



# **UNIVERSIDAD DE ANTIOQUIA**

**Facultad de Ingeniería**

**Modelacion de las clases**

**Jeronimo Herrera Sanchez**

**Juan Camilo Sierra Gomez**

**Informatica 2**

**Profesores: Augusto Enrique Salazar Jimenez , Jonathan  
Ferney Gomez Hurtado**

5 de abril de 2022

## Intrroducción

Principalmente el juego esta basado en una estructura 2D donde podemos encontrar el personaje principal y algunas estructuras que le faciliten la supervivencia en el juego:

- Estructuras defensivas
- Estructuras de ofensiva
- Estructuras de mejora
- Otro tipo de estructura

El escenario cambiara dependiendo de en que oleada se encuentre el jugador de forma que haga el ciclo del día y la noche. Dependiendo del avance del día las oleadas se tornarán más difíciles para el jugador. Al inicio de su supervivencia el jugador se podrá encontrar con pocos enemigos y no muy peligrosos, si el jugador logra avanzar se encontrará con muchos mas enemigos y más difíciles:

- Enemigos normales: estos enemigos aparecen al inicio de la supervivencia no son muy peligrosos ya que no poseen mucha vida, poca velocidad y generan un bajo daño al personaje.

- Enemigo tanque: estos enemigos aparecerán después de ciertas oleadas y aumentaran la dificultad del juego ya que poseen mas vida y generan mas daño al personaje, pero menos velocidad.

- Enemigo rango: estos enemigos serán los últimos en aparecer y complicaran bastante el juego ya que este enemigo lanzara objetos tipo pócima y estos provocaran la máxima cantidad de daño al personaje. En el escenario habrá dos máquinas las cuales le permitan mejorar al jugador des distintas formas

En el escenario habrá dos máquinas las cuales le permitan mejorar al jugador des distintas formas

- Maquina de compra: esta máquina le permitirá comprar al jugador munición y unas habilidades especiales que podrá activar cuando el desee algunas de ellas se acabarán en un margen de tiempo, pero todas las habilidades tendrán que ser compradas después de gastarlas

- Maquina de suerte: esta maquina tomara tu suerte para obtener distintas cosas como, por ejemplo (objetos de vida, objetos de mejora, enemigos y dinero)

Habrà un minijuego después de cierta cantidad de oleades, el minijuego consistirá en unas rocas que se deslizaran desde el lado derecho del escenario hasta el izquierdo y el jugador con la ayuda de unos trampolines para esquivar las rocas

## Clases

### 1. Clase: personaje

Esta clase nos representan el personaje principal el cual posee atributos que le permiten almacenar dinero, municiones y tener una velocidad predeterminada la cual puede variar dependiendo de las habilidades que posea

#### Atributos

Velocidad: Esta será la velocidad la cual tendrá el personaje al inicio del juego, Esta podrá variar dependiendo si logra conseguir un accesorio llamado botas.

Munición: Será la munición inicial que tendrá el jugador al iniciar el juego.

Dinero: un contador con el cual el jugador vaya guardando el dinero que consigue por matar a un enemigo.

#### Constructor

Setdinero: cambiara el dinero que posea el jugador.

Getvelocidad: dará la velocidad del personaje para la utilización en varias partes del programa como, por ejemplo, los movimientos físicos.

Setvelocidad: esta recibirá el cambio en la velocidad al momento de adquirir las habilidades

Getmunición: retornara la cantidad de munición que posee el personaje

Setmunición: recibirá y modificará la cantidad de munición que tenga el personaje

### 2. Clase: armas

Esta clase nos representa el proyectil de las armas que posea el jugador la cual posee un daño fijo al inicio pero puede aumentar con la cantidad de enemigos que elimine con esta arma y un tiempo de recarga que varia dependiendo el tipo de arma

### **Atributos**

Daño: Será el daño que haga cada tipo de arma al enemigo

Recarga: Dependiendo del tipo de arma recargara cada numero de balas del arma

Mejora: Este atributo aumentara el daño del arma dependiendo de la cantidad de enemigos que mate con la misma

### **Constructor**

getDaño: retornara la cantidad de daño que hará el arma que posea en el momento con su nivel de mejora

setDaño: cambiara la cantidad de daño de cada tipo de arma y de su nivel de mejora

getRecarga: retorna el tiempo que se demora en recarga las armas

## **3. Clase: enemigos**

Esta clase nos representa el aumento de los atributos físicos de los enemigos

### **Atributos**

Daño: El daño que infligirá los enemigos dependiendo de en qué oleada este el jugador

Velocidad: La velocidad que tendrá los enemigos dependiendo de la oleada que se encuentre en cada jugador

### **Constructor**

GetDaño: retorna la cantidad de daño que inflige el enemigo al golpear al jugador

SetDaño: cambia la cantidad de daño que inflige cada enemigo dependiendo de la oleada en la que este el jugador

GetVelocidad: retorna la velocidad que tendrá el enemigo en el momento

SetVelocidad: cambiara la velocidad de movimiento del enemigo dependiendo de la oleada en que se encuentre el jugador

## 4. Clase: Movimientos

En esta clase se encontrará todos los movimientos físicos, para la utilización en muchos casos como por ejemplo para el movimiento de la rueda de la fortuna que estará girando de forma circular

Mov\_parabolico: En cuanto al Mov\_parabolico se desarrolla en la clase proye\_parabolico

Gravedad: la gravedad de la tierra

Mov\_circular este movimiento se vera apreciado en la rueda de la fortuna

Mov\_osilatorio por ultimo el movimiento oscilatorio el cual se utilizará para un minijuego en cual se deslizarán unas rocas por el lado derecho de la pantalla y tendrá que utilizar unos trampolines para saltarlos

### Atributos

Posx: posicion en x

Posy: posicion en y

### Constructor

Se utilizan las funciones trigonométricas del coseno y del seno

Setpos(posx,posy): se va cambiado la posición en la ruda de la fortuna

## 5. Clase: vidas\_salud

Se encarga de la gestión de vidas del jugador y de la salud de todos los objetos

### Atributos

Vida: Sera la cantidad de vidas que posea el personaje, si llega a 0 se acabara el juego

Salud: la cantidad de salud que tendrá ya sea el jugador antes de perder una vida o el enemigo antes de morir

### Constructor

Getvida: retorna la cantidad de vidas que tiene el jugador

Decrease: disminuye la cantidad de vidas del jugador en 1 si, la salud del jugador llega a 0

Getsalud: la salud que tenga el objeto ya sea jugador, enemigo, estructura

Setsalud: una función que vaya variando, dependiendo del daño que le inflijan

## **6. Clase: enemigo\_normal**

Se encarga de los atributos físicos del enemigo normal

### **Atributos**

Salud: vida de los enemigos normales al comienzo del juego

Ganacia: dinero que da al morir el enemigo

Slot posición: es una función que no retorna nada que se llame movi y este se encarga del movimiento del enemigo el cual siempre ira desde x- a x+ y viceversa

### **Constructor**

Getsalud: retorne la cantidad de salud que tenga el enemigo en el momento

Getganacia: retorna a una cantidad fija a la clase personaje

Setsalud: cambia la cantidad de salud que tenga el enemigo

## **7. Clase: enemigo\_rango**

Se encarga de los atributos físicos del enemigo rango

### **Atributos**

Salud: vida de los enemigos normales al comienzo del juego

Ganacia: dinero que da al morir el enemigo

Slot posición: es una función que no retorna nada que se llame movi y este se en-

carga del movimiento del enemigo el cual siempre ira desde x- a x+ y viceversa

### **Constructor**

Getsalud: retorne la cantidad de salud que tenga el enemigo en el momento

Getganacia: retorna a una cantidad fija a la clase personaje

Setsalud: cambia la cantidad de salud que tenga el enemigo

## **8. Clase: enemigo\_tanque**

Se encarga de los atributos físicos del enemigo tanque

### **Atributos**

Salud: vida de los enemigos normales al comienzo del juego

Ganacia: dinero que da al morir el enemigo

Slot posición: es una función que no retorna nada que se llame movi y este se encarga del movimiento del enemigo el cual siempre ira desde x- a x+ y viceversa.

### **Constructor**

Getsalud: retorne la cantidad de salud que tenga el enemigo en el momento

Getganacia: retorna a una cantidad fija a la clase personaje

Setsalud: cambia la cantidad de salud que tenga el enemigo

## **9. Clase: Juego**

Se desarrolla todo el juego en esta clase

### **Atributos**

enum mode Menu, Pause, Play, guardar, cargar los modos que tendrá el juego

### **Constructor**

keyPressEvent: se utilizará para el movimiento del personaje el cual se podrá mover ya sea con las flechas del teclado o con las letras (w,a,s,d), y con la z se reparará los muros o los cañones que estarán colocados en una posición en específico. Con la x se podrá comprar en la tienda o girar la tragaperras para probar suerte

Personaje\* player: se le da el apuntador al objeto personaje

estructura\* Estruc: Se inicializa el objeto estructura

Pausar: se utilizará para pausar el juego cuando desee el jugador

Inicio: se utilizará para iniciar el juego

Guardar: Se utilizará para guardar partida después de pasar una oleada

Cargar: Se utilizará para cargar una partida previamente guardada

Qwidget: Se encarga de crear la escena

## **10. Clase: estructura**

Se le dan los atributos y costos de las estructuras defensivas y ofensivas

### **Atributos**

Posx: la posición de la estructura ya sea muro o el cañón en el eje x predeterminadas  
Costo: costo de reparar las estructuras

### **Constructor**

Salud\_estructura: recibe de la clase vida\_salud la función que retorna la vida que tiene el objeto y puede cambiar la salud que tiene el objeto

get\_dinero: función que retorna el costo de las estructuras

estructura: recibe la posición del jugador y coloca la estructura correspondiente



## 11. Clase: MainWindow

### Constructor

*Q\_OBJECT*

MainWindow(QWidget \*parent = 0);

MainWindow();

*Press\_inicio : el botón de inicio del juego*

*Press\_pause : el botón para pausar el juego*

*Press\_guardar : el botón para guardar el progreso del juego*

*Press\_cargar : el botón para cargar el progreso de un juego previo*

## 12. Clase: proye\_parabolico

Esta clase representa el proyectil el cual tiene el movimiento físico parabólico

### Atributos

Daño: el daño que infligirá la bola del cañón a los enemigos

Angulo: Angulo para el movimiento parabólico

velocidad: la velocidad para las pociones de los enemigos de rango y el cañón. Esta velocidad será utilizada en los movimientos

Posx: posición en x

Posy: posición en y

Velocidadx: velocidad en x

Velocidady: velocidad en y

r: radio

### **Constructor**

Getdaño: una función que retorne el daño que inflige el proyectil a los enemigos

Calcularposicion: calcula la posición del proyectil

Actualizarvelocidad: actualiza la velocidad del proyectil

Calcularvelocidad: calcula la velocidad del proyectil

Movimiento: utiliza r para darle un tamaño fijo

Movimiento: recibe la posición en x, posición en y, la velocidad y el angulo para hacer el movimiento

getPosx: retorna la posición en x

getPosy: retorna la posición en y

## **13. Clase: roca**

Modela el objeto roca que será utilizado para el minijuego

### **Atributos**

Posx: posición en x de la roca

Daño: daño que inflige la roca al golpear al personaje

### **Constructor**

Setposx: una función que va cambio la posición de las rocas desde +x hasta -x

Getdaño: retorna la cantidad de daño que inflige la roca al tocar al enemigo

## **14. Clase: habilidades**

Le permitirán al jugador obtener una gran velocidad, un aumento de salud con un costo

## **Atributos**

salud: habilidad que aumenta la vida del jugador

velocidad: habilidad que aumenta la velocidad del jugador

Costo: costo de las habilidades

## **Constructor**

get\_salud: le retorna el aumento a la función set\_salud de la clase salud\_vida

get\_velocidad: le retorna el aumento a la función set\_velocidad de la clase personaje