Presentado por:

Daniela Tamayo Galeano

Andrés Camilo Agudelo Cadavid

Profesor:

Augusto Salazar

Curso:

Informática II

Facultad De Ingeniería

Universidad De Antioquia

La primera clase principal se llamará Juego, y contendrá los siguientes objetos como atributos públicos: el escenario, los duendes, el mago y los objetos(la barita). Esto será una herencia de clases.

**Mago \*jugador:** Representa a nuestro personaje del juego, el mago. Contendrá información como su posición en el escenario, su vida actual y sus habilidades.

Atributos:

1. Posición: guarda la posición actual del mago en el escenario. (X,Y)
2. Vida: indica la cantidad de vida restante del mago.

Métodos:

1. Moverse: permite al mago moverse en el escenario en respuesta a las acciones del jugador.
2. Disparar: activa el disparo de un proyectil bola de fuego.
3. Recolectar Barita: Maneja la lógica de recolectar la barita mágica.

**Duende \*enemigo []:** Este representará a los enemigos del juego. Cada duende contendrá información como su posición, vida y poderes.

Atributos:

1. Posición: guarda la posición actual del duende.
2. Vida: muestra la cantidad de vida restante del duende.

Métodos:

1. Moverse: determina el movimiento automático del duende en el escenario.
2. Atacar: controla la lógica de ataque del duende hacia el mago.
3. Morir: maneja las acciones a realizar cuando el duende es matado.

**Objeto \*barita[]:**Es un arreglo de Objetos que representa los objetos especiales del juego, como la barita mágica. Cada objeto tendrá información como su posición y tipo.

Atributos:

1. Posición: guarda la posición actual del objeto en el escenario.

Métodos:

1. Interactuar: define la interacción del mago con el objeto, como recolectarlo.

**BolaFuego \*proyectiles[]:** Esta serían los proyectiles, las bolas de fuego, utilizados por el mago. Cada proyectil tendrá información como su posición y velocidad de movimiento.

Atributos:

1. Posición: guarda la posición actual de la bola de fuego en el escenario.

1. Velocidad: determina la velocidad de movimiento de la bola de fuego.

Métodos:

1. Moverse: controla el movimiento de la bola de fuego en el escenario.
2. Colisionar: maneja las acciones a realizar cuando la bola de fuego colisiona con un enemigo u otro objeto.

Por otro lado, algunas funciones de la clase Juego podrían ser:

void iniciarJuego():

void actualizar():

void ManejarMago();

void choque();

void disparo();

void generarObjetos();

void verificarEstado():

void Eliminar():

**void iniciarJuego():** Aquí se inicializaría el juego, creando y configurando los objetos necesarios, estableciendo la escena, cargando los recursos.

**void actualizar():** En cada iteración del bucle principal del juego, esta se encargaría de actualizar el estado de los objetos, realizar movimientos, detectar colisiones, gestionar eventos, etc. En este se aplicaría la lógica del juego.

**void mostrar():** Esta función se encargaría de mostrar los objetos en la escena, actualizando su posición y apariencia.

**void manejarMago ():** Esta función se encargaría de gestionar la entrada del jugador, detectando las teclas pulsadas y realizando las acciones correspondientes, como mover al mago, realizar disparos.

**void choque ():** Solo se activa cuando hay una colisión entre el mago y un duende o un muro. Puede implementar las acciones que ocurren cuando se produce una colisión, como disminuir la vida del mago, eliminar un duende o activar algún efecto especial.

**void disparo ():** Este se encarga de permitir al mago disparar las bolas de fuego. Puede controlar la creación de un nuevo proyectil y su movimiento en el escenario, así como detectar colisiones con los duendes u otros objetos.

**void generarObjetos():** Generará objetos en el escenario, como la barita mágica que el mago debe recolectar.

**void verificarEstado():** Esta función se encargaría de verificar si se ha alcanzado una condición de ganar o derrota en el juego. Puede comprobar si el mago ha recolectado todos las baritas, si se ha agotado la vida del mago.

#ifndef JUEGO\_H

#define JUEGO\_H

#include "MagoJugador.h"

#include "Duende.h"

#include "Objeto.h"

#include "BolaFuego.h"

#include “Enemigo.”

class Juego {

private:

MagoJugador mago;

Duende\* enemigos[];

Objeto\* barita[];

BolaFuego\* proyectiles[];

public:

Juego();

void iniciarJuego();

void actualizar();

void mostrar();

void manejarMago();

void choque();

void disparo();

void Eliminar():

void generarObjetos();

};

#endif

**Interfaz Gráfica**

Nuestro juego será desarrollado en Pixel Art, en un entorno 2D con vista UpDown utilizando las bibliotecas de interfaz gráfica de Qt, específicamente QGraphicsPixmapItem y QGraphicsScene. Utilizaremos estas bibliotecas para cargar y mostrar los elementos del juego en pantalla.

En cuanto a las interacciones del jugador, utilizaremos las teclas ASDW para controlar el movimiento del personaje principal. La tecla ESPACIO se asignará para que el mago pueda lanzar disparos.

A través de QGraphicsPixmapItem, podremos representar los sprites de los personajes y los elementos del juego. Utilizando QGraphicsScene, podremos crear y administrar la escena del juego, cargar los elementos en ella y realizar operaciones como detección de colisiones, actualizaciones y visualización en pantalla.

Imagen de la pantalla de un video juego

Descripción generada automáticamente con confianza baja