QMainWindow

MainWindow

+Ui::MainWindow \*ui;  
+QGraphicsScene \*scene; +QGraphicsView \*view;   
+arma\* miArma;  
+proyectil\* Min;  
+int angulo;  
+QGraphicsPixmapItem \*bart;  
+Protagonista \*protagonista; +enemigo \*enemigo1 = nullptr;  
+enemigo \*enemigo2 = nullptr;  
+enemigo \*enemigo3 = nullptr;  
+ int sceneWidth = 800  
+ int sceneHeight = 600  
+ int tileSize = 50

QGraphicsRectItem

obstaculo

Obstaculo(qreal x, qreal y, qreal width, qreal height, QObject \*parent)

Nivel

-QGraphicsScene\* scene;  
-Protagonista\* protagonista;

void actualizarEnemigos();  
 void dispararProyectil();

suelo

Suelo(qreal x, qreal y, qreal width, qreal height, QGraphicsItem\* parent)

+Nivel(QGraphicsScene\* scene);  
+void crearNivel();

QObject

QPixmap

QGraphicsPixmapItem

Personaje

- double posX;  
- double posY;  
- double velocidad;  
+ unsigned short int vidas;  
  
 + virtual void IniciarAnimacion(bool caminar);

Protagonista

Protagonista  
  
- bool detenerMovimiento = false;  
- int x, y;  
- int speed\_x, speed\_y;  
- int vidas;  
- bool isDead;  
- bool angulo;  
- QPixmap spriteReposo; - QPixmap spriteCaminar1;   
- QPixmap spriteCaminar2; - QPixmap spriteSalto;  
  
+ Protagonista(int startX, int startY, QGraphicsPixmapItem\* parent = nullptr);  
+ ~Protagonista();  
+ void keyPressEvent(QKeyEvent\* event) override;  
+ void keyReleaseEvent(QKeyEvent\* event) override;  
+ void update();  
+ void recibirDano(int dano);  
+void IniciarAnimacion(bool caminar) override;  
+ void habilitarArma();  
+ void ejecutarArmaAnimacion();  
+ void dispararConArma();  
+ int obtenerDireccionDelProtagonistda();  
+ void setArma(arma\* nuevaArma);  
- void mover(int dx, int dy);  
- void actualizarAnimacion();  
- void morir();  
- void perderVida();

-int x, y;  
- int speed\_x, speed\_y;  
- Tipo tipoEnemigo;  
- unsigned short int vidas;  
- QPixmap imagen1;   
- QPixmap imagen2;  
- bool alternarFrame;  
 + enum class Tipo { Enemigo1, Enemigo2, Enemigo3 };

Enemigo

1 N

Arma

1 1  
  
  
  
  
  
  
  
  
   
  
 1   
   
   
  
 N

+ Explicit arma(QObject \*parent = nullptr);  
+ void verificarColision(QGraphicsItem\* protagonista);

// Método para disparar proyectiles, tomando como parámetros la dirección y velocidad

void disparar(int velocidadX, int velocidadY, qreal posX, qreal posY);

int getVelocidad();

- bool recogida;

QGraphicsEllipseItem

+ explicit enemigo(int startX, int startY, Tipo tipo, QObject \*parent = nullptr);  
+ void mover();  
+ void update();  
+ void recibirDano(int dano);  
+ void morir();  
+void IniciarAnimacion(bool caminar) override;

N

Proyectil

N

- int x, y;   
- int speedX, speedY;   
- int angulo;  
- const int gravedad = 1; QTimer\* timerMovimiento;

+ void actualizarPosicion();  
+ void iniciarMovimiento();   
+ int getposX();  
+ int getposY();