



INFORMATICA II
PROYECTO FINAL

CAPITULO 10, TEMPORADA 3 DE RICK Y MORTY

LUIS DAVID MUÑOZ
RAUL DAZA LIÑAN

DEPARTAMENTO DE INGENIERIA ELECTRONICA Y
TELECOMUNICACIONES
FACULTAD DE INGENIERIA
UNIVERSIDAD DE ANTIOQUIA
MEDELLIN 4/12/2023

CONTENIDO

- Descripción general del juego realizado.
- Clase inicio
- Clase héroe
 - Clase Rick
 - Clase Morty
- Clase enemigo
 - Clase alíen
 - Clase escolta
- Clase balas
- Clase obstáculo
- Clase jugando
- Diagrama de clases
- Conclusiones

DESCRIPCION DEL JUEGO

Basado en el desarrollo de el capitulo 10 de la temporada 3 de Rick y Morty se diseño un juego implementando lenguaje C++. Se diseñaron 2 ventanas y 3 niveles de juego alusivos a escenas presentadas en dicha serie animada seleccionada.

La ventana inicial tiene como objetivo interactuar con el usuario, esta interacción se basa en que el usuario por medio del mouse (clic) escoja la dificultad de juego que requiere y además de seleccionar el personaje principal el cual actuara como héroe en los 3 niveles de juego e iniciar a jugar.

La ventana siguiente que es la principal, tiene como objetivo llevar a cabo el desarrollo de los 3 niveles de juego representados en 3 escenas diferentes, en las cuales a medida que avanza de nivel, la dificultad va aumentando y se presentan diferentes físicas implementadas en los movimientos en el nivel 3 con respecto a los niveles 1 y 2.

CLASE INICIO

La clase de inicio es llamada start, esta clase se encarga de la ventana secundaria que es la que se muestra al iniciar el juego, esta ventana esta encargada de permitir al usuario escoger ciertos parámetros para iniciar a jugar de la mejor manera.

Esta ventana presenta un fondo relacionado a la serie animada, adicional tiene botones (PushButton) que representas señales las cuales serán interpretadas en el código de la implementación. Estos botones permiten seleccionar el modo de dificultad que se quiere para el juego, además de seleccionar el personaje que desea tener como héroe, y finalmente para iniciar a jugar se debe presionar el botón de start que permite cambiar a la siguiente ventana donde se desarrolla el entorno de juego.

Si ningún botón es clickeado, es decir si no se escoge ningún parámetro de dificultad ni de personaje de héroe, se tiene programado por defecto un valor para la dificultad y un personaje como héroe.



IMAGEN 1. Ventana de interacción con el usuario.

CLASE HEROE

La clase héroe tiene como objetivo desarrollar las interacciones y movimientos del personaje principal.

Los atributos del personaje como la velocidad, la cual se implementa para realizar los movimientos físicos, agacharse que se implementa para reducir el tamaño, saltar que se implementa para esquivar obstáculos aplicando la física de movimiento parabólico.

Los métodos permiten acceder a los atributos y definir señales o movimientos que afectan el comportamiento del personaje, métodos como obtener las posiciones en escena, realizar los movimientos en varias direcciones por medio del teclado, saltar y agacharse para esquivar obstáculos, acelerar para mover mas rápido y escalar el personaje en la escena.

Clase Rick

Esta clase se implementa para definir el héroe Rick, es decir se pinta en escena el personaje correspondiente a Rick y se hereda todos los movimientos e interacciones de la clase héroe, por lo tanto, esta es una clase que se compone de otra clase más general.

Clase Morty

Esta clase se implementa para definir el héroe Morty, es decir se pinta en escena el personaje correspondiente a Morty y se hereda todos los movimientos e interacciones de la clase héroe, por lo tanto, esta es una clase que se compone de otra clase más general.

CLASE ENEMIGO

La clase enemigo tiene como objetivo desarrollar las interacciones y movimientos de los enemigos de las diferentes escenas, principalmente de las escenas 1 y 2, correspondientes a los niveles 1 y 2 respectivamente.

El atributo que presentan los enemigos es su velocidad, la cual se implementa para realizar los respectivos movimientos de los enemigos.

Los métodos que se implementan en esta clase corresponden a obtener las posiciones en la escena y realizar los movimientos de los diferentes enemigos presentados.

Clase alien

Esta clase se implementa para definir el enemigo alien que corresponde al villano del primer nivel, es decir se pinta en escena el villano y se hereda todos los movimientos e interacciones de la clase enemigo, por lo tanto, esta es una clase que se compone de otra clase más general.

Clase escolta

Esta clase se implementa para definir el enemigo escolta que corresponde al villano del segundo nivel, es decir se pinta en escena el villano y se hereda todos los movimientos e interacciones de la clase enemigo, por lo tanto, esta es una clase que se compone de otra clase más general.

CLASE BALAS

Esta clase es la encargada de generar los proyectiles (balas) en forma de elipses o círculos que serán asignados tanto al personaje principal como al villano en los niveles 1 y 2, en el nivel 3, esta clase se encarga de generar los proyectiles que viajan en línea recta y en círculo.

Los atributos de esta clase los definen la velocidad, las dimensiones de tamaño como el alto y ancho, una variable booleana que se usa para realizar el movimiento circular, y dos variables flotantes que se utilizan para las ecuaciones paramétricas de movimiento.

Los métodos definen diferentes acciones como obtener las posiciones, velocidades y dimensiones de las balas, también funciones como mover las balas hacia una dirección especificada además de asignarles color a las elipses tanto las instanciadas para el héroe como para el villano.

CLASE OBSTACULO

Esta clase esta encargada del obstáculo del nivel 3 el cual será cargado en forma de una imagen que será añadida a escena, este obstáculo va a tener un movimiento rectilíneo en una dirección fija, ese obstáculo debe ser superado por el héroe saltando por encima de él, a medida que va transcurriendo el nivel irán apareciendo obstáculos de la misma clase.

Esta clase tiene como atributo la velocidad en el eje x con la cual se va a ir desplazando.

Los métodos se encargan de obtener la velocidad, y realizar el respectivo movimiento del obstáculo en la escena.

CLASE JUGANDO

Esta es la clase principal del juego, donde se desarrollan las diferentes instancias de las clases diseñadas y se implementa el desarrollo del juego con sus respectivos niveles, esta clase es la mainwindow y corresponde a la segunda y principal ventana de juego.

Los atributos de esta clase están los diferentes timers que se utilizan para ejecutar parámetros cada cierto tiempo, las escenas que corresponden a cada nivel de juego, variables de tipo flotante que se encargan de las posiciones en escena y el tamaño de en escena, crear las diferentes listas que se utilizaron con el fin de almacenar instancias de los diferentes objetos y así recorrer las listas y buscar interacciones para ser tratadas, además de 2 variables tipo entera que son las que se encargan de llevar el registro de vidas y puntuación del personaje con el fin de llevar a cabo los cambios de nivel del juego o en su defecto la finalización del mismo.

Los métodos tienen como objetivo realizar instancias de los diferentes objetos como los villanos, las balas tanto de jugador como villano, realizar los diferentes movimientos de los objetos que interactúan, establecer las acciones cuando el usuario interactúa con el teclado y el mouse, definir las interacciones entre objetos (colisiones) y las diferentes interpretaciones que se presentan, además de llevar registro de los puntos y la vida del personaje en cada nivel.

DIAGRAMA DE CLASES

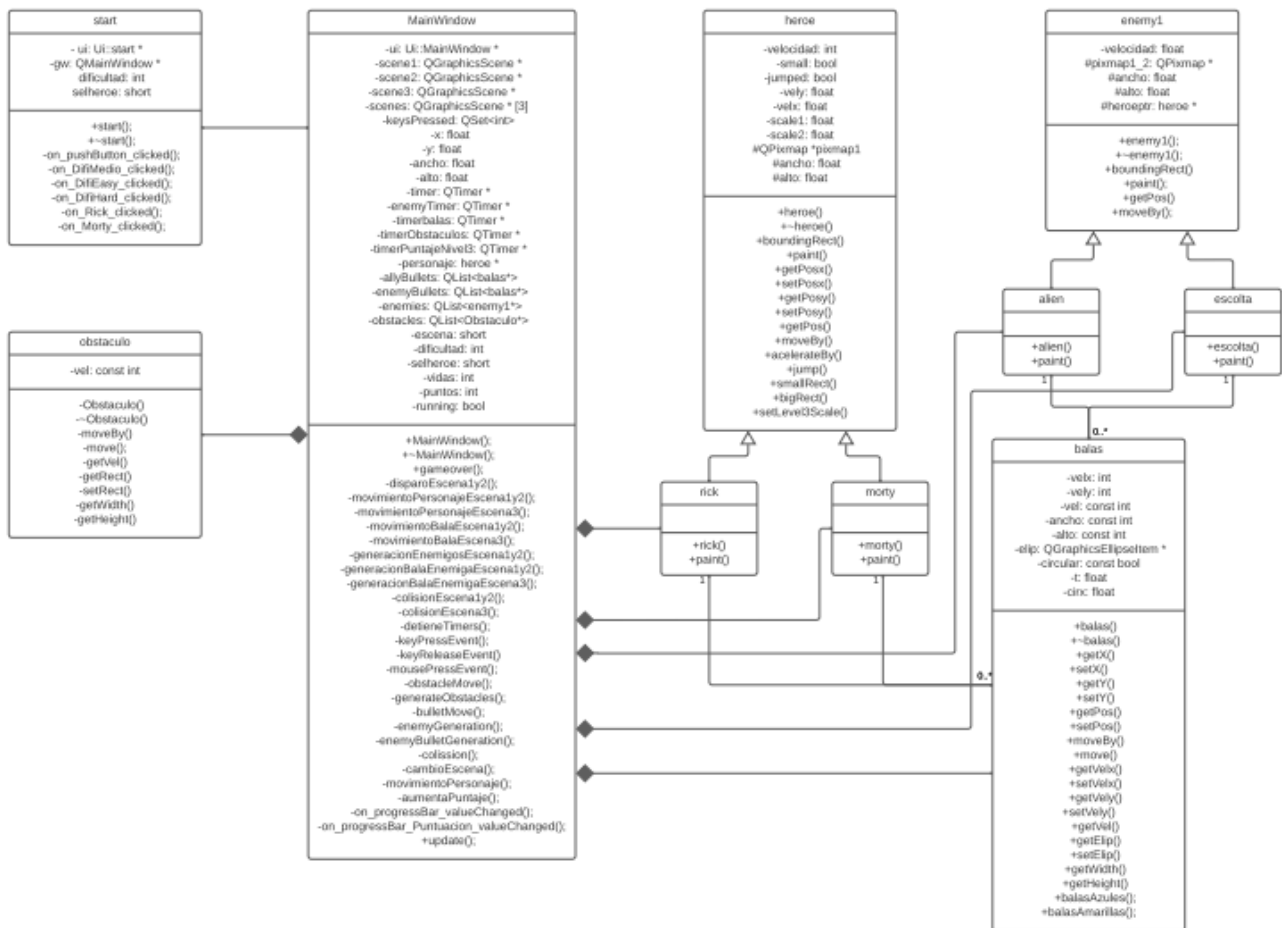


IMAGEN 2. Diagrama de clases implementadas.

CONCLUSIONES

- A lo largo del desarrollo del juego, se afianzaron conocimientos sobre el lenguaje C++, se permitió poner en práctica todos los conceptos explicados teóricamente.
- La programación orientada a objetos permite la creación de clases y objetos, lo que facilita la abstracción y encapsulamiento de datos y funciones. Esto ayuda a organizar y estructurar el código de manera más clara y eficiente, lo que es crucial en el desarrollo de aplicaciones y juegos como en este caso, una buena organización y estructuración permite que el código sea entendible por otros programadores.
- La herencia es crucial en el desarrollo de juegos, ya que permite la creación de jerarquías de clases, donde diferentes elementos comparten características comunes, de esta manera se permite que el desarrollo de código no se extienda demasiado y tener una mejor estructura de código.
- La reutilización de código es un principio fundamental. En el desarrollo de videojuegos, donde ciertos componentes y funcionalidades pueden ser comunes entre diferentes elementos u objetos del juego, la reutilización de clases y objetos facilita la escritura de código eficiente y reduce la duplicación optimizando recursos computacionales.
- La gestión de memoria es crucial para el rendimiento en videojuegos. Cada programador debe ser consciente del uso de memoria en sus respectivos programas teniendo en cuenta los recursos con los que cuenta su máquina. El uso adecuado de punteros y la liberación de memoria son aspectos clave a tener en cuenta.