

Diseño

Proyecto Final

Mariana Noreña Vásquez
Santiago Vélez Arboleda

Departamento de Ingeniería Electrónica y
Telecomunicaciones
Universidad de Antioquia
Medellín
Octubre de 2021

Índice

1. Diseño	2
2. Cronograma	5

1. Diseño

A continuación, se presentan las clases y objetos que estructuran a GEHRIN.

1. Levels: Contiene la distribución de los diferentes niveles.
2. Plataform: Esta clase representa las plataformas de cada nivel. Como atributos, tiene las posiciones en x,y, el ancho y alto de la plataforma. Así mismo, contará con los siguientes métodos:
 - a) El constructor el cual recibe las dimensiones de las plataformas junto a su posición en el escenario.
 - b) BoundingRect y paint: Dibuja la forma de plataforma para corroborar su posición. La clase tiene los métodos get de los atributos que posee.
3. Spike: Esta clase es la encargada de los pinchos que estarán en el segundo nivel. De forma similar a Plataform, recibe como atributos las posiciones en x,y, el ancho y alto de la plataforma. Ahora, cuando el jugador interactúe con él, se le restará una vida o morirá en su defecto. También contará con los métodos anteriormente mencionados.
4. Player: La clase representará a ambos personajes que componen el juego (The guy y Steven). Está compuesto por los atributos:
 - a) Sprites
 - b) Timers para la muerte, ataque, salto y spown en el escenario.
 - c) Velocidades, aceleraciones y posiciones en el sistema corrdenado x,y
 - d) Booleanos para el movimiento del personaje.
 - e) Vida del personaje y daño que genera al interactuar con un enemigo.

De igual forma, contará con los siguientes métodos:

- a) Constructor de la clase: Este recibe un carácter el cual permite identificar que personaje se usará en el juego. Inicializa los booleanos, velocidades y aceleraciones previamente mencionados.
- b) WalkGuy: Esta función hace referencia a los Sprites del movimiento de The Guy. Para el caso en el que esté saltando, moviéndose a lo largo del escenario o atacando a un enemigo.
- c) WalkSteven: Esta función hace referencia a los Sprites del movimiento de Steven. Para el caso en el que esté saltando, moviéndose a lo largo del escenario o atacando a un enemigo.
- d) Cinemática: Esta función se encargará del salto del personaje con un movimiento parabólico en Y y un movimiento rectilíneo en X.
- e) Jump: Esta función se encarga de la gestión de los sprites del personaje cuando esté saltando. Hará un intercambio de los mismos cuando esté subiendo o bajando en Y.

- f)* updateSpown: Se encarga de hacer el Spown de The Guy.
 - g)* updateDeath: Se encarga de mostrar los sprites para animar la muerte de los personajes.
 - h)* updateAttack: Muestra la animación del ataque de los personajes
5. GuyAttack: Esta clase es la encargada del ataque de los personajes. Como atributos, se tendrán:
- a)* Sprites del ataque de TheGuy y Steven.
 - b)* Timer
 - c)* Variable para el rango de ataque
 - d)* Posición en x,y del ataque
 - e)* Velocidad
 - f)* Dos caracteres para la dirección del ataque y a que personaje pertenece el ataque.

Métodos:

- a)* Constructor: Recibe las posiciones, la dirección del ataque y personaje. Inicializa las posiciones del ataque a la mitad del cuerpo del personaje y conecta el timer para los sprites.
 - b)* Sprites: Dependiendo del rango y del personaje, se mostrará en escena los diferentes sprites del ataque.
6. Enemigos: Representa a dos de los enemigos del juego (Huesos y hammer). Como atributos:
- a)* Sprites
 - b)* Timers para el movimiento, ataque, muerte, victoria y Spown de Huesos
 - c)* Variable para la vida, constante de restitución, posiciones, velocidad, rango, booleanos para el estado en que se encuentra y dirección.

Se tendrán los siguientes métodos:

- a)* Constructor: Recibe los parámetros de posiciones en x,y. Inicializa los atributos de posición y de estado.
 - b)* Ataque: Construye el timer de ataque y lo conecta con el slot HuesosAttack.
 - c)* Mover: Construye el timer de Move y lo conecta con el slot de Movimiento.
 - d)* Idle: Inicializa el timer y lo conecta con el slot de HuesosIdle.
7. Slots:

- a) HuesosAttack: Se escogerá de manera aleatoria la animación del ataque y dependerá de la dirección en la que se mueva el enemigo.
 - b) HuesosIdle: se define de manera aleatoria la dirección en la que se orienta en la escena.
 - c) HuesosIdle: se define de manera aleatoria la dirección en la que se orienta en la escena.
 - d) HammerAttack: Se presentará la animación de ataque de Hammer. Así mismo, dependerá de la dirección en la que se mueva el enemigo. Para este ataque se usará un tiro parabólico.
 - e) HuesosWin: Esta función representa los sprites en escena cuando el jugador muera.
 - f) HuesosDeath: Esta función representa los sprites de huesos cuando el jugador lo elimina
 - g) Métodos get y set para los atributos
8. HAttack: Esta clase representa el ataque del enemigo Hammer. Como atributos recibe:
- a) Sprite del martillo.
 - b) Timer
 - c) Carácter para la dirección del ataque.
 - d) Variables para posición, velocidad y rango.

Como métodos, se tendrá:

- a) Constructor
 - b) Slot de animacion de martillo
 - c) Métodos get y set para los atributos
9. Object: Esta clase tendrá las propiedades de los objetos. Tiene como atributos:
- a) Constante de restitución
 - b) Velocidad

Como métodos:

- a) Constructor que recibe un entero
 - b) Rebote y colisiones
 - c) Tiro parabólico.
10. Bills: Esta clase contendrá todas las propiedades del enemigo final del primer nivel Bills. Como atributos tendrá:
- a) Sprites para ataque, muerte, spown.

- b)* Tendrá timers para el ataque, muerte, movimiento y spown.
- c)* Variables para la posición, rango de ataque, velocidad y vida.

Métodos:

- a)* Constructor.
- b)* Método para el ataque
- c)* Método para la muerte
- d)* Spown.

11. BillsAttack: Esta clase contendrá las propiedades para el ataque de Bills. Como atributos tendrá:

- a)* Sprites para el ataque.
- b)* Posiciones.
- c)* Timers.
- d)* Velocidad
- e)* Daño

Tendrá los siguientes métodos:

- a)* Constructor
- b)* Slot para el movimiento del ataque

12. Dj Sray: Esta clase contendrá las propiedades del enemigo final. Como atributos tendrá:

- a)* Lista con los movimientos de baile.
- b)* Sprites.
- c)* Timer

Tendrá como métodos:

- a)* Validar movimientos.
- b)* Constructor de la clase.

2. Cronograma

A continuación, se presenta el cronograma para el proyecto:

	Semana 1					Semana 2					Semana 3				
	L	M	W	J	V	L	M	W	J	V	L	M	W	J	V
Diseño Niveles	X	X	X		X										
Jugadores: Movimiento- Sprites	X		X		X		X		X						
Enemigo: Huesos		X		X		X									
Enemigo: Hammer		X		X		X									
Enemigo: Bills			X		X		X								
Enemigo: Dj Sray						X	X	X	X						
Colisión Jugador- Plataforma	X		X		X		X		X						
Colisión: Enemigo-Jugador Enemigo-AtaqueJ		X	X	X	X	X	X								
Almacenamiento Datos						X	X		X						
Interacción Usuario-Interfaz		X			X		X	X		X	X	X			
Pruebas	X	X	X	X	X	X	X	X	X	X	X	X	X	X	

Figura 1: Captura