

# V(add10) Versión 1.0

Documento descriptivo de programa con  
fines de Registro de Software ante la  
Dirección Nacional de Derechos de Autor

Por: Alejandro De la Espriella Mazo  
Dany Stevan Rios Madrid  
[proyectotesisvideojuego@gmail.com](mailto:proyectotesisvideojuego@gmail.com)

**Artes de la Grabación y Producción Musical**  
**Facultad de Artes y Humanidades**  
**ITM**  
**2021**

# Contenido

---

1. INTRODUCCIÓN.....	4
2. CONTEXTUALIZACIÓN .....	4
3. PROPÓSITO.....	7
4. FUNCIONALIDADES .....	7
5. DIAGRAMA DE CONCEPTO.....	9
6. DIAGRAMA DE FUNCIONES.....	10
7. DETECCIÓN DE COMANDOS.....	11
8. INTERFAZ DE USUARIO.....	18
9. PRUEBAS Y FUNCIONAMIENTO.....	19
10. GLOSARIO.....	22

# Lista de imágenes

---

Imagen 1. Diagrama de concepto .....	9
Imagen 2. Diagrama de funciones .....	10
Imagen 3. Personaje principal y componentes.....	11
Imagen 4. Animator personaje principal y parámetros.....	12
Imagen 5. Menú principal y componentes .....	12
Imagen 6. Enemigo fungus y componentes.....	13
Imagen 7. Enemigo Bat eye y componentes.....	13
Imagen 8. Enemigo skeleton y componentes.....	13
Imagen 9. Enemigo slime y componentes .....	14
Imagen 10. Jefe Bringer of death y componentes .....	14
Imagen 11. Animator Skeleton y parámetros.....	15
Imagen 12. Ambiente lago.....	16
Imagen 13. Ambiente montaña .....	16
Imagen 14. Ambiente selva.....	17
Imagen 15. Objetos destruibles .....	17
Imagen 16. Poder y vida.....	17
Imagen 17. Santuario .....	18
Imagen 18. Virgen guía .....	18
Imagen 19. Elementos de interfaz .....	18
Imagen 20. Características computador gama baja.....	19
Imagen 21. Características computador gama media .....	20
Imagen 22. Características computador gama alta .....	20
Imagen 23. Indicadores personal de prueba .....	21
Imagen 24. Indicadores personal de prueba .....	21
Imagen 25. Indicadores personal de prueba .....	21

# 1. INTRODUCCIÓN

---

El proyecto V(add10) surge del interés por investigar y aprender sobre la aplicación de la producción musical en la sonorización de productos audiovisuales interactivos como lo son los videojuegos. Del mismo modo, nace del interés por entender el funcionamiento interno de un videojuego y las posibilidades de articulación con el sonido a través de un lenguaje de programación.

El videojuego V(add10) fue concebido en la plataforma de desarrollo en tiempo real Unity en conjunto con el editor de código Visual Studio. Por su parte, el diseño sonoro, la grabación de audio y la composición musical fueron desarrollados en las estaciones de audio digital Studio One y Ableton Live. Para la integración del sonido en Unity se usó el motor de efectos sonoros FMOD.

V(add10) es un videojuego de acción en formato 2D, que se ambienta en la selva colombiana de la región del Chocó, y adapta los entornos sonoros y la música acorde a los eventos en pantalla.

## 2. CONTEXTUALIZACIÓN

---

### **Narrativa**

Uno de los elementos de los que se vale la narrativa en los videojuegos es la no-linealidad de la historia: permite al jugador la construcción aleatoria de la misma, o la ralentiza hasta que ocurra un avance significativo dentro del juego, que amerite la activación de nuevos fragmentos narrativos. Sin embargo, las historias no deben ser necesariamente complejas, ni con una estructura narrativa robusta como se acostumbra en otros productos audiovisuales como el cine, tal y como lo menciona Green:

No todos los juegos cuentan historias, algunos simplemente no tienen una historia propia y no deben ser catalogados como fracasos por ello. Otros pueden tender a contar una historia, pero fallan en el nivel narrativo de la misma en comparación con otros juegos con otro formato narrativo, presentando insatisfactoriamente una historia que enganche, sea interesante o convincente.<sup>1</sup> (Green, 2018)

Por su parte, Planells de la Maza explica cómo la narrativa al concebirse como un elemento no lineal dentro de la historia del juego, permite al jugador ser partícipe de esta, alterándola en cuanto a su tiempo de desarrollo, orden y conclusión. Esto da como resultado una narrativa particular, que depende de las decisiones o intervenciones del jugador.

---

<sup>1</sup> Not all video games tell stories – some simply do not have that as their aim and should not be faulted for it. Others may attempt to tell stories, but fail on a narrative level in the same way that their counterparts in other storytelling formats may prove unsuccessful in presenting a gripping, interesting, or compelling story. (Traducción nuestra)

El autor entiende que la narrativa lineal se caracteriza por historias cerradas, relatos preestablecidos y la hegemonía de un narrador. Es el caso del cine o las novelas, donde es habitual encontrar la estructura aristotélica de los tres actos: inicio, nudo y desenlace.

Este modelo se contrapone a la narrativa interactiva constituida por historias abiertas con relatos variables. Es decir, no existe un principio y fin único y preestablecido, sino que caben varios posibles inicios, así como también múltiples finales.

En cualquier caso, la selección de estos caminos narrativos y la conformación del relato recaen en manos del usuario. (Planells de la Maza, 2010)

Lo anterior presenta como sustento, la forma en la que se construye la narrativa en los videojuegos, la cual, puede estar explícita desde el inicio de estos, o puede constituirse de fragmentos que sirvan de guía al jugador: mientras este, a medida que evoluciona el juego, construye y une las piezas narrativas que se encuentran inmersas en la trama y que pueden ser aleatorias o lineales conforme al formato del juego.

### **Desarrollo y diseño de videojuegos**

Unity ofrece un sistema completo de herramientas para la creación de ambientes visuales, animación de personajes y objetos, sistemas de partículas y manipulación de efectos, así como un motor de sonido, representando para el usuario lo necesario para el diseño e implementación 2D y 3D de eventos interactivos.

La creación y diseño de un videojuego involucra diversos elementos que componen su funcionalidad. Al igual que en las producciones cinematográficas, los desarrollos de videojuegos se trabajan por equipos que atienden las necesidades de diseño gráfico, animación, programación, cinemáticas, efectos, físicas, mecánicas de jugabilidad, diseño sonoro, programación de audio, composición musical, diseño adaptativo sonoro, diálogos, narrativa, entre otros.

El componente interactivo representa un reto mayor a la hora de crear un videojuego: se convierte en la piedra angular e impone la premisa de que, a pesar de tener un escenario con una historia y elementos lineales, el juego es un producto no lineal que no tiene definidos los momentos en los que el jugador va a interactuar. Se asume de esta manera que cada una de las variables contempladas inicialmente en las etapas de diseño tiene un componente de aleatoriedad que permite la interacción por parte del usuario en cualquier momento sin ver interrumpida o violentada la continuidad de la historia y eventos. (Collins, 2008)

### **Diseño sonoro**

FMOD es un motor de audio que permite la integración directa con plataformas de desarrollo de videojuegos como Unity y Unreal Engine. Este motor hace posible la construcción de eventos sonoros 2D y 3D para dar respuesta a diferentes necesidades de diseño de productos interactivos.

El diseño sonoro en un videojuego no se aborda a partir de la lógica de la sincronización lineal como en los productos cinematográficos, sino que se desarrolla a partir de eventos que se activan o desactivan conforme el jugador realiza acciones o avanza a través de este.

Los eventos sonoros en un videojuego pueden subdividirse en dos:

- 2D, que contienen aquellos sonidos que estarán presentes durante todo el tiempo de ejecución, sin que su ubicación dependa de la posición del jugador.
- 3D, que son aquellos que hacen uso de la herramienta de ecolocación que le proporciona al jugador la posibilidad de buscar la fuente que emite el sonido.

El diseño sonoro y la musicalización de un videojuego van en función de atrapar o enganchar al jugador en el universo planteado y que, en la medida que se logren avances, el jugador pueda apreciar eventos sonoros dinámicos que contribuyan con el enriquecimiento de la historia planteada.

La música y el sonido se convierten en elementos narrativos fundamentales, ya que, en articulación con los elementos visuales, procuran la construcción de un escenario de tensión, generando el clímax que el jugador debe experimentar dentro del videojuego.

Los eventos narrativos no se articulan necesariamente de manera lineal o secuencial, sino que estos deberán contener un componente de aleatoriedad que permita al jugador crear su propia línea narrativa conforme a los avances que va logrando, a partir de su evolución en el juego. (Collins, 2008)

### **Música generativa**

La música generativa es un concepto que se le atribuye a Brian Eno. Consiste principalmente en hacer sonar una pieza musical de forma distinta cada vez que llega al oyente, tanto en vivo como en formatos pregrabados, por lo que puede generar diferentes emociones en el espectador.

La música generativa debe estar ligada a un medio de reproducción aleatorio seleccionado por el autor de la pieza con anterioridad. Un ejemplo podría ser un arpegiador MIDI en modo aleatorio. En este caso, dicho arpegiador generará notas en diferente orden cada vez que sea interpretado, lo que producirá la sensación de impredecibilidad de la pieza, creando lo que se conoce como un “tema generativo”.

Un tema generativo consta de una serie de notas que deben ser interpretadas en órdenes aleatorios cada vez que se presentan. Este tipo de música supone un problema con los medios de distribución actuales, ya que formatos como el mp3, wav, OGG, etc., hacen de la música, sus motivos y sus temas como algo preestablecido que llegará igual a los oyentes cada que sean reproducidos. Por lo anterior, la música generativa adquiere validez sólo cuando se interpreta en vivo.

Actualmente, con el avance tecnológico puede hallarse una alternativa potencial a la imposibilidad de reproducir música generativa a partir de formatos de audio convencionales: los motores de audio de los videojuegos, con scripts que generen aleatoriedad sobre los elementos a reproducir (como las notas de un tema musical), podrían convertir la música en algo generativo.

## 3. PROPÓSITO

---

V (add10) surgió de la iniciativa de explorar y profundizar en el campo del diseño sonoro, sonorización mediante efectos y ambientes, y musicalización aplicada a los videojuegos, articulando conceptos de planeación y desarrollo hasta concebir el producto final.

El presente videojuego implementa eventos interactivos y dinámicos apoyados en el diseño sonoro y la música, contribuyendo a la generación de eventos adaptativos desde lo sonoro hacia lo visual, conforme a las interacciones que ocurren en escenarios no lineales, diferentes a los propuestos en otros productos audiovisuales de construcción narrativa lineal.

## 4. FUNCIONALIDADES

---

V (add10) es un software que procesa datos en tiempo real, creando un entorno virtual adaptable a la situación del jugador a partir de las siguientes funciones:

- 1. Detecta comandos y ejecuta acciones en consecuencia** Sistema diseñado para reconocer las entradas de comandos que el jugador ingresa a través de teclado del computador, y ejecuta las acciones y eventos correspondientes. Existen dos casos en los cuales los comandos varían:
  - Menús: permiten al jugador acceder y navegar entre diferentes funciones de juego por medio de los botones del ratón o de las teclas de navegación del teclado para moverse entre las opciones. Entre las funciones se encuentran “mapa”, “inicio” y “cargado de partidas” y “salir del juego”.
  - Personaje principal *in game*:<sup>2</sup> ejecuta las acciones saltar, atacar, correr e interactuar con objetos y personajes no jugables (NPCs) dentro del escenario. Para ello se realiza el reconocimiento constante de las teclas de navegación del teclado, tecla de función “F2”, tecla de escritura “Space” y tecla de control “Ctrl”.
- 2. Activa, desactiva y adapta eventos sonoros de FMOD acorde con los eventos de Unity realizados por el jugador** Sucesos interactivos que se ejecutan de acuerdo con las interacciones que tiene el jugador con el entorno, permitiéndole una experiencia sonora

---

<sup>2</sup> Personaje principal *in game* hace referencia exclusivamente al momento del juego donde el personaje está activo y es controlable por el jugador. Cuando el menú principal, la vista del mapa o una cinemática está activa, no forman parte del caso “personaje principal in game”.

realista que refuerza los movimientos, efectos y acciones visuales del juego. Existen cuatro fuentes sonoras principales, las cuales se adaptan continuamente y ejecutan sonidos resultantes de las interacciones:

- **Personaje Principal:** su diseño sonoro contiene elementos Foley y efectos de cada una de sus acciones como lo son saltar, correr, recibir daño, atacar y muerte.
- **Ambientes:** el juego está dividido en zonas con una ambientación que refuerza el paisaje visual y presenta elementos inspirados en la selva colombiana del Chocó.
- **Enemigos:** reflejan un diseño sonoro fantasioso que contiene elementos Foley y efectos de cada una de sus acciones, además permiten al jugador el reconocimiento sonoro de su ubicación por medio de ecolocación.
- **Objetos:** son elementos que al interactuar con el jugador principal ejecutan acciones especiales como “adición de vida” y “ruptura de objetos”, viajar mediante “puertas entre escenarios<sup>3</sup>” y otros elementos interactivos del juego.

**3. Adecuación de ambientación musical acorde al estado del jugador** El programa hace un reconocimiento permanente de los condicionales que dictan el estado del juego y adapta la música acorde con ellos. Dichos estados son:

- **Calma:** estado que se activa mientras el jugador permanece estático o dentro del menú.
- **Exploración:** es activado en el momento que el jugador comienza a recorrer el escenario sin entrar en estado batalla.
- **Batalla:** momento en el cual el jugador se enfrenta en un combate importante. Como ejemplo, la “zona de combate” ubicada en la escena No. 8.

---

<sup>3</sup> Las puertas en el contexto de los videojuegos hacen alusión a un elemento interactivo cuya función es conectar 2 escenas. Por ejemplo: la puerta que permite el paso del nivel 1 hacia el nivel 2.



## 5. DIAGRAMA DE CONCEPTO

---

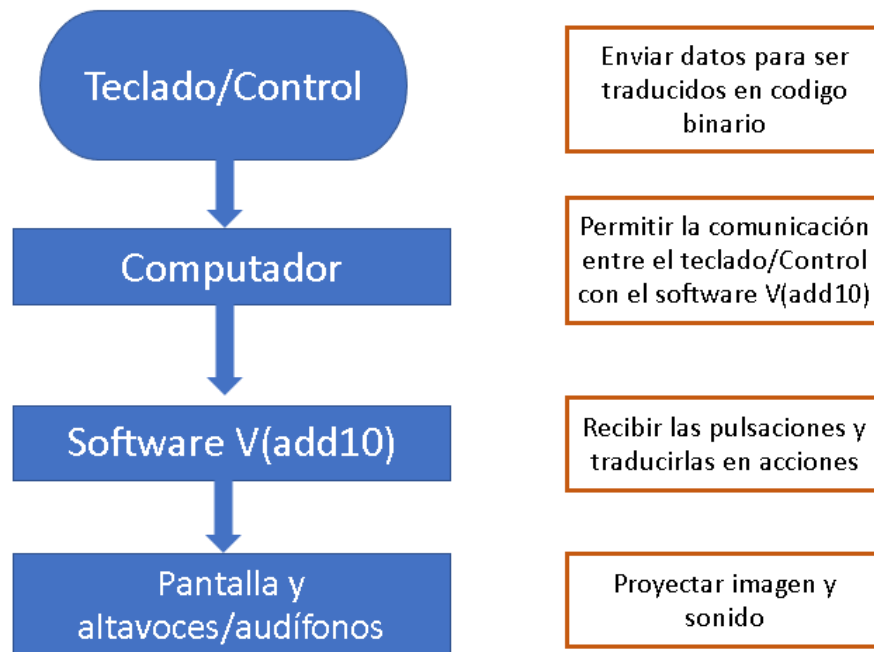


Imagen 1. Diagrama de concepto

# 6. DIAGRAMA DE FUNCIONES

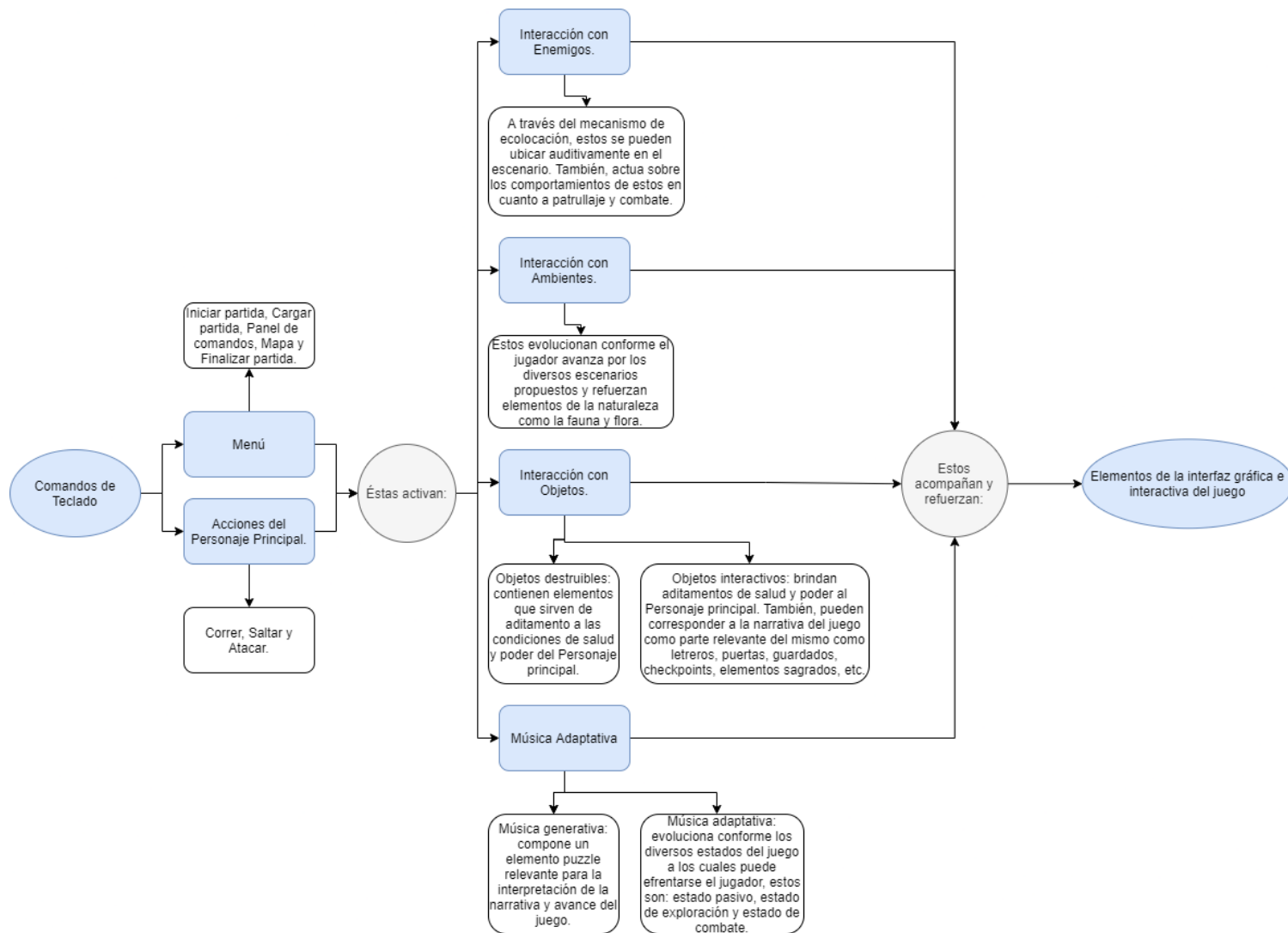


Imagen 2. Diagrama de funciones

## 7. DETECCIÓN DE COMANDOS

---

- Personaje Principal:

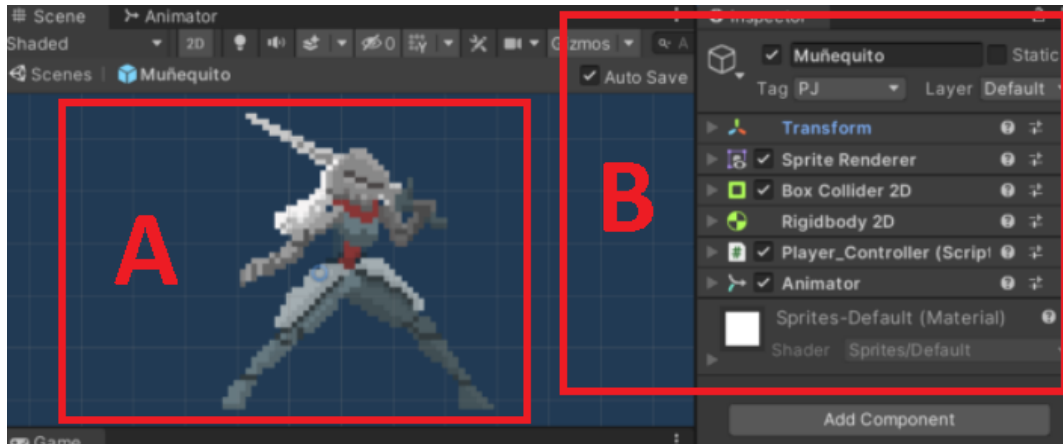


Imagen 3. Personaje principal y componentes

**A:** Previsualización del personaje principal.

**B:** Apartado para agregar componentes que permitan el funcionamiento del personaje dentro del juego. Dichos componentes son:

- **Transform:** define la ubicación del personaje en el espacio.
- **Sprite renderer:** permite mostrar una imagen en pantalla como *Sprite*.
- **Box Collider 2D:** elemento que da la capacidad de colisionar con otros elementos de la escena. Es 2D debido a que su eje "Z" se considera infinito.
- **Rigidbody 2D:** elemento que otorga las físicas al objeto equipado.
- **Player\_Controller (Script):** código que ejecuta las acciones del jugador y lo relacionado con él.
- **Animator:** elemento que permite animar al personaje en sus diferentes estados: *Idle*, *Run*, *Jump*, *Fall*, *Attack 1*, *Attack 2*. El *Animator* contiene diversos elementos que permiten su configuración y manipulación, pero los principales son "Animaciones", "Parámetros" y "Transiciones".

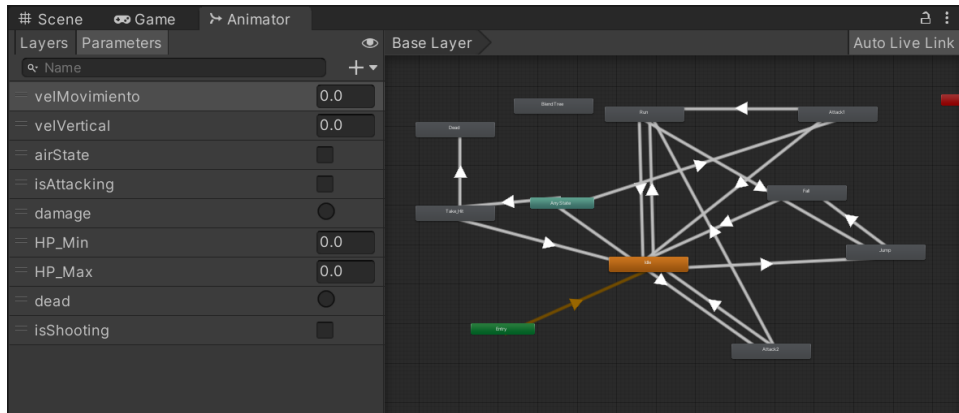


Imagen 4. Animator personaje principal y parámetros

- ✓ **Animaciones:** elemento que permite contener una serie de *Sprites* con un orden en específico. Hace posible la reproducción en secuencia de imágenes para generar la ilusión de movimiento. Son representadas dentro del *Animator* por medio de cajas de color Celeste, Verde, Naranja y gris.
- ✓ **Parámetros:** son una serie de condicionales que delimitan el flujo entre dos o más animaciones. Pueden ser números enteros, números flotantes, *triggers* o *booleanos*. Se ubican en un desplegable a la izquierda del *Animator*.
- ✓ **Transiciones:** están representadas con flechas que salen de las cajas de animaciones. Contienen en su interior uno o más parámetros. La cabeza de la flecha indica el flujo hacia la animación que continuará la secuencia, siempre y cuando llegue a cumplirse una condición establecida durante su configuración.

- Menú:

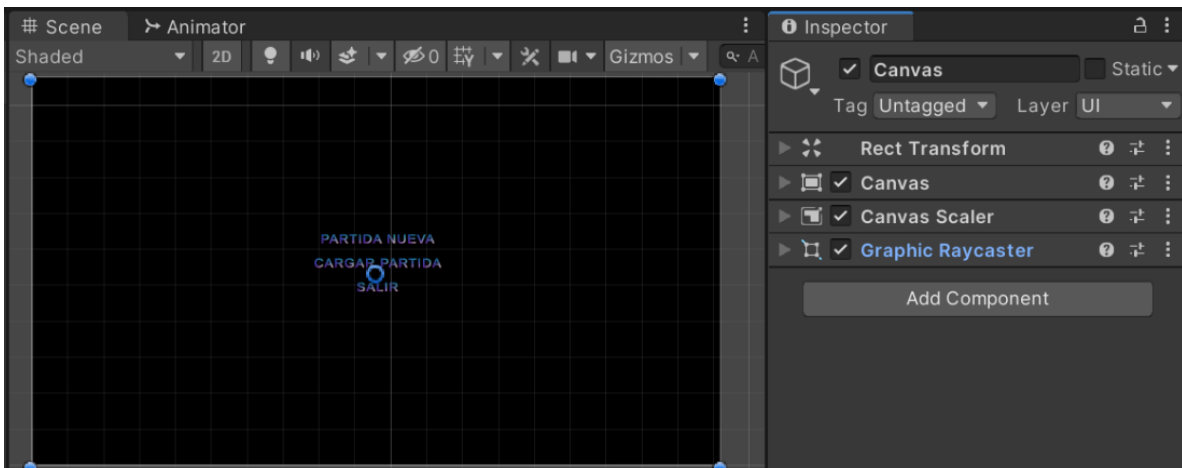


Imagen 5. Menú principal y componentes

- **Canvas:** define la ubicación espacial de los elementos que permanecen fijos en pantalla como: la barra de vida, indicador de poder y letreros.

- Enemigos:

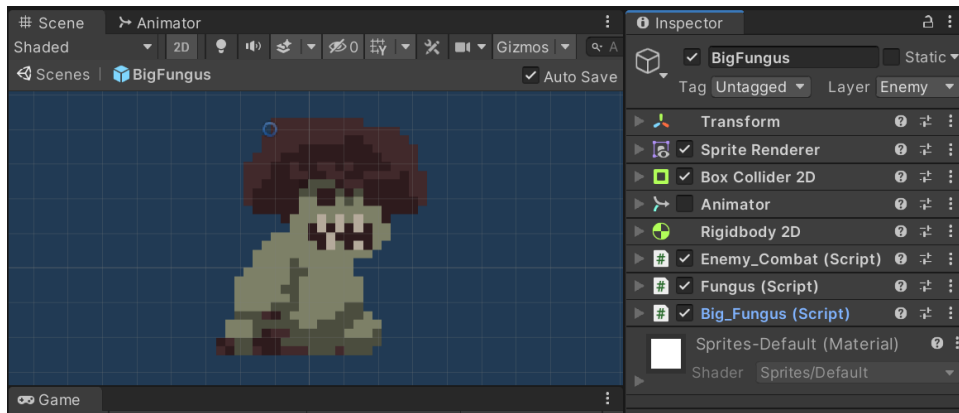


Imagen 6. Enemigo fungus y componentes

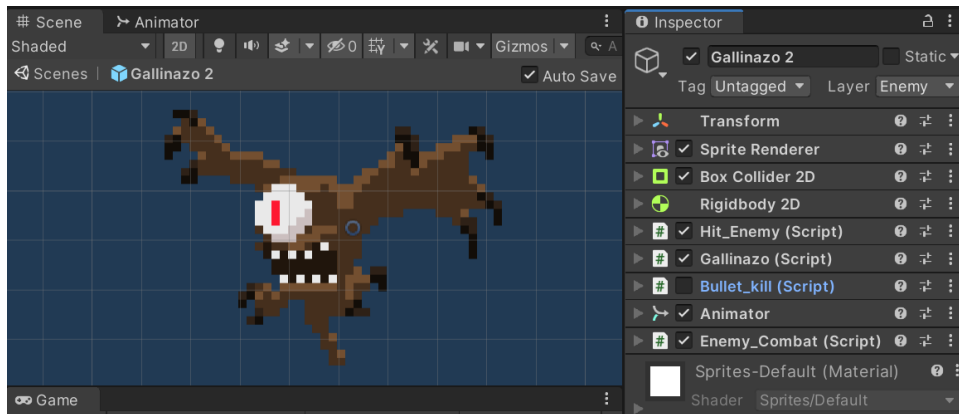


Imagen 7. Enemigo Bat eye y componentes

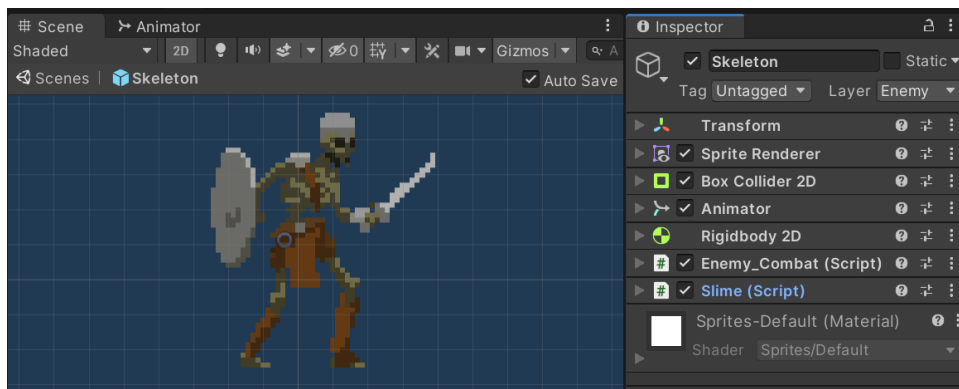


Imagen 8. Enemigo skeleton y componentes

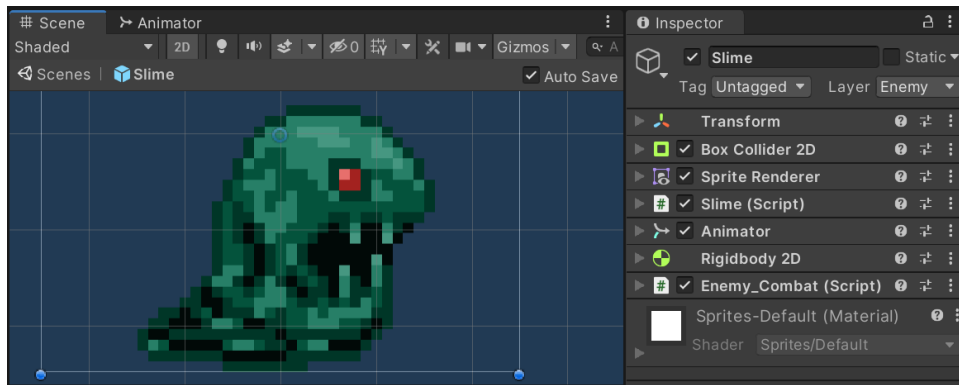


Imagen 9. Enemigo slime y componentes

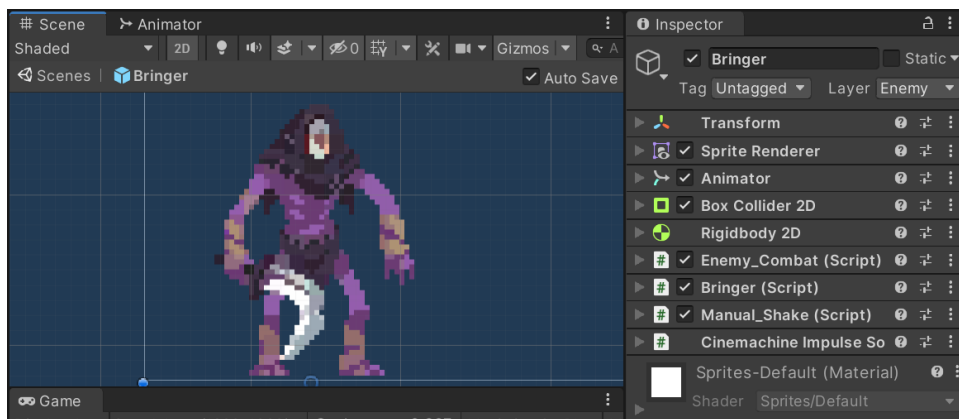


Imagen 10. Jefe Bringer of death y componentes

- **Transform:** define la ubicación del personaje en el espacio.
- **Sprite renderer:** muestra una imagen en pantalla asociada a un *GameObject*.
- **Box Collider 2D:** elemento que permite la colisión con otros elementos de la escena. Es 2D debido a que su eje "Z" se considera infinito.
- **Rigidbody 2D:** elemento que otorga las físicas al objeto equipado.
- **Enemy\_Combat (Script):** código que ejecuta las acciones del enemigo dependiente del tipo de interacción con el jugador.
- **Hit\_Enemy (Script):** este elemento permite a los enemigos causar daño al jugador, reproduciendo en este último la animación correspondiente a cuando es golpeado por un atacante. Además, permite asignar a cada ataque la cantidad de daño que proporcionará.
- **Fungus (Script):** código que genera la inteligencia artificial para el enemigo Fungus, contiene diferentes comportamientos frente a situaciones como: Despertar cuando el jugador pasa cerca de él; patrullar mientras no ve al jugador; perseguir y atacar cuando ve al jugador.

- **Big\_Fungus (Script):** código que genera los comportamientos del enemigo Big Fungus, funciona de manera similar al código Fungus, con la diferencia de que permite a Big Fungus disparar proyectiles.
- **Slime (Script):** Contiene los comportamientos del enemigo Slime, el más básico de los adversarios. Le permite patrullar, perseguir y atacar al jugador.
- **Gallinazo (Script):** es el código que contiene los comportamientos del enemigo Gallinazo. Le permiten volar, realizar ataques tipo “*spin*” y “*fireball*”. El campo de visión del enemigo Gallinazo tendrá forma de esfera, lo cual permitirá a la inteligencia artificial definir un tipo ataque dependiendo de la distancia con respecto al jugador.
- **Bringer (Script):** código que genera la inteligencia artificial para el enemigo Bringer.
- **Manual\_Shake (Script):** código que genera un efecto de temblor en la cámara, es disparado por las animaciones del enemigo Bringer.
- **Bullet\_Kill (Script):** determina el tiempo de vida o permanencia de los proyectiles dentro de la escena.
- **Cinemachine Impulse Source (Script):** código que permite la conexión entre cualquier elemento que lo posea y la cámara para que esta reciba los impulsos y efectos.
- **Animator:** elemento que permite animar al personaje en sus diferentes estados: *Idle*, *Run*, *Jump*, *Fall*, *Attack 1*, *Attack 2*.

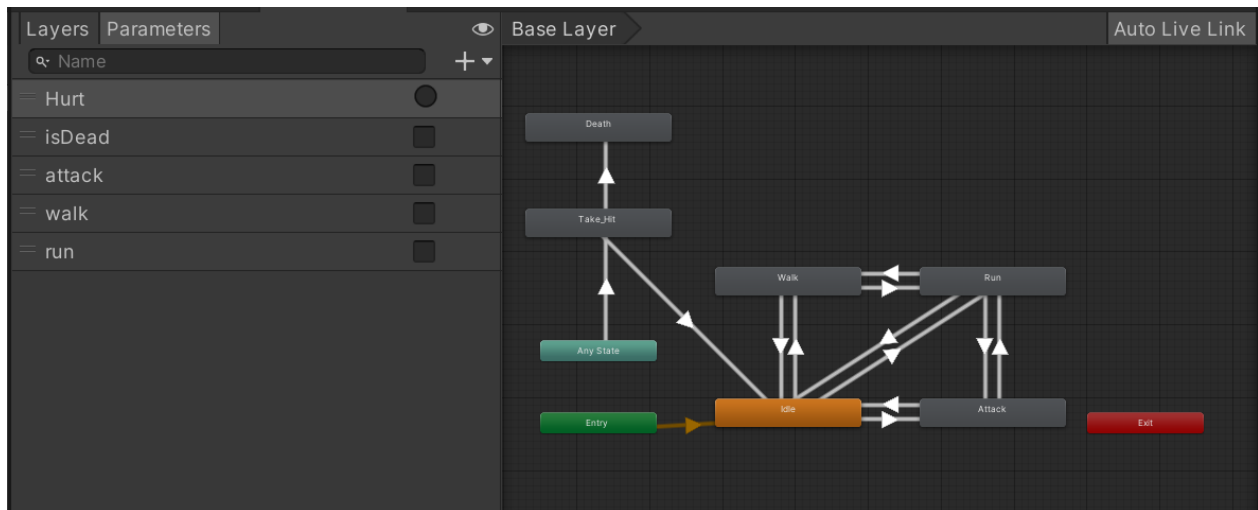


Imagen 11. Animator Skeleton y parámetros

El *Animator* describe, a modo de flujo de señal, la activación de las animaciones donde cada una de las transiciones indica las posibles rutas que pueden activarse en función del estado de animación actual del personaje. Dichas transiciones están regidas por los parámetros (visibles a la izquierda de la imagen) que funcionan a modo de condiciones: de cumplirse la condición, se ejecutará la transición.

- Ambientes:

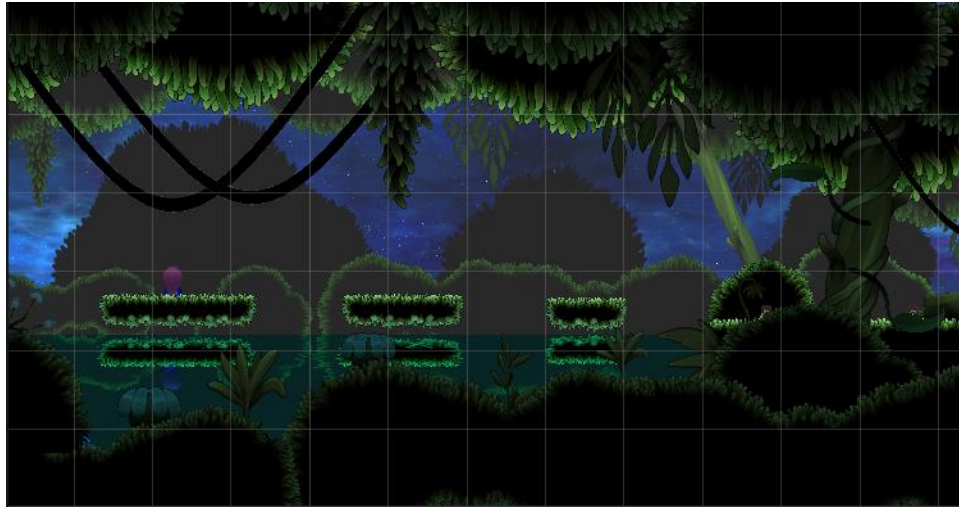


Imagen 12. Ambiente lago

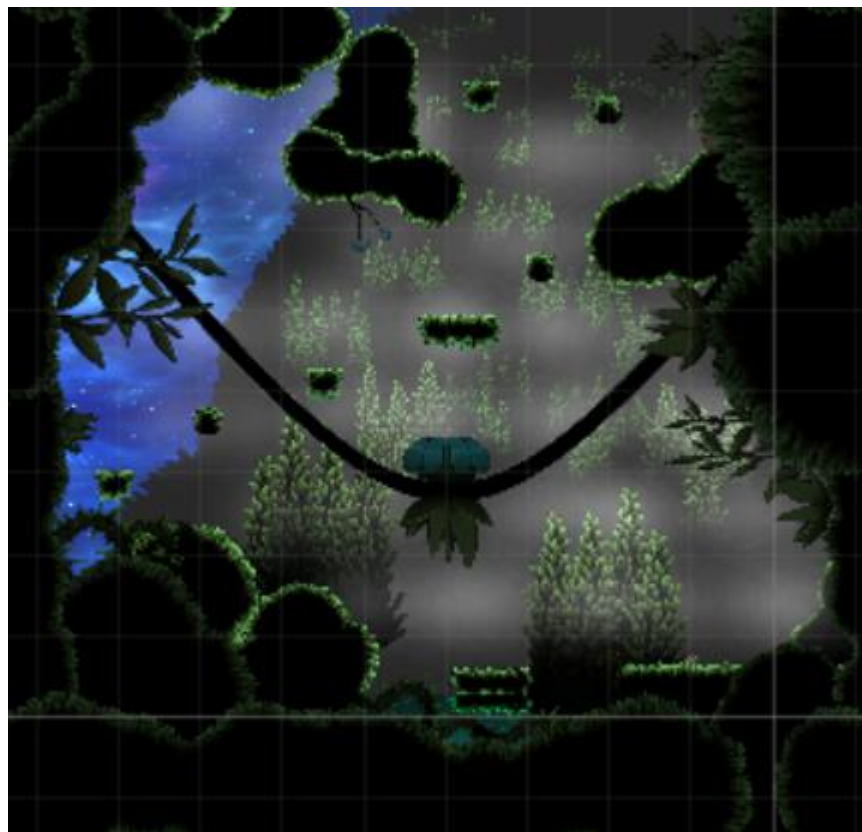
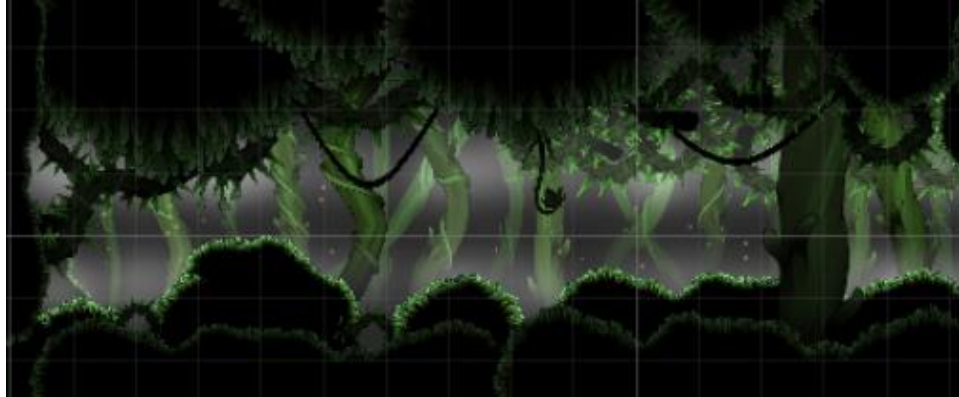


Imagen 13. Ambiente montaña

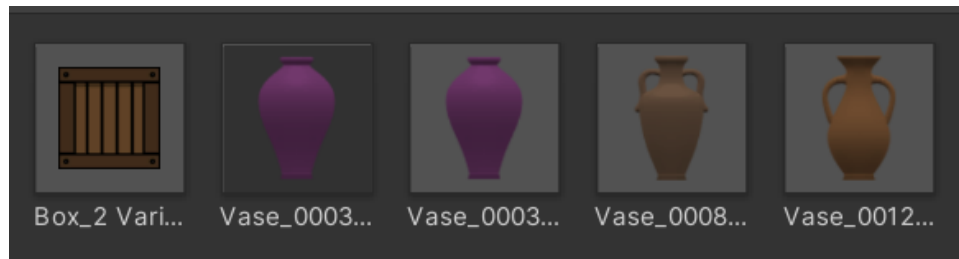




**Imagen 14. Ambiente selva**

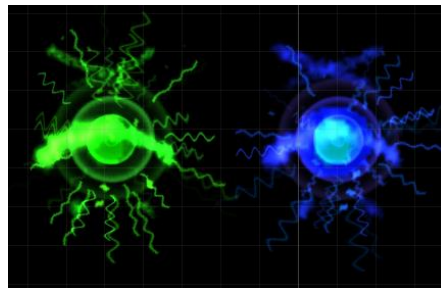
Los ambientes son inspirados en elementos de la selva colombiana del Chocó, con un estilo animado y elementos que surgen de la creatividad de los diseñadores y productores de este videojuego. Los escenarios proponen un ambiente inmersivo que apoya la narrativa del juego y presenta paisajes que generan diversas sensaciones, aportando al concepto: un personaje adentrándose en una selva desconocida plagada de enemigos y diversos retos.

- **Objetos:**



**Imagen 15. Objetos destruibles**

Objetos destruibles y contenedores de elementos de vida y poder para el Personaje Principal que interactúan con él mediante el uso del arma principal.



**Imagen 16. Poder y vida**

Objetos que brindan al Personaje Principal aditamentos de vida y poder respectivamente.



Imagen 17. Santuario

Objeto sagrado vinculado al concepto de pureza y poder del templo protector incluido en la historia del juego.

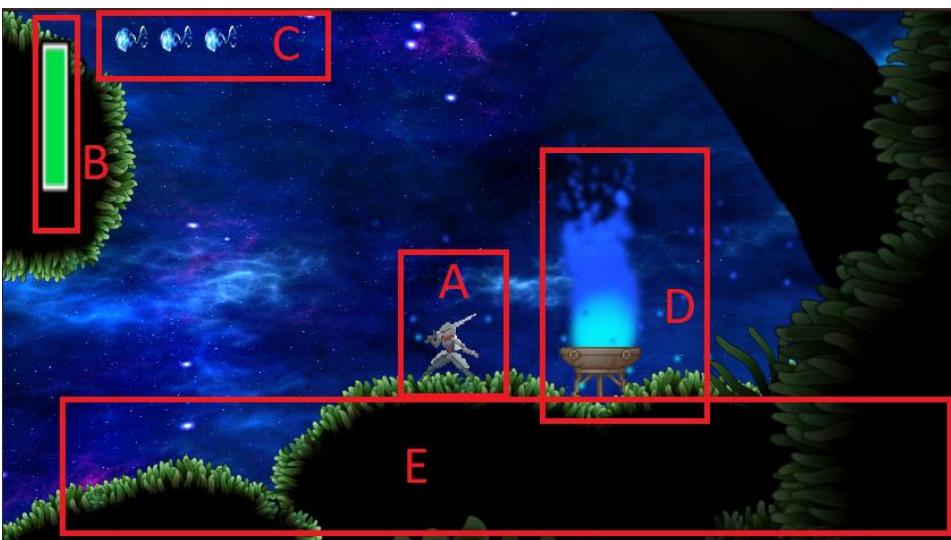


Imagen 18. Virgen guía

Objeto guía alusivo a una "Virgen" que, en la cultura colombiana puede relacionarse con una entidad que orienta con su sabiduría a quien la escucha.

## 8. INTERFAZ DE USUARIO

---



- A. Personaje principal.
- B. Barra de vida.
- C. Indicador de poder.
- D. Objeto interactivo.
- E. Escenario.

Imagen 19. Elementos de interfaz

## 9. PRUEBAS Y FUNCIONAMIENTO

V (add10) fue desarrollado y probado en una computadora gama alta con las siguientes especificaciones:

- MainBoard: ASUS ROG Crosshair VIII Hero
- Procesador: Ryzen 7 3800XT
- Memoria RAM: 32 GB/ 3800 MHZ
- Tarjeta Gráfica: NVIDIA RTX 2060 ASUS

Por esta razón, el videojuego está mejor optimizado en pc que cumplan con estas características. Sin embargo, se han realizado pruebas de rendimiento en diferentes computadoras de las gamas media y baja.

Se realizó un análisis sobre el rendimiento del videojuego en cada gama y con los resultados obtenidos, se realizaron ajustes para lograr una mejor optimización general. Actualmente V (add10) tiene el siguiente comportamiento:

- **Gama baja (Intel Celeron):** En algunos momentos presenta latencia en el reconocimiento de comandos y sufre de caídas en la tasa de FPS (cuadros por segundo). Aunque no sea recomendable ejecutar V (add10) en computadoras con estas características, es posible correr el programa y completar el videojuego.

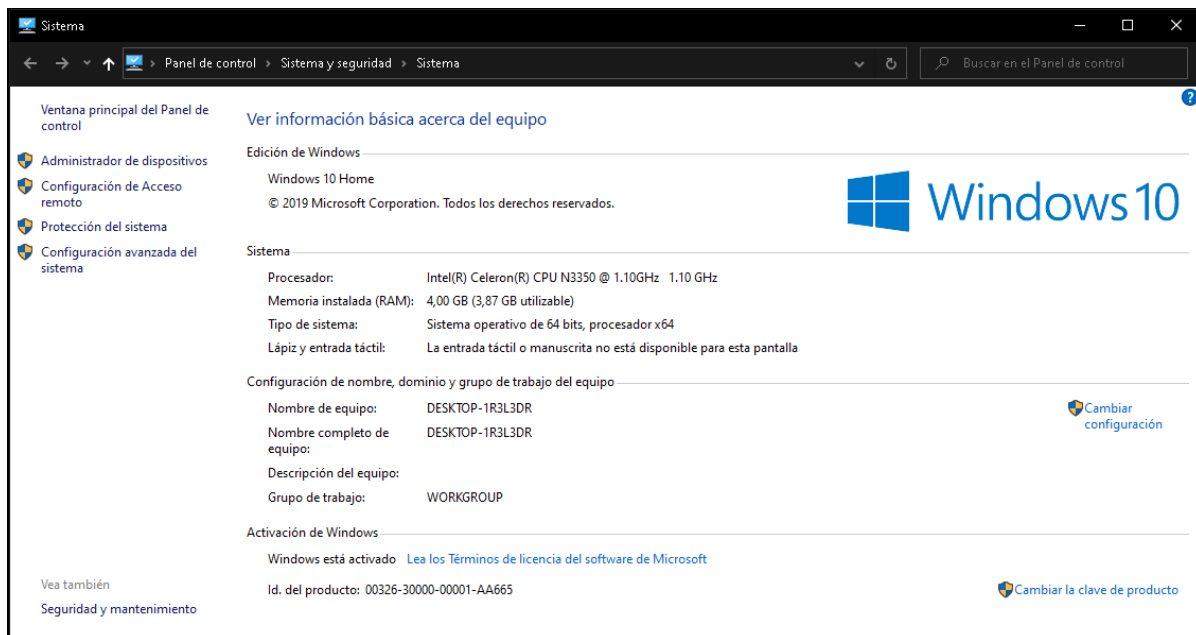


Imagen 20. Características computador gama baja

- **Gama media (Ryzen 7 3700U):**

Sistema

Procesador:	AMD Ryzen 7 3700U with Radeon Vega Mobile Gfx 2.30 GHz
Memoria instalada (RAM):	8,00 GB (5,94 GB utilizable)
Tipo de sistema:	Sistema operativo de 64 bits, procesador x64
Lápiz y entrada táctil:	La entrada táctil o manuscrita no está disponible para esta pantalla



[Información de soporte técnico](#)

**Imagen 21. Características computador gama media V (add10) funciona con *glitches* ocasionales menores.**

- **Gama alta (Ryzen 7 3800XT):**

<b>About</b>		
<b>Device specifications</b>		
Device name	DESKTOP-VU4DHUL	
Processor	AMD Ryzen 7 3800XT 8-Core Processor	3.89 GHz
Installed RAM	32.0 GB	
Device ID	6ECA033C-5719-47E7-A419-F375D6E4880E	
Product ID	00326-30000-00001-AA748	
System type	64-bit operating system, x64-based processor	
Pen and touch	No pen or touch input is available for this display	

**Imagen 22. Características computador gama alta**  
El videojuego se desempeña correctamente en computadoras que cumplan con estas características o similares.

El puzzle sonoro “Bringer’s gatekeep” fue sometido a prueba con un grupo de personas que en su mayoría no tenían entrenamiento auditivo o musical, la siguiente gráfica representa el porcentaje de los individuos que lograron resolverlo y el porcentaje de los que no lo lograron.

¿Posees conocimientos, formación o habilidades musicales?

10 respuestas

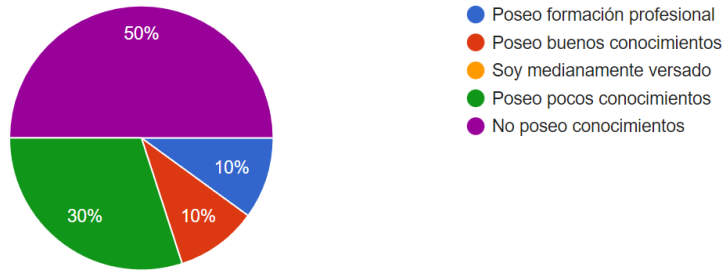


Imagen 23. Indicadores personal de prueba

¿Cómo calificarías la dificultad del Puzzle conforme al tiempo empleado para resolverlo?

10 respuestas

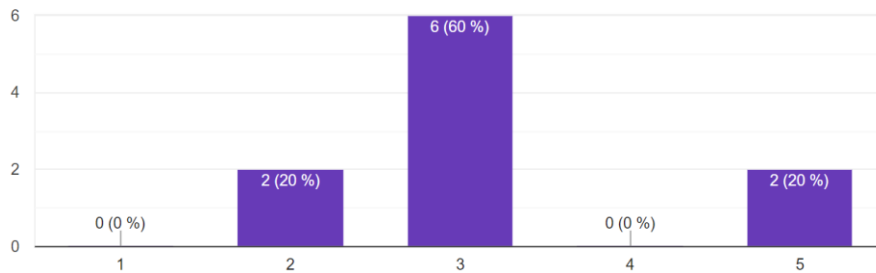


Imagen 24. Indicadores personal de prueba

**Nota:** siendo 1 el nivel de dificultad mínimo y 5 el nivel de dificultad máximo.

¿Cuánto tiempo tardaste en resolver el Puzzle?

10 respuestas

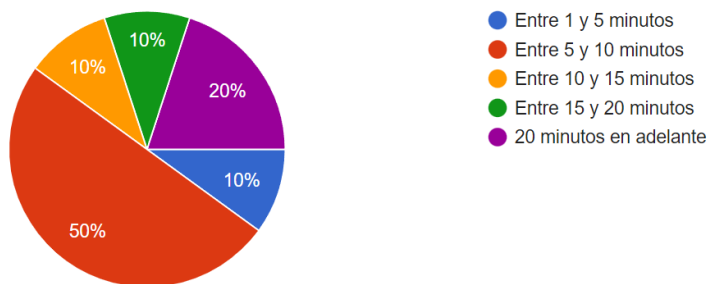


Imagen 25. Indicadores personal de prueba

Con base en los resultados de la gráfica, se puede concluir que el puzzle posee un nivel de dificultad suficiente para ser resuelto por personas con entrenamiento auditivo y personas sin entrenamiento auditivo.

## 10. GLOSARIO

---

- DAW (Digital Audio Workstation): Sistema integrado de hardware y software que permite al usuario grabar, editar y mezclar audio.
- FMOD: Motor de audio que permite crear eventos sonoros que se vinculan directamente con plataformas de desarrollo interactivo como Unity y Unreal.
- Unity: Plataforma de desarrollo interactivo y visual para productos como videojuegos y software dinámico.
- Puzzle: Acertijo o prueba a la que se somete el jugador como parte de los eventos interactivos del juego.
- Recolectables: Objetos a manera de recursos para el jugador y que pueden ser recolectados para uso inmediato o posterior.
- Roomtone: Sonido que representa la base tímbrica característica de un espacio acústico.

## 11. REFERENCIAS

---

- Acosta, A. M. (2016). *Plan de negocio para la creación de una empresa desarrolladora de software*. Bogotá: Universidad EAN.
- Casallas, J. F., & Riveros, Ó. C. (2019). *Creación de videojuegos en la enseñanza del uso placas de desarrollo de hardware para ingenieros de sistemas*. Ibagué: Universidad Cooperativa de Colombia.
- Collins, K. (2008). *Game Sound: An Introduction to the History, Theory, and Practice of Video Game Music and Sound Design*. London: The mit press.
- D'Aoust, K. (2014). *Unity Game Development Scripting*. Birmingham: Packt publishing.
- Delmas, G., Champagnat, R., & Augeraud, M. (2020). *From Tabletop RPG to Interactive Storytelling: Definition of a Story Manager for Videogames*. Montreal.
- Egea Canales, J. (2020). *Desarrollo De Un Videojuego Para Móviles Con Unity O Cocos2d-X*. Alicante: Escuela politécnica superior.
- Gáfaró, R. E., Contreras, M. R., & Villamizar, L. A. (2017). *Gestión de alcance en proyectos de desarrollo de videojuegos*. Universidad de Pamplona.
- González, J., & Obando, O. L. (2011). *Clasificar los videojuegos como tarea dinámica*.
- González, R. M. (2015). *Doblaje y videojuegos: La incidencia de la industria del cine en un nuevo sector de ocio*. San Vicente del Raspeig: Universitat d'Alacant.

- Green, A. M. (2018). *Storytelling in Video Games: The Art of the Digital Narrative*. Matthew Wilhelm Kapell.
- Horowitz, S., & Looney, S. R. (2014). *The Essential Guide to Game Audio: The Theory and Practice of Sound for Games*. New York: Routledge.
- Ibar, E. B. (2018). *El sonido como herramienta narrativa en los videojuegos*. Cataluña: Universitat Politècnica de Catalunya.
- Iwata, S., Kondo, K., Yokota, M., & Osawa, T. (17 de 06 de 2011). Iwata pregunta: The legend of Zelda: Ocarina of Time 3D. (S. Iwata, Entrevistador)
- Lebowitz, J., & Klug, C. (2011). *Interactive storytelling for video games: A player-centered approach to creating memorable characters and stories*. Taylor & Francis.
- Marks, A. (2009). *The complete guide to game audio: for composers, musicians, sound designers, and game developers*. Elsevier.
- Marks, T. (20 de 04 de 2020). *ing.com*. Obtenido de *ing.com*: <https://www.ign.com/articles/2018/06/22/hollow-knight-review>
- Mukherjee, S. (2015). *Video games and storytelling: Reading games and playing books*. London: Palgrave Macmillan.
- Pastor, E. M., & Pazos, A. J. (2020). *Los videojuegos como contenido de la telefonía*. Murcia: Universidad Católica San Antonio.
- Perron, B. (2009). *Horror Video Games: Essays on the Fusion of Fear and Play*. McFarland & Co.
- Planells de la Maza, A. J. (2010). *La evolución narrativa en los videojuegos de aventuras (1975-1998)*.
- Porras Orozco, J. (2020). *Diseño sonoro para videojuegos*. Bogotá.
- Sánchez, J. L. (2010). *Jugabilidad. Caracterización de la experiencia del jugador en videojuegos*. Madrid: Universidad de Granada.
- Schilling, C. (05 de 05 de 2018). *pcgamer.com*. Obtenido de *pcgamer.com*: <https://www.pcgamer.com/the-making-of-undertale/>
- Ulaş, E. S. (2014). *Virtual environment design and storytelling in video games*. Intellect.
- Villagrà Arnedo, C. J., Gallego Durán, F. J., Molina Carmona, R., & Llorens Largo, F. (2014). *ABPgame+: siete asignaturas, un proyecto*. Alicante: Universidad de Alicante.
- Vives, J. Á. (2018). *Creando universos: proceso para la realización de Concept Art para videojuegos del tipo plataformer en el Perú*. Lima.
- Zamora, V., & Darío, R. (2016). *Los videojuegos, nuevas narrativas del siglo XXI*. Madrid.
- Zehnder, S. M. (2006). *The Role of Music in Video Games*. París.