

# Propuesta del Proyecto final

Julián Mauricio Sánchez Ceballos, Mateo Restrepo Mesa

## 1 Sinopsis

Nuestro videojuego se llama: DAS LAND DES SITTICH.

El juego nos introduce a dos personajes principales Eliodoro y Joaquin, los cuales son unos cantantes que quieren alcanzar la fama, no a nivel mundial si no a nivel interestelar. Ellos son un dúo inseparable y tienen como misión conseguir el objeto de la antigua tribu Los Illuminati-brr, el objeto es un amuleto llamado “Sittich” y esté tiene una magia especial que consiste en teletransportar a una nueva dimensión a los seres que toquen el amuleto y dotarlos de habilidades excepcionales en el canto. Lastimosamente su travesía no será tan sencilla debido a que el amuleto está protegido por guardianes antiguos, los tomboys son unos fornidos guerreros con grandes poderes. Los personajes empezaran en la tierra y si logran conseguir el amuleto terrenal serán teletransportados al mítico y lejano lugar llamado “das Land des Sittichs” el lugar donde está resguardado el amuleto que les dará la las habilidades de canto necesarias para convertirse en los cantantes más famosos de todo el universo

## 2 Mecanismos de Juego

El inicio en la pantalla además del título del juego y un fondo habrá un menú donde pueda escoger factores antes del inicio de la partida como el modo de dificultad y el número de jugadores así como también podrá elegir cargar su partida. El planteamiento inicial de este apartado se propone respecto a nociones y manejo de variables simples por medio de interfaz gráfica.

Gráficamente la propuesta es un juego ambientado en imágenes en 8bits y su jugabilidad será de un plataformero basado en recorrer una pantalla con varios obstáculos y saltos, a medida de que avanza en la pantalla ya sea hacia arriba o a los lados los personajes tendrán que recoger objetos que les proporcionará puntos y vencer enemigos que irán apareciendo en el mapa. Dentro de la pantalla de juego habrá una barra de vida y un contador de puntaje que, a medida de que el o los jugadores avanzan en el nivel, irá cambiando respecto a las acciones del jugador.

El jugador podrá hacer el movimiento de su respectivo personaje con las teclas: A, D para hacer los movimientos de izquierda a derecha respectivamente y de la tecla W para saltar, al mismo tiempo podrá combinar las teclas de dirección con la tecla de salto para hacer un salto dirigido describiendo así una

especie de tiro parabólico, por último respecto a los movimientos y acciones que pueda realizar el jugador tenemos la tecla de ataque que será la tecla E.

Para el multijugador se realizará el mismo procedimiento con diferente asignación de teclas para los respectivos movimientos. Esta elección se hará en la pantalla de inicio, donde se decidirá si hacer o no un segundo objeto personaje.

Los personajes contarán con la implementación física de movimientos como: el movimiento uniformemente acelerado para sus movimientos, el movimiento parabólico para sus saltos y el movimiento de caída libre para sus caídas de la plataformas repartidas por el mapa.

El juego tendrá 2 dificultades, el nivel normal y el nivel leyenda vallenata, en el normal cada persona tendrá 3 vidas en donde si pierden las 3 vidas tendrán que repetir el nivel desde cero o si ya han completado el primer nivel, tendrán que repetir desde el segundo nivel.

## **3 Clases del videojuego**

### **3.1 class Personajes**

Es la clase donde se implementaran los métodos de los personajes principales del juego, los movimientos físicos a implementar en esta clase son: MRUA, movimiento parabólico y caída libre.

Además de esto entre varias características que tendrá los personajes será: su posición en x, su posición en y tamaño y velocidad de movimiento. Algunos de los métodos van a ser: Vida, moveUp(movimiento hacia arriba) moveDown(movimiento hacia abajo) moveLeft(movimiento a la izquierda), moveRight(movimiento a la derecha) paint y boundingRect, dos metodos que permitan a la clase dibujarse en la escena.

### **3.2 Class Enemigos**

En esta clase se implementan los movimientos de los enemigos junto con su inteligencia artificial y su vida, generándolos de manera aleatoria en el mapa. la implementación de esta clase dentro del juego se hace mediante la evaluación las colisiones entre muros, los personajes jugables a quienes les restará vida con esta colisión o disparos del jugador el cual resta vida (una de las características del objeto).

algunas de las características a incluir en este objeto serán la velocidad la posición en x y la posición en y, vida y sus métodos principalmente se basaran en la evaluación de colisiones el movimiento periódico en areas distribuidas en el mapa

### **3.3 Class disparos**

Una clase dispuesta para los disparos que realiza los personajes, equipado con una implementación física del movimiento de tiro parabólico, el objeto disparo

se creará con un evento de tecla y basándose en las coordenadas que lleve el personaje en el momento en el que se presiona la tecla de disparo.

De la características principales de este objeto se destaca el daño la velocidad, las coordenadas en x,y y en sus características la actualización e implementación de la físicas, además de la herencia de `QGraphicsItem` que lo pueda convertir en un objeto graficable.

### **3.4 Class barra de vida**

Una clase destinada a graficar el porcentaje de vida de el personaje por lo que cada personaje tendrá un objeto tipo barra de vida que determinará la vida del personaje.

### **3.5 Class Puntaje**

una clase que llevara el contador de los puntos que podrá heredar cada personaje para llevar el puntaje de cada personaje. Evaluando colisiones con monedas y las colisiones de los disparos contra los enemigos.

### **3.6 Class monedas**

Una clase con métodos gráficos para dibujar monedas en el mapa. las monedas. La implementación de esta clase se hará por medio de una lista donde se tendrá las coordenadas de cada moneda para posteriormente ser añadida a la escena. Habrá una función específica que leerá un archivo de texto para cada coordenada de las monedas y objetos necesarios en el juego de cada nivel.

Con algunos metodos como el `paint` y el `boundingRect` que permitiran graficar dentro de la escena los objetos tipo monedas que se generen la lista.

### **3.7 Class Plataformas**

Una clase con métodos gráficos que permitirá crear y dibujar plataformas en la escena. La implementación de esta clase se hara por medio de una lista donde se tendrá las coordenadas de cada plataforma para posteriormente ser añadida a la escena. Habrá una función específica que leerá un archivo de texto para cada coordenada de las plataformas de cada nivel.

Con algunos metodos como el `paint` y el `boundingRect` que permitiran graficar dentro de la escena los objetos tipo plataformas que se generen la lista.

## **4 Cronograma**

el objetivo del cronograma es programar un plan de desarrollo estructurado para las semanas restantes hasta la fecha de entrega previamente por el profesor, el plan contiene inicio y duración de los cada bloque de trabajo para un total de seis bloques de trabajo desde la fecha 18 de diciembre del 2020 al 24 de enero del 2020, donde la etapa de producción del proyecto se debe llevar acabo en su

totalidad incluyendo la etapa de pruebas, realización de trailer y vídeo. Cada bloque indica el trabajo a realizarse.

#### **4.1 Bloque 1 (18 de diciembre al 25 de diciembre)**

Entrega de la propuesta del juego.

Elecciones de las imágenes necesarias en la escena y de los personajes.

Primer bloque de trabajo enfocado a la realización e implementación de los objetos enemigos, personajes y disparos a la par de la implementación de sistemas físicos correspondientes a cada objeto.

#### **4.2 Bloque 2 (25 de diciembre al 1 de enero)**

segundo bloque de trabajo enfocado a la realización de los objetos enemigos, personaje principal, segundo personaje, disparos, vidas, y partidas. Cada personaje y objeto con sus respectivas físicas y actualización de coordenadas (movimientos).

Desarrollo de las nociones de puntaje multijugador y dificultad.

#### **4.3 Bloque 3 (1 de enero al 9 de enero)**

Desarrollo del apartado Administrador de partidas: Nueva partida, cargar partida, guardar partida, eliminar partida, reiniciar partida y niveles.

Primer bloque de trabajo enfocado a la interfaz gráfica: Escenario, plataformas, monedas.

#### **4.4 Bloque 4 (9 de enero al 16 de enero)**

Corrección de las colisiones necesarias acerca de los personajes con los objetos de la escena.

Segundo bloque de trabajo enfocado a la interfaz gráfica: elección de las imágenes de los objetos enemigos, plataformas de la escena e implementación de los sprites.

correcciones pertinentes antes de la entrega 1.

#### **4.5 Bloque 5 (16 de enero al 23 de enero)**

Entrega 1.

Realización del vídeo y el tráiler.

Correcciones para la entrega 2.

#### **4.6 Bloque 6 (13 de enero al 30 de enero)**

Pruebas, correcciones finales y entrega 2, entrega final del proyecto, edición del trailer.