

DIAGRAMA DE CLASES

Chernobil - Ultima Lúz

Jorge Enrique Rueda Urrea - 1.020.488.162

Edison Fredy Serrano Álvarez - 1.055.273.822

Ingeniería de Telecomunicaciones

Fecha: Mayo 2024



UNIVERSIDAD
DE ANTIOQUIA

Facultad de Ingeniería

Keywords: Simulación, Punteros, Métodos, Clases, Arreglos

1 CONSIDERACIONES

- Se utiliza el paradigma de POO para encapsular la información y el comportamiento de los elementos.
- Se definen clases para representar cada elemento con sus atributos y métodos correspondientes.
- Las relaciones entre las clases se establecen mediante asociaciones.

2 ESTRUCTURAS DE DATOS IMPORTANTES

- **Arreglos:** Se utilizarán arreglos dinámicos para almacenar las colecciones de elementos.
- **Punteros:** Se utilizarán punteros para gestionar las relaciones entre las clases y para almacenar las referencias a los elementos.

3 DIAGRAMA DE CLASES

• CLASE#1: enemigo

• Atributos:

- int size
- float x
- float y
- float auxx
- float auxy
- float r
- float the
- float v

• Métodos:

- enemigo()
- enemigo(float posmx, float posmy, int tam, int vel)
- void movimiento()

• CLASE#2: MainWindow

• Atributos:

- Ui::MainWindow *ui
- QGraphicsScene *scene
- QGraphicsScene *scene2

- sprite *Yuri
- QList<pared> paredes
- QList<puerta> puertas
- pasarnivel *nextLevelTrigger
- QLabel *nextLevelLabel
- QMap<puerta*, QPointF> puertaOriginalPositions
- QMap<puerta*, QTimer> puertaTimers

• Métodos:

- explicit MainWindow(QWidget *parent = nullptr)
- MainWindow()
- void keyPressEvent(QKeyEvent *event)
- void closeDoor(puerta *p)
- void switchToNextScene()
- void setupExternalWalls(QGraphicsScene *scene)

• CLASE#3: pared

• Atributos:

- QRectF mrect
- QColor mcolor

• Métodos:

- pared(qreal x, qreal y, qreal width, qreal height, QColor color = QColor(0, 0, 0), QGraphicsItem *parent = nullptr)
- QRectF boundingRect() const
- void paint(QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget = nullptr)

• CLASE#4: puerta

• Atributos:

- int posX
- int posY
- int w
- int h

• Métodos:

- puerta()
- puerta(int x, int y, int w, int h)
- QRectF boundingRect() const
- void paint(QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget)

• CLASE#5: pasarnivel

• Métodos:

- pasarnivel(int x, int y, int width, int height)

• CLASE#6: sprite

• Atributos:

- QTimer *timer
- QPixmap *pixmap
- float filas
- float columnas
- float ancho
- float alto
- Direction currentDirection

• Métodos:

- explicit sprite(QObject *parent = nullptr)
- QRectF boundingRect() const
- void paint(QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget)
- void setDirection(Direction direction)
- bool checkCollision(int newX, int newY)
- void Actualizacion()

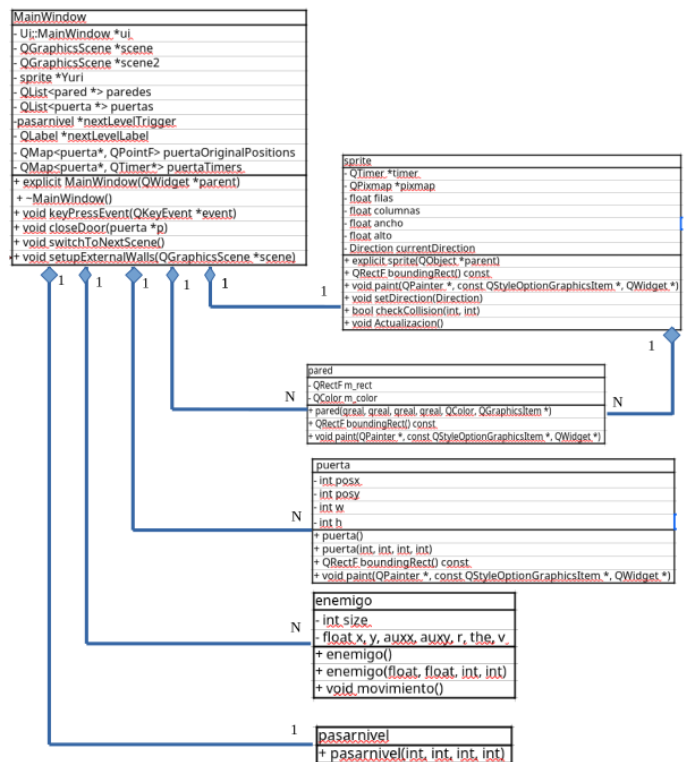


Figure 1: Diagrama de clases simplificado.

En este diagrama se utilizan los especificadores de acceso "-" para indicar atributos privados y "+" para indicar métodos públicos.

4 RELACIONES

- MainWindow contiene listas de pared y puerta.
- MainWindow tiene un objeto sprite (Yuri) y un objeto pasarnivel (nextLevelTrigger)
- MainWindow puede crear y añadir objetos enemigo a sus escenas.
- puerta y pared heredan de QGraphicsItem.
- sprite hereda de QObject y QGraphicsItem.
- enemigo hereda de QObject y QGraphicsPixmapItem.