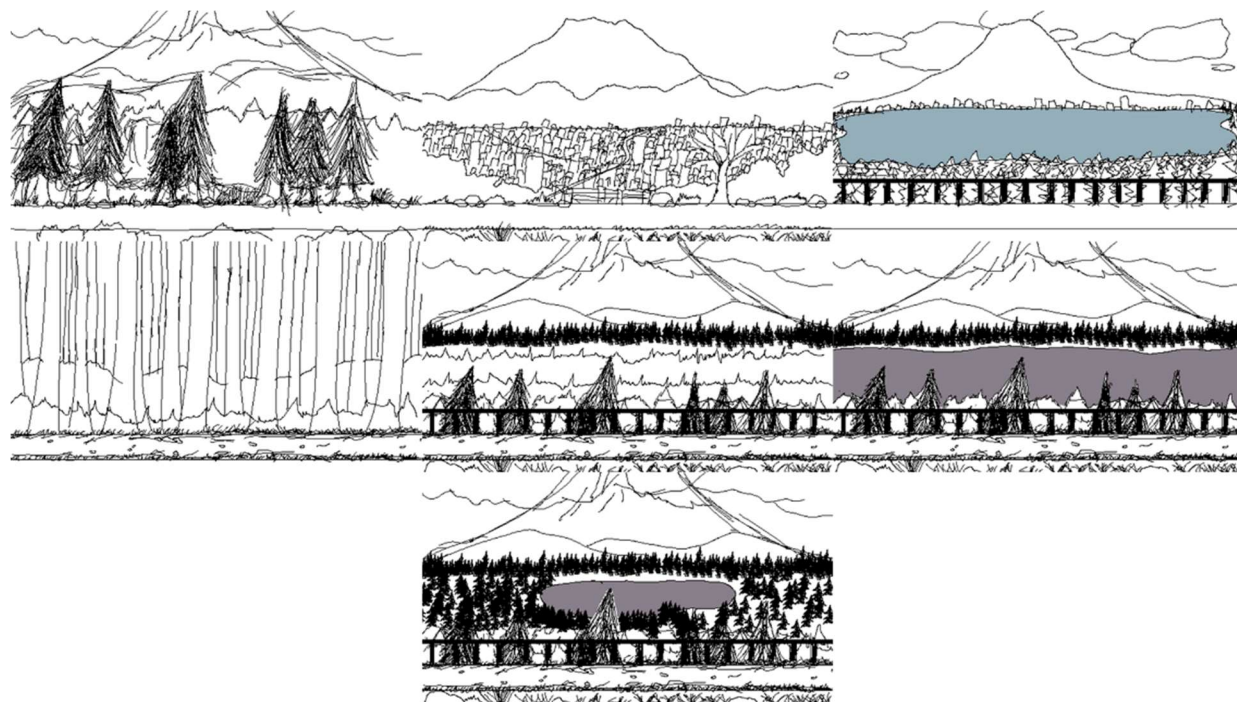


Figura 31 Pruebas de composición y paleta de color

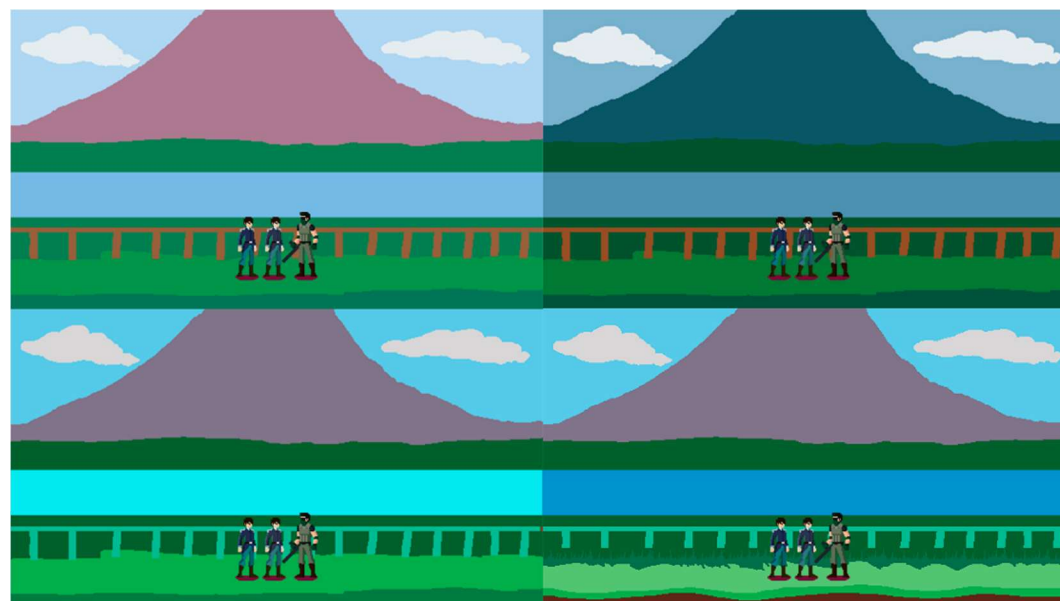


Figura 32 Propuesta final de fondo

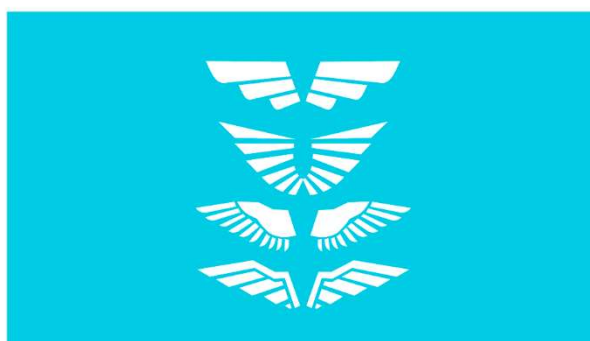
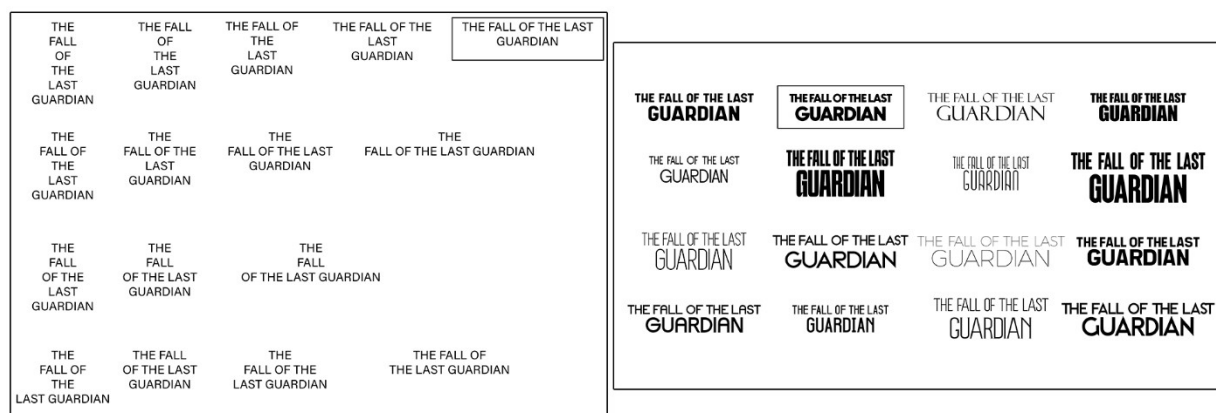


**Imagotipo.** Un imagotipo es la unión entre un logotipo y un isotipo, lo que significa que es la fusión entre la parte textual y la parte gráfica que representan o identifican una marca. Su característica más distintiva es que la parte textual y la gráfica están bien delimitadas una de la otra y no terminan mezclándose, sin embargo, ambas funcionan en conjunto dando un equilibrio a la imagen.

En el videojuego, el imagotipo se desarrolló teniendo en cuenta que era necesaria una imagen que sirviera de identidad visual para el proyecto, que fuera llamativa y de fácil lectura para el espectador. Con estas características en cuenta, se bocetaron las primeras ideas en cuanto a la disposición de las palabras dentro de esta imagen. De estos bosquejos, se eligió el número cinco, ya que no ocupaba demasiado espacio tanto vertical como horizontalmente, además de que la lectura se centraba dos puntos específicos de la oración, teniendo en el primer renglón el predicado “The fall of the last”, creando una incógnita o intriga en el espectador sobre quién o qué se habla; y en el segundo renglón “Guardian”, el sujeto al que se está refiriendo la oración. Después se hicieron pruebas tipográficas de diferentes tamaños y estilos, y de todas las propuestas se eligió la número dos por tener un diseño acorde a las tendencias en identidades de marca en videojuegos, además de un equilibrio visual entre los dos renglones. Al tener la parte tipográfica lista, se procedió a trabajar en la parte gráfica que acompañara a esta. Se bocetaron cuatro diseños de alas extendidas, las cuales hacen una alegoría a la esperanza. De los diseños planteados, se tomó el primero debido a que es un

diseño sencillo con pocos elementos, y que comunica el mensaje deseado, además de combinar de una forma más limpia con el apartado tipográfico.

*Figura 33 Proceso de diseño de imagotipo*

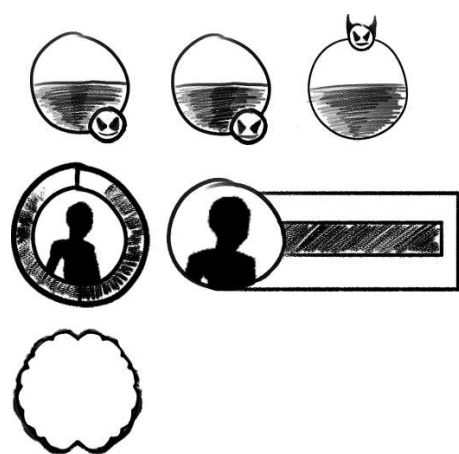


*Figura 34 Propuesta final de diseño final de imagotipo*

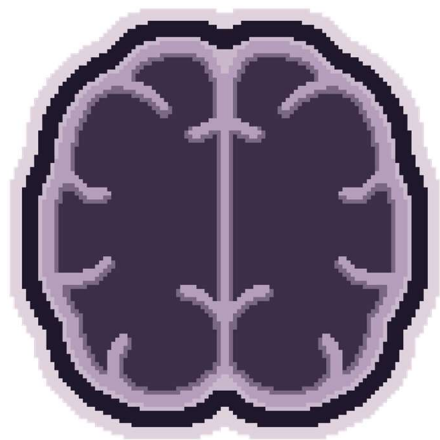


**Diseño de Barra de Estado Mental.** Encontrar el diseño final para la barra de estado mental se realizaron bocetos que servirían de indicador para el jugador sobre el estado mental del personaje. De estas opciones se eligió la última opción, la silueta del cerebro, ya que es una interpretación directa común relacionada con estado de la salud mental de las personas. Al tener el boceto inicial se trabajó en un canvas de 112 X 112 pixeles la estructura del cerebro y la paleta de colores, tomando en consideración la paleta de color del fondo para que ningún elemento resaltara de manera disonante uno sobre el otro.

*Figura 35 Bocetos del diseño de la barra de estado mental*



*Figura 36 Propuesta final de la barra de estado mental*



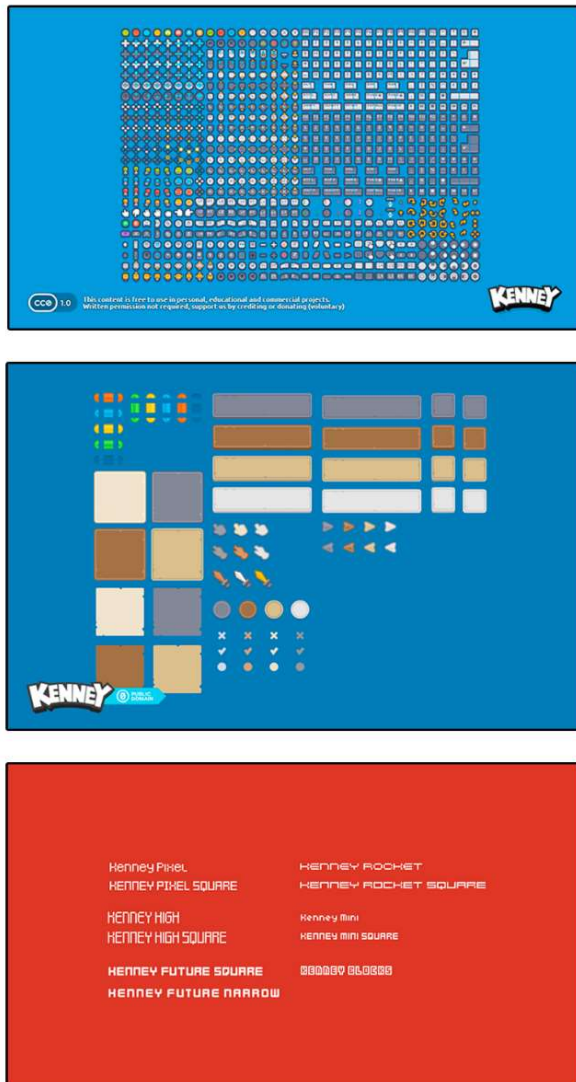


**Botones, Paneles y Tipografía.** Gracias al trabajo en el apartado artístico de la página del desarrollador Kenney.nl, sitio donde se pueden descargar y utilizar assets de dominio público, se decidió implementar en el videojuego tres paquetes de assets, los cuales fueron los siguientes:

- UI Pack: RPG Expansion
- Input Prompts Pixel 16×
- Kenney Fonts

Utilizar estos assets no solo brindaron una mejora en la estética del videojuego, sino que redujeron los tiempos de producción en esta área, y fue posible dedicar el tiempo destinado a estas actividades a otras que lo requiriesen.

Figura 37 Assets utilizados para el proyecto, de Kenney.nl



**Animaciones del Personaje Principal y del Enemigo.** Las animaciones a trabajar para el personaje principal se basaron en la cantidad de estados que tenía el animator controller de este mismo personaje, teniendo que desarrollar un total de seis animaciones, las cuales fueron:

- Idle
- Attack
- Rest
- Calm
- MentalBreakdown

- Dead

Y de la misma forma para las animaciones del enemigo, se utilizaron los estados de su máquina de estados como referencia para contemplar la cantidad de animaciones a desarrollar, siendo un total de cuatro animaciones, las cuales fueron:

- Idle
- Follow
- Attack
- Dead

El canvas de trabajo dentro de Aseprite varió para cada una de las animaciones, pero siempre se mantuvo una altura de 96 píxeles. Previo a animar, se eligieron algunos de los 12 principios básicos de la animación escritos por Ollie Johnston y Frank Thomas en su libro “Disney Animation: The Illusion of Life” para utilizar en algunas animaciones, los principios elegidos fueron anticipation, staging y pose to pose. Lo primero para realizar las animaciones fue bocetar los fotogramas clave, y posteriormente crear los fotogramas entre estos para crear pequeñas transiciones entre acciones. Después se pulieron los bocetos y se comenzó el coloreado de los personajes. Finalmente, al tener listas todas las animaciones, estas debían ser exportadas para poder ser usadas dentro de Unity, y gracias a una herramienta de Aseprite fue posible exportar cada una de las animaciones en un atlas de textura (sprite sheet), la cual, guarda en una imagen cada uno de los fotogramas de la animación.

Figura 38 Sprite sheet del personaje principal

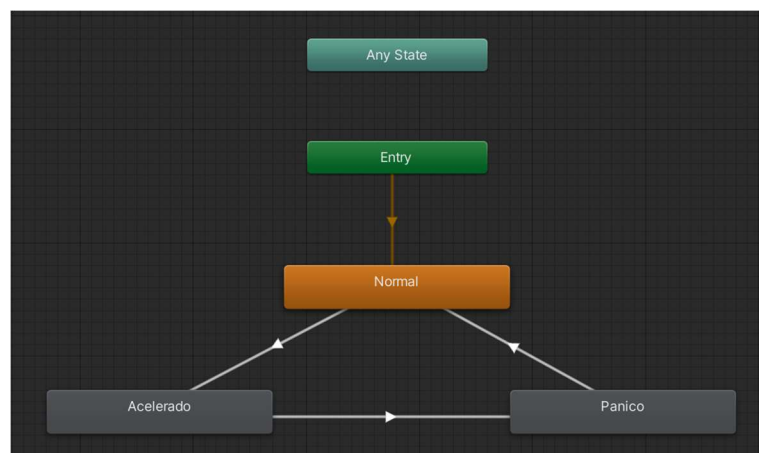


Figura 39 Sprite sheet del enemigo



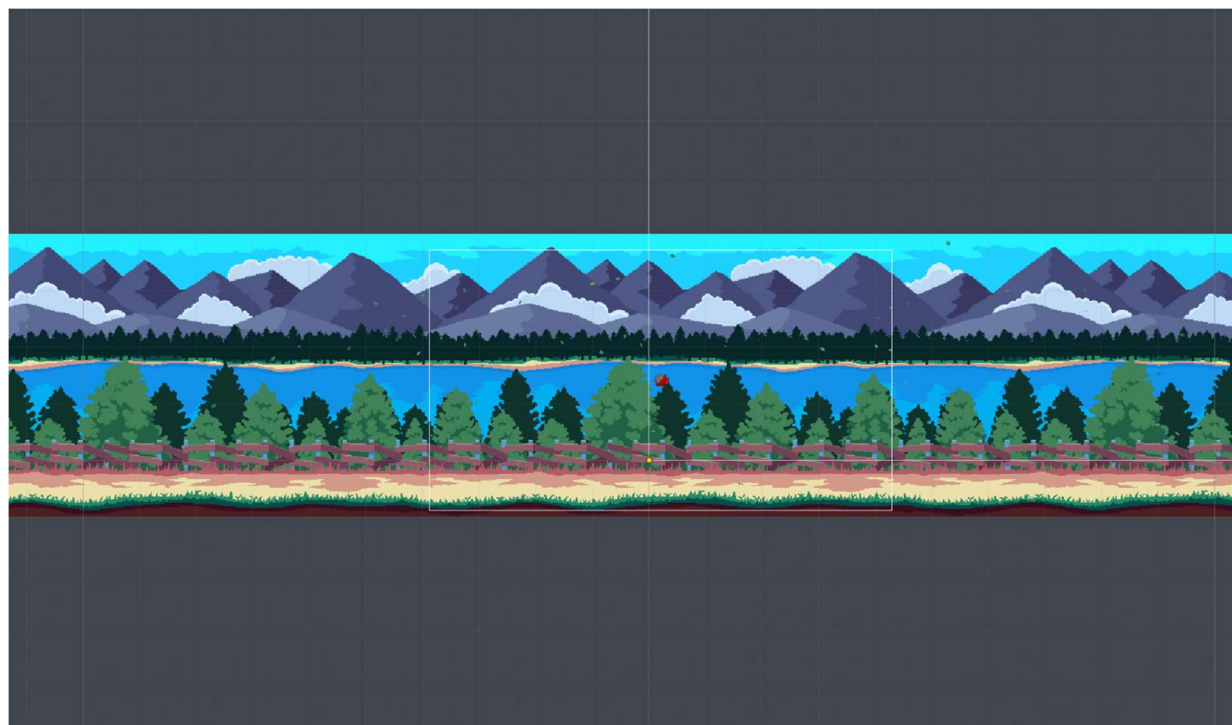
**Animación de la Barra de Estado Mental.** Al ser la barra de estado mental una representación sobre el estado de salud mental del personaje principal, y esta al ser parte de las mecánicas de juego como un elemento que hay que controlar para poder continuar jugando mediante un slider, se pensaron tres estados en los que la barra de estado mental reflejara los cambios constantes del valor del slider durante la partida; en donde la barra palpitara como un corazón, y a mayor o menor valor, la barra palpitaría con mayor o menor fuerza. Para la animación de la barra mental se utilizó directamente la herramienta de Animation de unity, ya que el objetivo únicamente era modificar los valores de la escala del sprite. Por ello, al objeto que contiene el slider de la barra de estado mental, se le asignó y creó un Animator controller, y sobre este se crearon tres clips de animación, normal, acelerado y pánico; cada uno de ellos modificaba la escala del slider para hacerlo más grande y regresarlo a su valor original, pero la diferencia entre cada estado era la velocidad con la que hacían ese cambio, a mayor duración del clip de animación, más tranquilo era la simulación de palpitación, y a menor duración, mayor era esta.

Figura 40 Clips de animación de la barra de estado mental



**Implementación en Unity.** En el caso del fondo se importaron los doce sprites que lo conformaban, a unity con una configuración de importación que permitiera trabajar con imágenes de tan baja resolución, pero sin perder calidad. Dentro del objeto Parallax se agregaron y renombraron doce nuevos objetos, acorde al sprite que representaban, manteniendo la estructura y la configuración del sprite renderer del objeto base de parallax. Dentro de la configuración que cada objeto tuvo en su respectivo script de parallax, la variable que se modificó fue la de Parallax speed, partiendo desde cero para el objeto más cercano que era el suelo, hasta el 0.9 que era el cielo del fondo.

Figura 41 Implementación del fondo en unity



Finalmente, para las animaciones del jugador y del enemigo, se procedió a importar todos los elementos visuales a unity manteniendo la configuración de importación usada en el fondo. Las animaciones se asignaron a los clips correspondientes dentro del panel Animation de cada uno de los objetos Player y Enemy. Para su correcto funcionamiento, se configuraron cada transición ligada a cada clip para que no hubiera problemas de reproducción al cambiar de clip.

Figura 42 Implementación de las animaciones en Unity



**Postprocesado y efectos visuales.** Gracias a las herramientas disponibles con el asset de Postprocessing de Unity, fue posible crear efectos visuales en la cámara para reforzar la mecánica de juego de controlar el aumento de la barra de estado mental. Para su uso dentro del nivel primero se asignó a la Main Camera el componente de Post-process Layer, y seleccionar la propia cámara, esto para poder aplicar el postprocesado en la cámara; y lo segundo fue agregar el componente Post – Process volume, crear un perfil de postprocesado, y agregar dos efectos, los cuales fueron los siguientes.:

- Vignette
  - Este efecto permitió crear una viñeta de color negro que al modificar su valor Intensity, se podía limitar la visibilidad en la pantalla, permitiendo ver únicamente el centro de la pantalla, o en su defecto, al personaje, debido a que la cámara sigue en todo momento al personaje. Este efecto permitió agregar un grado de dificultad y complejidad al juego, ya que, mediante código, se ligó el cambio de valor en el slider de la barra de estado mental con el valor de Intensity del efecto, logrando que si no se controlaba la barra y se llenaba, la vignette iría creciendo, limitando la anticipación del jugador a la aparición de los enemigos, complicando jugar en ese estado, a menos que se redujera el



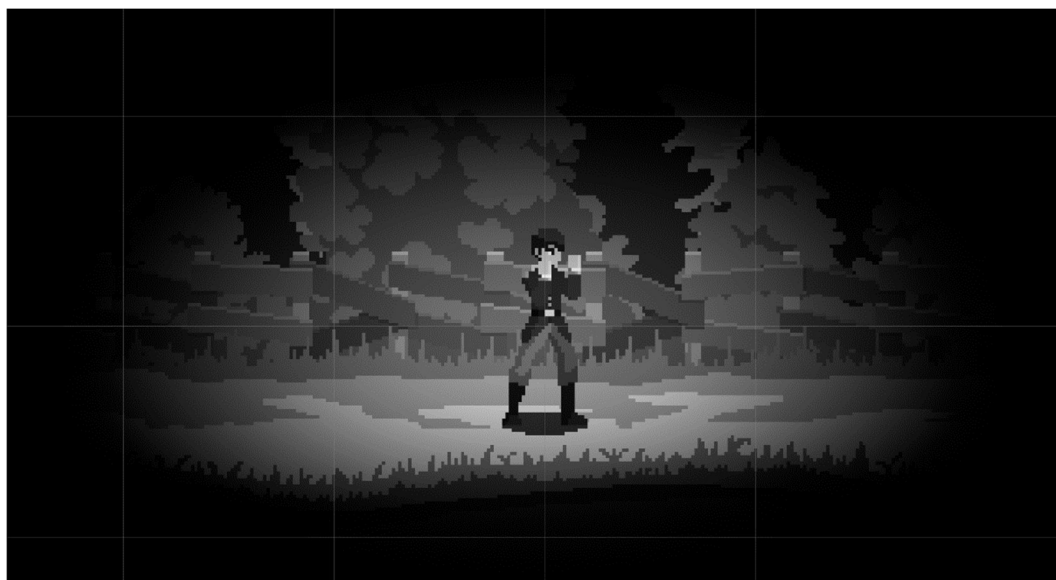
estrés de la barra y la vignette reduciría su tamaño volviendo a su estado original en el que no es visible en la pantalla.

- Color Grading

- Este efecto permitió en primera instancia ajustar ligeramente los colores para tener una escena un poco más fría, y no tan cálida como estaba diseñada. Al igual que el efecto anterior, el color grading también se utilizó para reforzar visualmente la mecánica de controlar la barra de estado mental, ligando nuevamente el cambio de valor en el slider de la barra de estado mental con el valor de la saturación, logrando que, al tener un nivel alto de estrés en la barra, la escena comenzaría a tornarse a escala de grises, pero a un valor menor de estrés, la paleta de color volvería a su estado original.

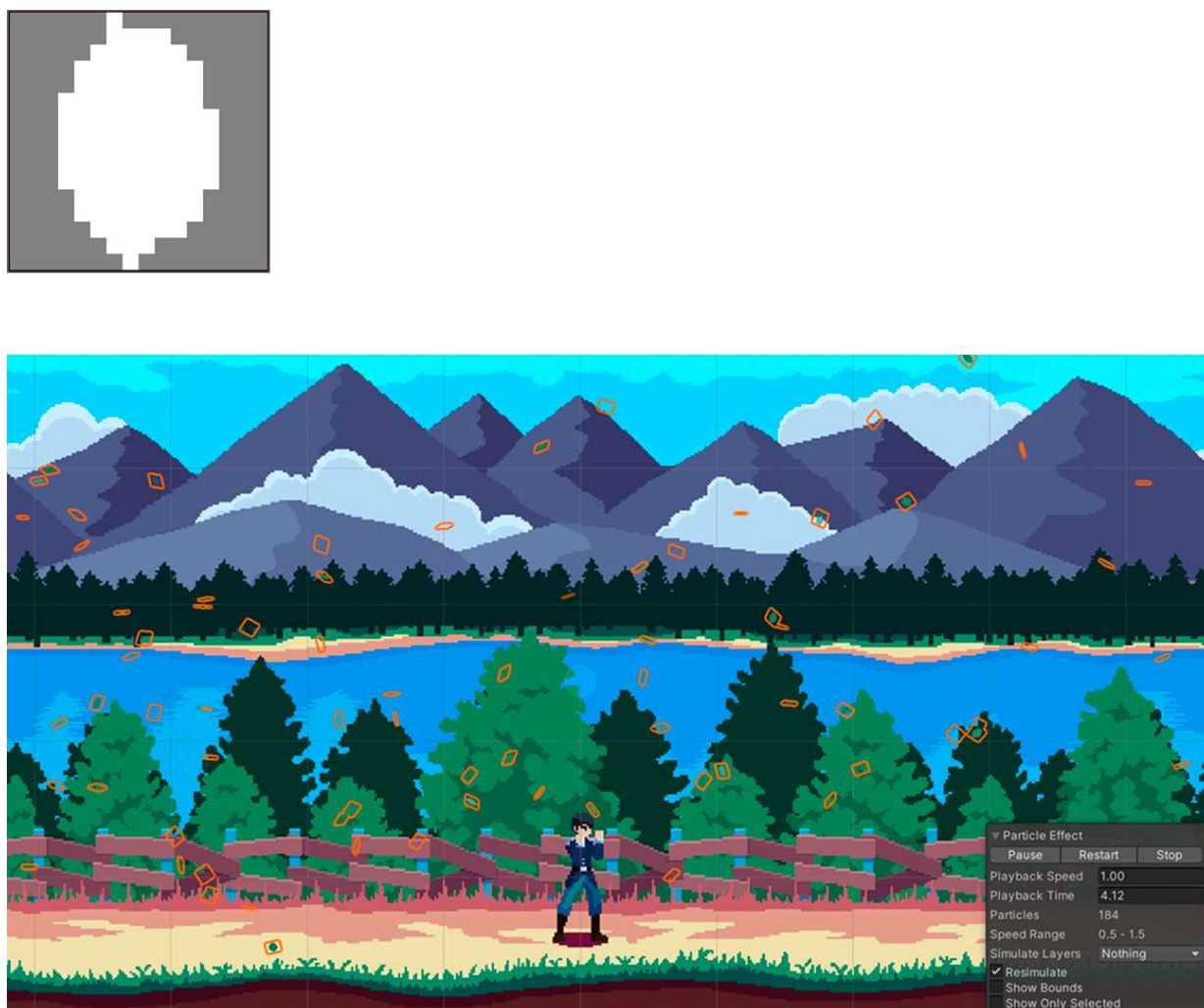
Además de utilizar los componentes de postprocesado, también se utilizó el componente CinemachineVirtualCamera, ya que dentro de sus propiedades se encuentra una llamada Noise, la cual genera un efecto de tambaleo en la cámara, y al igual que con los efectos, se ligaron los valores del Noise al cambio de valor del slider de la barra mental, provocando que a mayor estrés la cámara se tambaleara ligeramente, y si el valor es menor hacer imperceptible el tambaleo.

*Figura 43 Implementación de postprocesado y efectos visuales*



Finalmente se decidió agregar un sistema de partículas que simulara el viento llevándose las hojas de los árboles. Se decidió agregar este elemento meramente para mejora la estética y brindar una experiencia un poco más inmersiva. Para la textura de las partículas se creó un sprite con forma de hoja de 16 X 16 pixeles y se creó un Material dentro de unity para darle color a esa textura.

*Figura 44 Implementación del sistema de partículas*



### 7.2.3. Música y Sonido

Gracias al trabajo libre de derechos de autor de desarrolladores y autores musicales, se decidió ocupar este tipo materiales dentro del videojuego, acreditando dicho material prestado dentro del mismo juego. Para los sonidos se usaron los siguientes paquetes:

- Interface Sounds

- Impact Sounds

Nuevamente, estos assets fueron tomados de la página del desarrollador Kenney.nl, en donde la mayoría de assets cuentan con una licencia de Creative Commons de tipo CC0 1.0 Universal, lo cual vuelve el material de uso de dominio público.

Para las piezas musicales (soundtrack) que acompañarían tanto el menú principal como el nivel de juego, fueron escogidos de dos artistas diferentes; la primera pieza se titula “A bit of hope”, del álbum Arcade Kid lanzado en 2020, y compuesta por David Fesliyan. Esta pieza se encuentra disponible en la página fesliyanstudios.com, una biblioteca de música para su uso gratuito o por licencia, dependiendo del uso que se le dé a la pieza musical. Para la música utilizada, puede usarse gratis si no se usa para un proyecto comercial o monetizado.

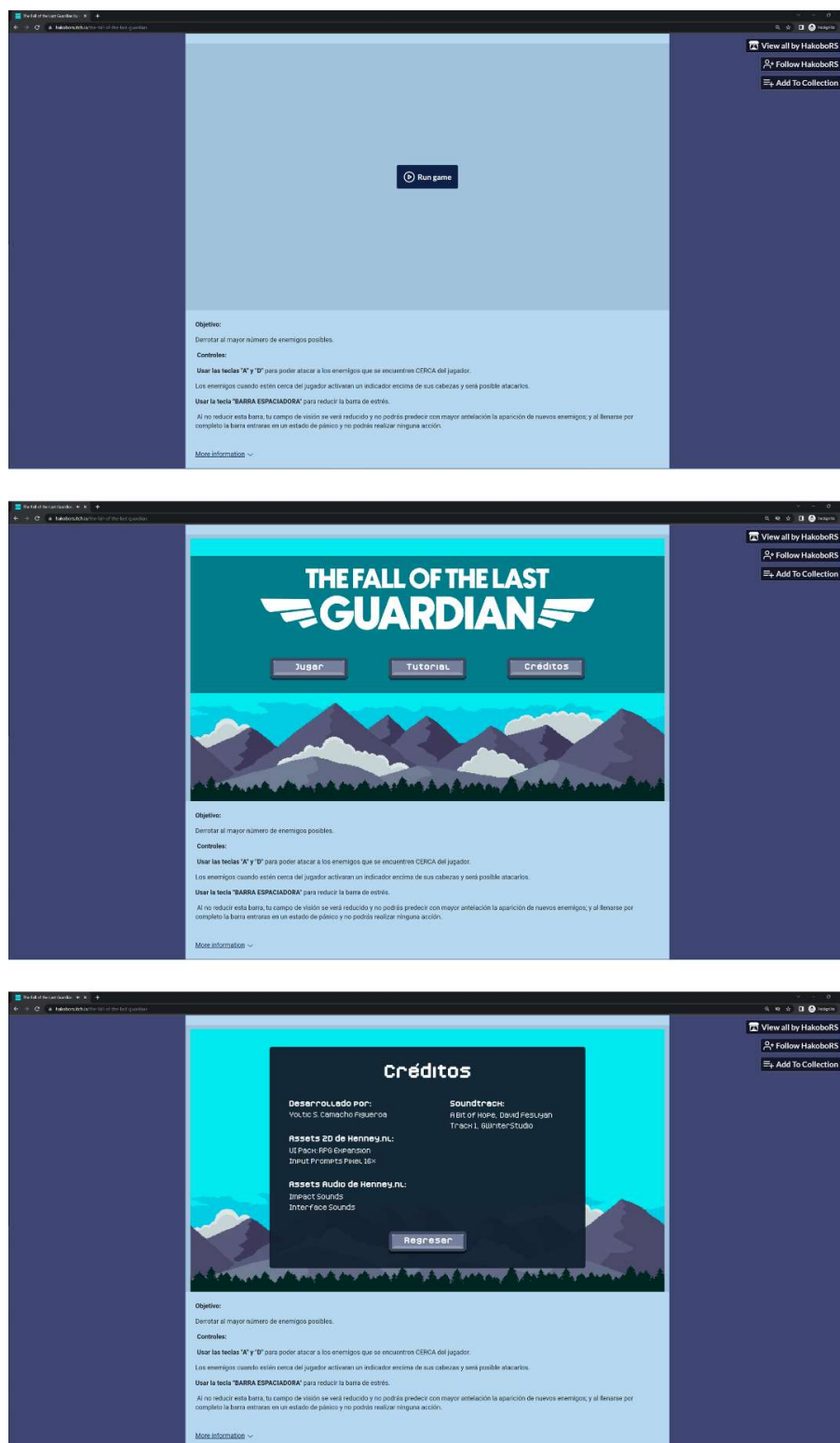
Y la segunda pieza fue obtenida de la asset store de unity, la cual se titula “Track 1”, del asset 8Bit Music – 062022, lanzado en 2022 y compuesta por GwriterStudio. En cuanto a la licencia de uso, el autor menciona en la descripción del asset que la música puede ser usada para fines comerciales y no comerciales, y no requiere una acreditación en la obra.

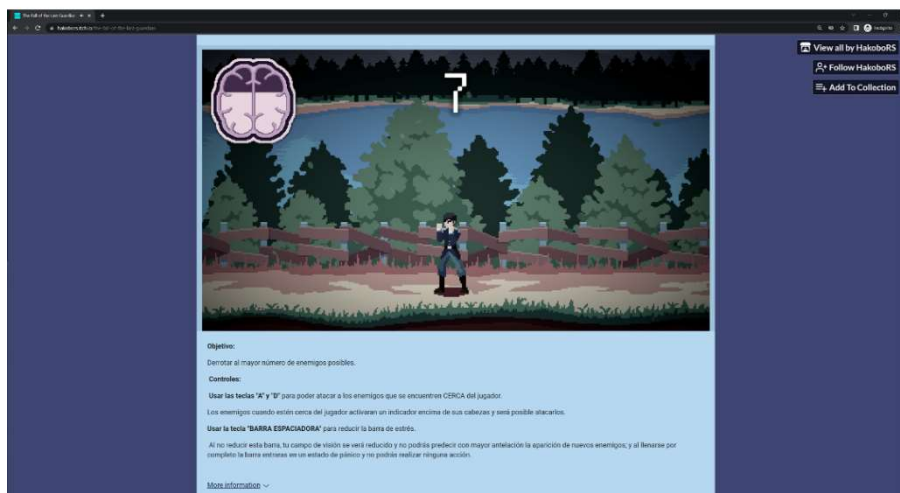
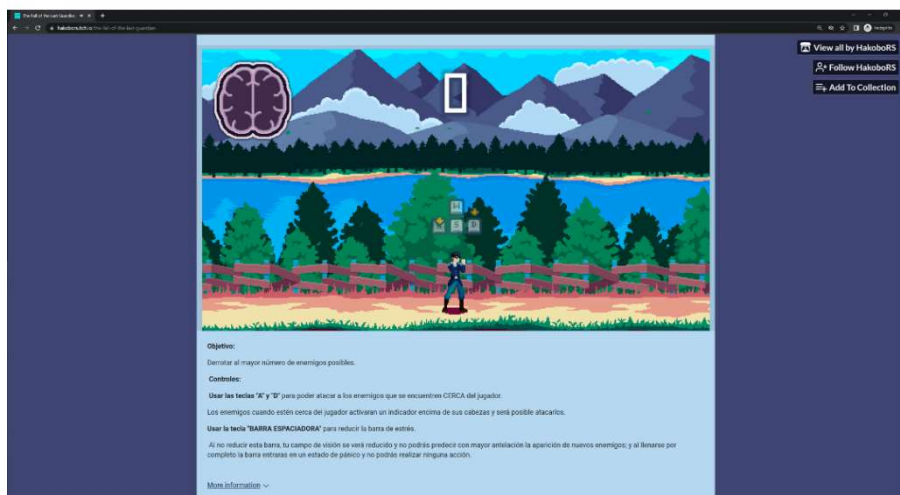
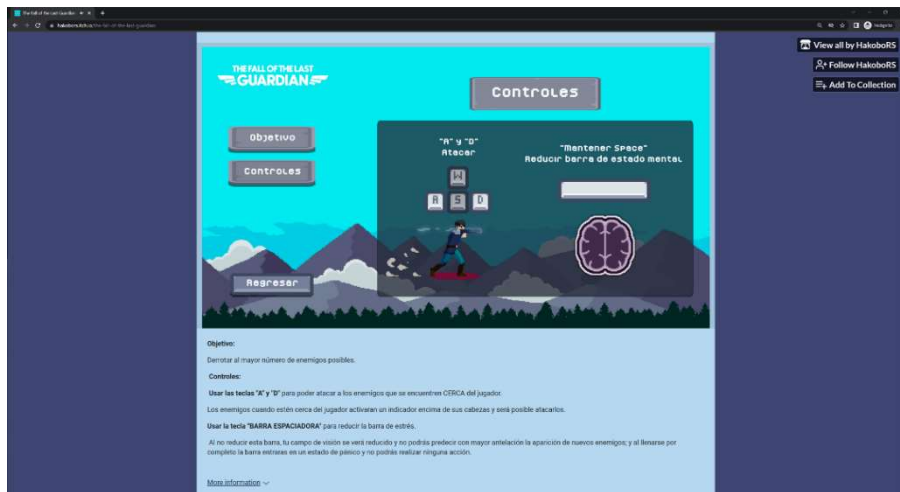
### **7.3. Publicación**

Al tener listo el prototipo del videojuego, se exportó el videojuego en formato WebGL, el cual permite publicar el proyecto como un programa de JavaScript utilizando HTML5 y el renderizado de WebGL.

El videojuego se publicó en la página de Itch.io por ser una plataforma de fácil uso, además por brindar una gran variedad de herramientas que permiten administrar de mejor manera el proyecto, gracias a las analíticas, formas de distribución dentro de la página, e interacciones y la creación de un devlog para poder estar en contacto con la comunidad interesada en el videojuego.

Figura 45 Capturas de pantalla del videojuego desde Itch.io





## **8. Resultados y conclusiones**

La industria de los videojuegos en los últimos años ha comenzado a romper paradigmas en torno a su percepción negativa dentro de la sociedad, esto gracias a investigaciones de profesionales que demuestran que estos pueden crear entornos de relajación, reflexión y espacios sociales, además de que sus usos van más allá del entretenimiento y estos tienen aplicaciones clínicas para el tratamiento de personas con ciertas enfermedades de salud mental; también por parte de los desarrolladores, que abordan nuevas temáticas en sus videojuegos en las cuales el jugador se ve involucrado tanto en la narrativa como en las mecánicas de juego, creando experiencias en las cuales se identifique con los personajes de la historia; y sobre todo por los jugadores, que se permiten jugar títulos diferentes más allá de los ya conocidos, mejor establecidos en el mercado, o de los que comúnmente están acostumbrados a jugar, permitiéndoles experimentar nuevas narrativas, estilos visuales y mecánicas de juego, y que dicha interacción motive al jugar a la reflexión sobre su comportamiento, pensamientos y su interacción con el medio que los rodea.

Sin embargo, los videojuegos que abordan la salud mental, en específico el trastorno de pánico, como tema principal para representar las dificultades y retos que viven personas que padecen de este tipo de problema de salud es reducido, y si bien existen videojuegos que abordan la ansiedad y estrés que son factores determinantes en el trastorno de pánico, ningún estudio de videojuegos tanto independiente como de renombre, ha desarrollado un videojuego centrado específicamente en este tipo de problema de salud mental.

Este proyecto tuvo como objetivo diseñar, desarrollar y publicar el prototipo de un videojuego de acción en la página Itch.io, que tomara como base el trastorno de pánico para definir las mecánicas de juego.

Para lograrlo, fue necesario documentarse en páginas de clínicas especializadas, en manuales de diagnóstico, y de testimonios de personas que hayan o estuvieran sufriendo de este tipo de trastorno. Esto permitió desarrollar no solo una investigación más realista, justificada y sensible acerca de este tipo de trastorno, sino que fueron las bases sobre las cuales las mecánicas están fundamentadas.

Como resultado, se logró crear el documento de diseño, el desarrollo del prototipo del videojuego y la publicación del mismo en la página de Itch.io. Además de presentar el proyecto ante

alumnos de nuevo ingreso de la licenciatura de Arte Digital de la generación Primavera 2023 de la Escuela de Artes Plásticas y Audiovisuales (ARPA) de la Benemérita Universidad Autónoma de Puebla (BUAP) , y a un grupo de alumnos de séptimo y octavo semestres pertenecientes a un programa de servicio social enfocado en el diseño y desarrollo de juegos y videojuegos perteneciente al laboratorio de desarrollo de videojuegos de la misma escuela llamado ARPA Game Lab, en ambas presentaciones se recibieron comentarios en su mayoría positivos, en los que destacaban la jugabilidad y la implementación de las mecánicas de juego en relación al tema central del videojuego, además de ofrecer cambios o mejoras a futuro que mejoraran la experiencia del jugador al momento de aumentar el estrés y de sufrir el ataque de pánico si esta no se controlaba, sin embargo, existieron elementos que no quedaron lo suficientemente claros al inicio de las primeras partidas, el primero fue el tutorial, ya que no era lo suficientemente claro tanto en la partida como en el apartado desde el menú principal, dificultando el que se debía hacer al inicio del juego, y el segundo punto fueron los indicadores de los enemigos, ya que no eran claros en lo que indicaban y su función dentro del juego; para ambos puntos se recibieron recomendaciones para mejorar a futuro el videojuego para que fuera más entendible la jugabilidad al inicio de la partida, ya que una vez acostumbrado a lo que sucedía en el juego, y tras varios intentos, era fácil entender que se debía hacer, y entretenido al tener que superar la puntuación más alta entre las personas que jugaban el juego.

Este proyecto ofrece una representación desde las mecánicas de juego del género de acción para representar el trastorno de pánico, logrando visibilizar desde esta perspectiva las dificultades que viven estas personas en su día a día, no obstante, cabe resaltar que este videojuego es un prototipo, lo que significa que si bien todas las funcionalidades básicas están implementadas y es posible comprender y entender el mensaje que el prototipo quiere transmitir, este es aún un proyecto en desarrollo en el cual aún queda mucho por diseñar, implementar y mejorar, abriendo un abanico de posibilidades importantes que permitan ofrecer al jugador una experiencia más completa que aborde otros temas relacionados al trastorno de pánico, y de esta forma encaminar el proyecto a una producción profesional de videojuegos.

### Bibliografía

- Adobe. (n.d.). *Learn How to Make Pixel Art: Tutorial with Tips & Tools*. Retrieved April 15, 2022, from Adobe: <https://www.adobe.com/creativecloud/design/discover/pixel-art.html>
- American Psychiatric Association. (2013). *Diagnostic and Statistical Manual of Mental Disorders: DSM-5*. American Psychiatric Association. Retrieved from [http://repository.poltekkes-kaltim.ac.id/657/1/Diagnostic%20and%20statistical%20manual%20of%20mental%20disorders%20\\_%20DSM-5%20%28%20PDFDrive.com%20%29.pdf](http://repository.poltekkes-kaltim.ac.id/657/1/Diagnostic%20and%20statistical%20manual%20of%20mental%20disorders%20_%20DSM-5%20%28%20PDFDrive.com%20%29.pdf)
- Anas, B. (2020, Mayo 26). *Understanding color theory: the color wheel and finding complementary colors*. Retrieved April 15, 2022, from InVision: <https://www.invisionapp.com/inside-design/understanding-color-theory-the-color-wheel-and-finding-complementary-colors/>
- Aseprite. (n.d.). *Docs*. Retrieved April 16, 2022, from Aseprite - Docs: <https://www.aseprite.org/docs/>
- Bach, D. (2021, Mayo 3). *Centro de noticias - Microsoft*. Retrieved from <https://news.microsoft.com/source/features/work-life/mind-games-how-gaming-can-play-a-positive-role-in-mental-health/>
- Bruce, G. (2022, Marzo 17). *You Gove*. Retrieved from <https://business.yougov.com/content/41600-us-charting-rise-indie-video-games>
- Buckland, M. (2004). *Programming game AI by example*. United States of America: Wordware Publishing, Inc.
- Burke, D., & Collins, D. (2022, Febrero 22). *What You Need to Know About Panic Attacks and Panic Disorder*. Retrieved Marzo 29, 2022, from Healthline: <https://www.healthline.com/health/panic-disorder>
- Clip Studio paint. (n.d.). *Guide to Creating Color Schemes | Art Rocket*. Retrieved April 15, 2022, from Clip Studio paint: <https://www.clipstudio.net/how-to-draw/archives/156922>
- Dattani, S., Ritchie, H., & Roser, M. (2021, Agosto). *Mental Health*. Retrieved March 22, 2022, from Our World in Data: <https://ourworldindata.org/mental-health#>
- DeGuzman, K. (2021, Septiembre 19). *What is Character Design — Tips on Creating Iconic Characters*. Retrieved April 15, 2022, from StudioBinder: <https://www.studiobinder.com/blog/what-is-character-design-definition/>
- Forcadell López, E., Fullana Rivas, M. À., Lázaro García, L., & Lera Miguel, S. (2019, Enero 29). *Trastornos de Ansiedad | ¿Qué es la Ansiedad? | PortalCLÍNICA*. Retrieved Marzo 29, 2022, from Hospital Clínic Barcelona: <https://www.clinicbarcelona.org/asistencia/enfermedades/trastornos-de-ansiedad>



- Gómez Ayala, A. E. (2012, Noviembre). Trastornos de ansiedad. Agorafobia y crisis de pánico. *Farmacia profesional*, 26(6), 32-39. Retrieved from <https://www.elsevier.es/es-revista-farmacia-profesional-3-articulo-trastornos-ansiedad-agorafobia-crisis-panico-X0213932412678054>
- Gregory, J. (2017). *Game Engine Architecture* (Segunda ed.). CRC Press. Retrieved from <http://ce.eng.usc.ac.ir/files/1511334027376.pdf>
- Healthline. (2022, Febrero 17). *Stress and Anxiety: How They Differ and How to Manage Them*. Retrieved Marzo 28, 2022, from Healthline: <https://www.healthline.com/health/stress-and-anxiety>
- Hunicke, R., LeBlanc, M., & Zubek, R. (n.d.). *MDA: un enfoque formal al diseño y la investigación de juegos*. Retrieved Abril 30, 2022, from [http://fba.unlp.edu.ar/lenguajemm/?wpfb\\_dl=103](http://fba.unlp.edu.ar/lenguajemm/?wpfb_dl=103)
- Jones, M. (n.d.). *Character Design: What It Is and Why It's Important*. Retrieved Abril 15, 2022, from MoGraph Mentor: <https://www.mographmentor.com/education/character-design-what-it-is-and-why-its-important/>
- Kim, J.-Y., Lee, W.-H., & Kim, J.-T. (2013, Septiembre). *A Study on Expression of the Digital Game Art Form*. Retrieved Marzo 26, 2022, from ResearchGate: [https://www.researchgate.net/publication/274612763\\_A\\_Study\\_on\\_Expression\\_of\\_the\\_Digital\\_Game\\_Art\\_Form](https://www.researchgate.net/publication/274612763_A_Study_on_Expression_of_the_Digital_Game_Art_Form)
- Krita. (2009, Diciembre 24). *Krita Features*. Retrieved Abril 15, 2022, from Krita: <https://krita.org/en/item/krita-features/>
- Manileve, V. (2021, Diciembre 17). *What is Narrative Design?* Retrieved Abril 15, 2022, from Ubisoft News: <https://news.ubisoft.com/en-gb/article/7m412GLSbfkaT0YheRYLVG/what-is-narrative-design>
- MasterClass. (2020, Noviembre 8). *Guide to Video Game Genres: 10 Popular Video Game Types - 2022*. Retrieved Abril 13, 2022, from MasterClass: <https://www.masterclass.com/articles/guide-to-video-game-genres#10-popular-video-game-genres>
- MasterClass. (2020, Noviembre 8). *How to Become a Video Game Level Designer - 2022*. Retrieved Abril 14, 2022, from MasterClass: <https://www.masterclass.com/articles/how-to-become-a-video-game-level-designer#quiz-0>
- MasterClass staff. (2020, Noviembre 8). *How to Design a Video Game Character - 2022*. Retrieved Abril 15, 2022, from MasterClass: <https://www.masterclass.com/articles/how-to-design-a-video-game-character#how-to-design-a-video-game-character>

- Mayo Clinic. (2018, Noviembre 20). *Los ataques de pánico y el trastorno de pánico - Síntomas y causas*. Retrieved Marzo 29, 2022, from Mayo Clinic: <https://www.mayoclinic.org/es-es/diseases-conditions/panic-attacks/symptoms-causes/syc-20376021>
- Mayo Clinic. (2021, Noviembre 11). *Trastornos de ansiedad - Síntomas y causas*. Retrieved Marzo 28, 2022, from Mayo Clinic: <https://www.mayoclinic.org/es-es/diseases-conditions/anxiety/symptoms-causes/syc-20350961>
- Microsoft .NET. (n.d.). *Unity Real-Time Development Platform | .NET*. Retrieved Abril 14, 2022, from Unity Powered by .NET: <https://dotnet.microsoft.com/en-us/apps/games/unity>
- Microsoft. (2022, Marzo 1). *Overview of Visual Studio*. Retrieved Abril 14, 2022, from Microsoft Docs: <https://docs.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2022>
- Nanou, E. (2021, Diciembre 11). *6 Differences Between Action Games and Adventure Games*. Retrieved Abril 13, 2022, from MakeUseOf: <https://www.makeuseof.com/action-games-adventure-games-differences/>
- National Institute of Mental Health. (2017, Febrero). *Trastorno de pánico: Cuando el miedo agobia*. Retrieved March 26, 2022, from National Institute of Mental Health: <https://www.nimh.nih.gov/health/publications/espanol/trastorno-de-panico-cuando-el-miedo-agobia>
- National Institute of Mental Health. (2018). *Anxiety Disorders*. Retrieved March 22, 2022, from National Institute of Mental Health: <https://www.nimh.nih.gov/health/topics/anxiety-disorders>
- National Institute of Mental Health. (n.d.). *NIMH » ¡Estoy tan estresado! hoja informativa*. Retrieved Marzo 28, 2022, from National Institute of Mental Health: <https://www.nimh.nih.gov/health/publications/espanol/estoy-tan-estresado>
- National Storytelling Network. (n.d.). *What Is Storytelling? – National Storytelling Network*. Retrieved Abril 15, 2022, from National Storytelling Network: <https://storynet.org/what-is-storytelling/>
- Nemours KidsHealth. (2013, Mayo). *Trastornos de ansiedad (para Adolescentes) - Nemours KidsHealth*. Retrieved Marzo 28, 2022, from Kids Health: <https://kidshealth.org/es/teens/anxiety.html>
- New England Institute of Technology. (2020, Mayo 25). *Video Game Design vs Video Game Development*. Retrieved Abril 14, 2022, from

- New England Institute of Technology: <https://www.neit.edu/blog/difference-between-game-designer-and-game-developer>
- Organización Mundial de la Salud. (2019, Noviembre 28). *Trastornos mentales*. Retrieved Marzo 28, 2022, from WHO | World Health Organization: <https://www.who.int/es/news-room/fact-sheets/detail/mental-disorders>
- Parker, L. (2019, Marzo 24). Depressed and Anxious? These Video Games Want to Help. *The New York Times*. Retrieved from <https://www.nytimes.com/2019/03/24/technology/personaltech/depression-anxiety-video-games.html>
- Regil Vargas, L. (2005, Octubre 10). Hipermedia: Medio, lenguaje herramienta del arte digital. *Revista Digital Universitaria (RDU)*, 6(10), 18. Retrieved from [https://www.revista.unam.mx/vol.6/num10/art97/oct\\_art97.pdf](https://www.revista.unam.mx/vol.6/num10/art97/oct_art97.pdf)
- Schell, J. (2015). *The Art of Game Design: A Book of Lenses* (Segunda ed.). CRC Press. Retrieved from [https://iums.ac.ir/uploads/%5BJesse\\_Schell%5D\\_The\\_Art\\_of\\_Game\\_Design\\_A\\_book\\_of\\_l\(BookFi\).pdf](https://iums.ac.ir/uploads/%5BJesse_Schell%5D_The_Art_of_Game_Design_A_book_of_l(BookFi).pdf)
- Starloop Studios. (2021, Junio 11). *How to Use Storytelling and Transmedia in Video Games?* Retrieved Abril 15, 2022, from Starloop Studios: <https://starloopstudios.com/how-to-use-storytelling-and-transmedia-in-video-games/>
- Tokio School. (2019, Noviembre 12). *Inicio Back Up Blog ¿Qué es el Game Design Document?* Retrieved Abril 12, 2022, from Tokio School: <https://www.tokioschool.com/noticias/que-es-game-design-document/>
- Tokio School. (2020, Enero 28). *¿Qué son las dinámicas de videojuegos?* Retrieved Abril 30, 2022, from Tokio School: <https://www.tokioschool.com/noticias/que-son-dinamicas-videojuegos/>
- Totten, C. (2014). *An Architectural Approach to Level Design* (Primera ed.). CRC Press.
- Will Harris, S. L. (2021, Mayo 25). *TheGamer*. Retrieved from <https://www.thegamer.com/mental-illness-health-video-games/#we-happy-few>
- World Health Organization. (2018, Marzo 30). *Mental health: strengthening our response*. Retrieved Marzo 28, 2022, from WHO | World Health Organization: <https://www.who.int/news-room/fact-sheets/detail/mental-health-strengthening-our-response>

## Anexos

### Anexo A. Índice de figuras

|                                                                                     |    |
|-------------------------------------------------------------------------------------|----|
| Figura 1 Esquema de controles para Windows -----                                    | 38 |
| Figura 2 Boceto del menú principal -----                                            | 40 |
| Figura 3 Boceto del nivel -----                                                     | 41 |
| Figura 4 Boceto del menú de pausa -----                                             | 42 |
| Figura 5 Boceto de pantalla de game over -----                                      | 43 |
| Figura 6 Boceto de tutorial -----                                                   | 44 |
| Figura 7 Boceto de pantalla de créditos -----                                       | 44 |
| Figura 8 Diagrama del flujo de interacciones de los estados del videojuego -----    | 45 |
| Figura 9 Diseño visual del personaje principal -----                                | 48 |
| Figura 10 Diseño visual del enemigo -----                                           | 49 |
| Figura 11 Línea de visión del nivel -----                                           | 52 |
| Figura 12 Estructura del GameObject Player -----                                    | 61 |
| Figura 13 Mapa de acciones de Player -----                                          | 62 |
| Figura 14 Animation curve que describe la aceleración del personaje principal ----- | 63 |
| Figura 15 Alcance máximo de desplazamiento -----                                    | 63 |
| Figura 16 Alcance de desplazamiento definido por un Raycast -----                   | 64 |
| Figura 17 Configuración de la CinemachineVirtualCamera -----                        | 65 |
| Figura 18 Diagrama de máquina de estados finita -----                               | 66 |
| Figura 19 Progreso del Slider del prototipo de la barra de estado mental -----      | 69 |
| Figura 20 Estructura del GameObject Enemy -----                                     | 71 |
| Figura 21 Diagrama de máquina de estados finita del enemigo -----                   | 73 |
| Figura 22 Estructura del objeto Parallax -----                                      | 75 |
| Figura 23 Implementación del efecto paralaje (Parallax) -----                       | 76 |
| Figura 24 Animator de jugador y enemigo (de arriba a abajo) -----                   | 77 |
| Figura 25 Interfaces de Usuario del nivel -----                                     | 79 |
| Figura 26 Bocetos del personaje principal -----                                     | 81 |
| Figura 27 Propuesta final de diseño de personaje principal -----                    | 82 |
| Figura 28 Bocetos de enemigo -----                                                  | 83 |
| Figura 29 Propuesta final de diseño de enemigo -----                                | 83 |
| Figura 30 Bocetos del diseño de fondo -----                                         | 85 |
| Figura 31 Pruebas de composición y paleta de color -----                            | 85 |

|                                                                    |     |
|--------------------------------------------------------------------|-----|
| Figura 32 Propuesta final de fondo-----                            | 86  |
| Figura 33 Proceso de diseño de imagotipo -----                     | 87  |
| Figura 34 Propuesta final de diseño final de imagotipo -----       | 87  |
| Figura 35 Bocetos del diseño de la barra de estado mental -----    | 88  |
| Figura 36 Propuesta final de la barra de estado mental-----        | 88  |
| Figura 37 Assets utilizados para el proyecto, de Kenney.nl-----    | 90  |
| Figura 38 Sprite sheet del personaje principal-----                | 92  |
| Figura 39 Sprite sheet del enemigo -----                           | 93  |
| Figura 40 Clips de animación de la barra de estado mental -----    | 94  |
| Figura 41 Implementación del fondo en unity -----                  | 95  |
| Figura 42 Implementación de las animaciones en Unity -----         | 96  |
| Figura 43 Implementación de postprocesado y efectos visuales ----- | 97  |
| Figura 44 Implementación del sistema de partículas-----            | 98  |
| Figura 45 Capturas de pantalla del videojuego desde Itch.io-----   | 100 |

## Anexo B. Códigos C#

### *Mover un objeto de un punto a otro*

```
ObjPersonaje.transform.position =
Vector3.Lerp(PosInicial, PosFinal, player.acelCurve.Evaluate(player.timeElapsed));
```

### *Guardar la información de colisión de un Physics2D.Raycast*

```
RaycastHit2D enemigoInfo =
Physics2D.Raycast(puntoInicial.transform.position, direction, rayDistance, enemiesMask);
```

### *Visualizar en el editor un Raycast*

```
Debug.DrawRay(puntoInicial.transform.position, direction * rayDistance, Color.yellow, 2f);
```

### *Clase base de los diferentes estados*

```
public abstract class PlayerBaseState
{
    public abstract void OnEnterState(PlayerStateManager player);

    public abstract void UpdateState(PlayerStateManager player);

    public abstract void OnExitState(PlayerStateManager player);
}
```

### *Estructura base para definición de estados de jugador*

```
public class Idle : PlayerBaseState
{
    public override void OnEnterState(PlayerStateManager player)
    {

    }

    public override void UpdateState(PlayerStateManager player)
    {

    }
}
```

```

public override void OnExitState(PlayerStateManager player)
{

}
}

```

### ***Clase PlayerStateManager***

```

public class PlayerStateManager : MonoBehaviour
{
    //Crear instancias de cada uno de los estados
    public PlayerMovement MovementState = new PlayerMovement();
    public PlayerIdle idleState = new PlayerIdle();
    public PlayerKnockedOut KnockedOut = new PlayerKnockedOut();
    public DetectEnemies detectEnemies = new DetectEnemies();
    public Rest restState = new Rest();
    public Dead deadState = new Dead();
    public MentalBreakdownState mentalBreakdown = new MentalBreakdownState();

    void Start()
    {
        currentState = idleState;
        currentState.OnEnterState(this);
    }

    void Update()
    {
        currentState.UpdateState(this);
    }

    public void SwitchState(PlayerBaseState state)
    {
        currentState.OnExitState(this);
        previousState = currentState;
        currentState = state;
        state.OnEnterState(this);
    }
}

```

***Clase StateController***

```

public class StateController : MonoBehaviour
{
    public State CurrentState;
    [SerializeField] private State _followPlayer;

    private void OnEnable()
    {
        CurrentState = _followPlayer;
    }

    void Update()
    {
        if (!_aiActivate)
        {
            return;
        }
        else
        {
            CurrentState.UpdateState(this);
        }
    }

    public void TransitionToState(State nextState)
    {
        if (nextState != _remainState)
        {
            CurrentState = nextState;
        }
    }
}

```

***Clase Action***

```

public abstract class Action : ScriptableObject
{
    public abstract void Act(StateController controller);
}

```



***Clase Desicion***

```
public abstract class Desicion : ScriptableObject
{
    public abstract bool Decide(StateController controller);
}
```

***Clase Transition***

```
public class Transition
{
    public Desicion Desicion;
    public State TrueState;
    public State FalseState;
}
```

***Clase State***

```
public class State : ScriptableObject
{
    public Action[] Actions;
    public Transition[] Transitions;

    public void UpdateState(StateController controller)
    {
        DoActions(controller);
        CheckTransitions(controller);
    }

    private void DoActions(StateController controller)
    {
        for (int i = 0; i < Actions.Length; i++)
        {
            Actions[i].Act(controller);
        }
    }

    private void CheckTransitions(StateController controller)
    {
        foreach (var transition in Transitions)
        {
            bool desicion = transition.Desicion.Decide(controller);
        }
    }
}
```

```
    if (desicion)
    {
        controller.TransitionToState(transition.TrueState);
    }
    else
    {
        controller.TransitionToState(transition.FalseState);
    }
}
}
```