**DECLARACIÓN DE CLASES**

*Ricardo José Portnoy Tobías*

*Luis Alberto Ardila Páez*

1. **INTRODUCCIÓN**

En el presente documento se explicará al detalle y se mostrará la declaración de las clases necesarias para la realización del proyecto final, videojuego de carreras en el espacio *Across Lighspeed*.

1. **DECLARACIÓN**

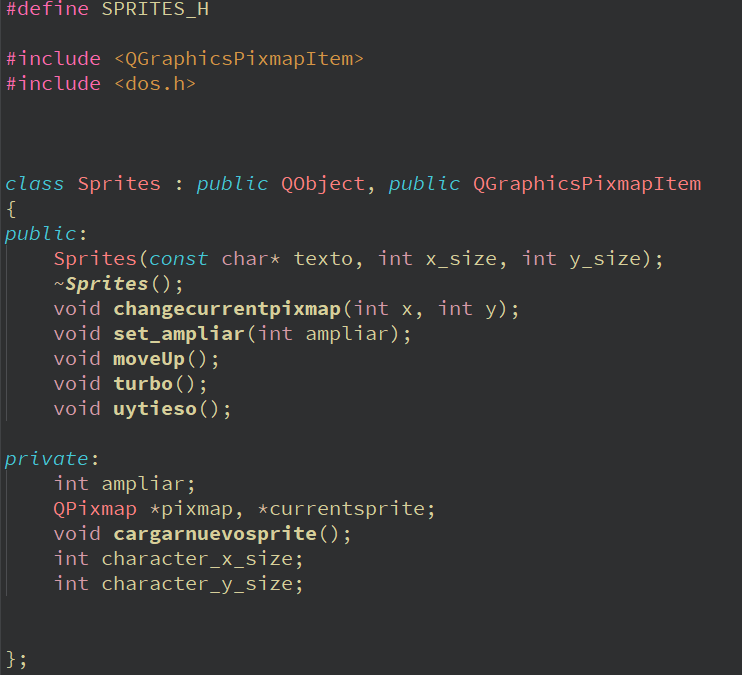
**II.I SPRITES**

Con la clase *SPRITES* se logrará representar gran cantidad de objetos en el juego como a la vez de cargar un Sprite para que aparezca en pantalla, esta clase heredara públicamente a *QObject* y a *QGraphicsPixmapItem.*

*SPRITES* nos permitirá manejar dos cosas importantes a la hora de realizar obstáculos, enemigos o la misma nave del jugador, el cual es además de un Sprite para que lusca agradable, una *hitbox* con la que en seremos capaces de detectar las colisiones.

*SPRITES* posee métodos para cambiar el Sprite en pantalla, esto servirá cuando por ejemplo el usuario quiera acelerar se le mostrará el Sprite de su nave acelerando o cuando este gire se muestre el Sprite de su nave cambiando de dirección. Proporciona métodos muy útiles para el manejo de los Sprite y permite que posteriormente se detecten colisiones con útil para clases posteriores.

***Ilustración clase SPRITE***

****

**II.II *BACKGROUND***

Los objetos tipo *BACKGROUND* son los encargados de representar los fondos de cada uno de los niveles del videojuego. Al ser el fondo un apartado sumamente importante en el tipo de juego que se planea construir, poseer una clase que establezca los fondos y distintos métodos para interactuar con ellos es indispensable. *BACKGROUND* esta conformada por atributos como un *QImage* el cual contendrá la imagen del mapa del nivel, cuatro enteros, un par representará las dimensiones de la pantalla del juego y el otro par las dimensiones de la imagen del nivel. Esta clase es la responsable de la sensación de perspectiva del nivel y en sí se podría decir que esta clase es un modificador de imágenes ya que de una imagen base la cual es el mapa del nivel según ciertos parámetros lo modificara para dar la sensación de perspectiva. Se necesitan como parámetros varios enteros que representen el ancho y largo del mapa, el largo y ancho de la pantalla y la escala del mundo. Flotantes como la posición del jugador en dicho mapa y el Angulo de visión del jugador y muy importante el *QColor* que representara las áreas por las cuales el jugador no podrá pasar y la imagen base la cual constantemente se le realizaran cambios para dar el efecto perspectiva.

Entre los distintos métodos encontramos los movimientos del jugador que son ir hacia adelante, atrás o girar en cierta direccione, los *Getters* y *Setters* de las variables antes mencionadas y el método actualizar que será el responsable de actualizar la imagen a medida que el jugador se va moviendo a lo largo del mapa.

***Ilustración clase BACKGROUND***

***Texto

Descripción generada automáticamente***

**II.III ENEMIGO**

Enemigo es la clase encargada de representar a los distintos adversarios del juego, esta clase heredara públicamente todos los atributos y métodos de *SPRITE* ya que, los enemigos necesitan un Sprite que los represente visualmente como también una *Hitbox* para identificar sí chocamos o no con ellos. La principal cualidad de los objetos enemigo es su nivel de dificultad, a medida que avanza el juego los enemigos varían su comportamiento dependiendo del nivel de dificultad que estos posean, además que tienen ciertos métodos únicos como *ATACAR* y *APROXIMARSE*. Los enemigos podrán atacar o aproximarse a un personaje indicado.

Atacar: se necesita un personaje el cual atacar, de aquí calcularemos su posición y dicho objeto enemigo variara su velocidad para provocar un choque.

Aproximarse: este método permite que los enemigos no sean superados tan fácilmente, sí la distancia entre un personaje y dicho enemigo es considerablemente alta el enemigo cambiara su posición en el mapa para estar cerca del personaje al cual quiera aproximarse.

***Ilustración clase ENEMIGO***

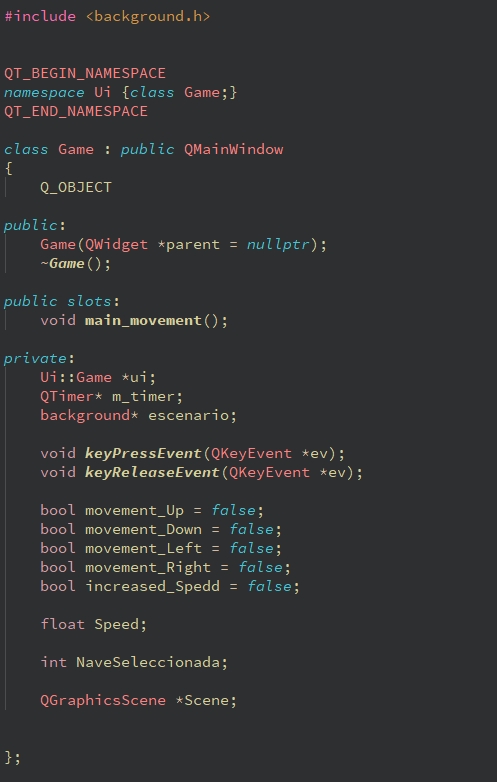
***Texto

Descripción generada automáticamente con confianza baja***

**II.IV JUEGO**

La clase *JUEGO* es responsable de coordinar todos los elementos del juego, desde el control de las teclas presionadas por el jugador, hasta el manejo de los personajes, enemigos y escenarios. Proporciona una interfaz de interacción y encapsula la lógica interna necesaria para que el proyecto funcione correctamente. Datos esenciales como el tiempo el cual lleva el jugador en el nivel, el numero de vueltas las cuales le faltan para terminar la carrera, el escenario a jugar, y la distinta asignación de los obstáculos en el mapa están realizados por esta clase.

***Ilustración clase JUEGO***

****