

UNIVERSIDAD DE ANTIOQUIA.

1.

INFORME PROYECTO FINAL:
CONSTANTINOPOLIS.



INTEGRANTES:

GIRALDO ÚSIGA MAURICIO.

PARRA OSORIO RAFAEL IGNACIO.

ASIGNATURA: INFORMÁTICA II.

DOCENTE: AUGUSTO ENRIQUE SALAZAR JIMENEZ.

MEDELLÍN.

MMXXIV.

ÍNDICE:

Idea.....2.

Diagrama de clases.....3.

IDEA:

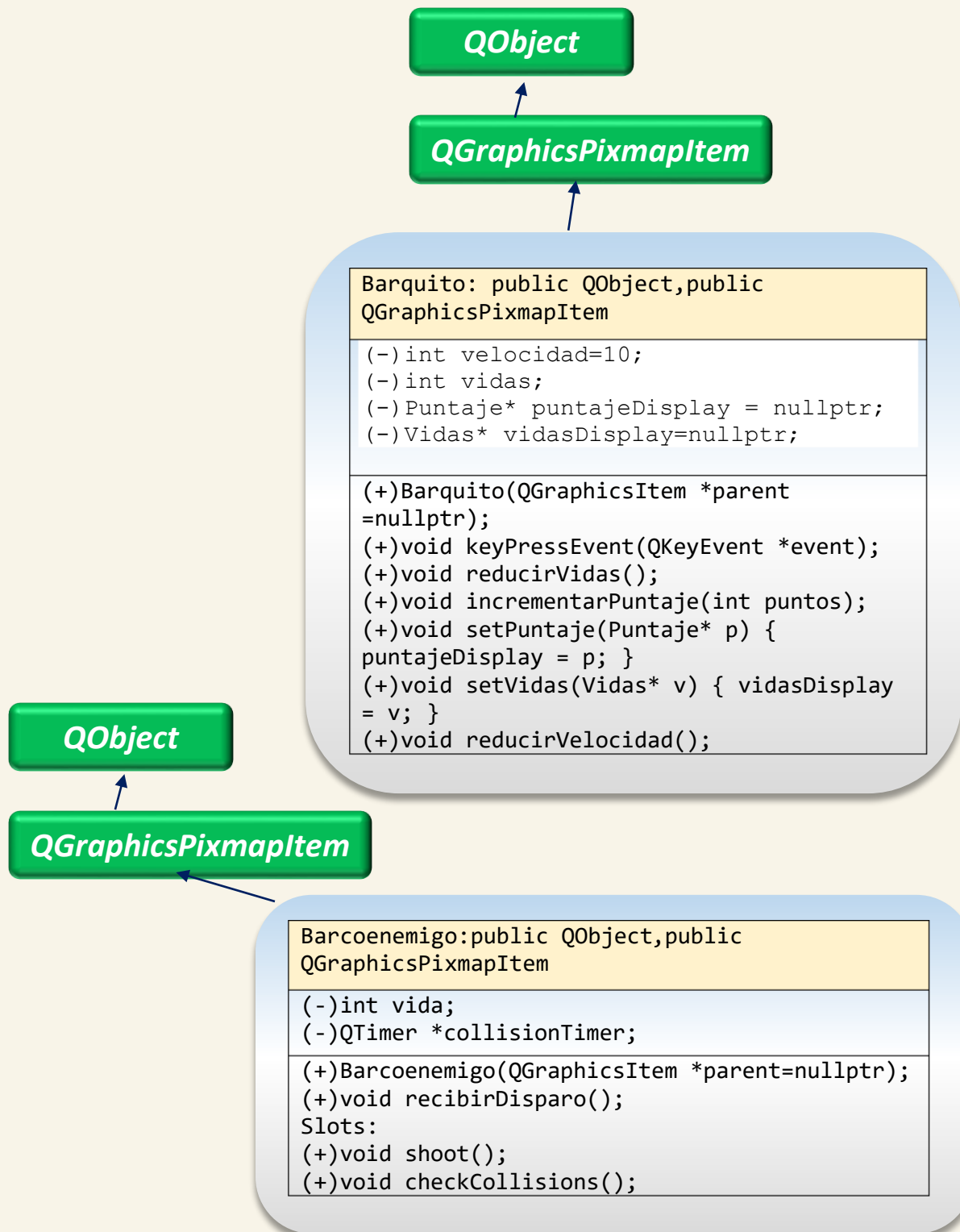
Para este proyecto nos hemos inspirado en un acontecimiento histórico que fue la caída de la ciudad amurallada de Constantinopla a manos del Imperio Otomano, gobernado por el sultán Mehmed II. Dicho acontecimiento tuvo lugar el 29 de mayo del año 1453 d.C.

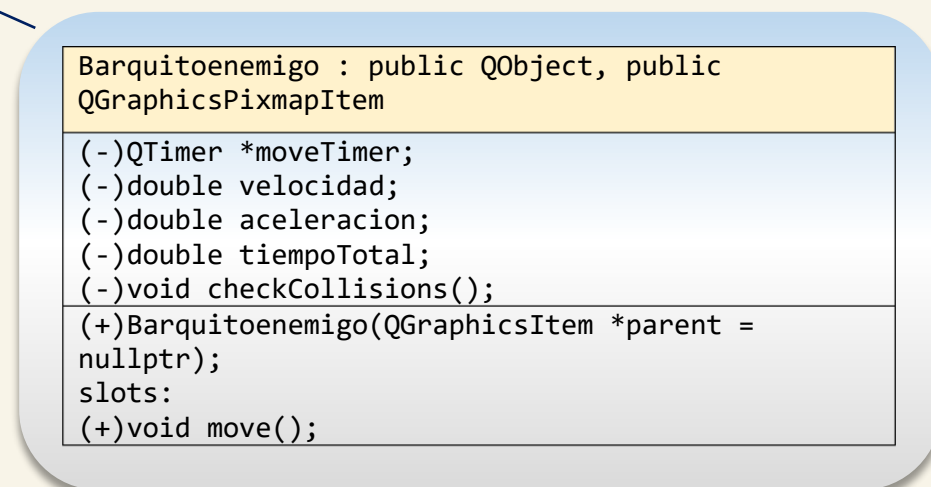
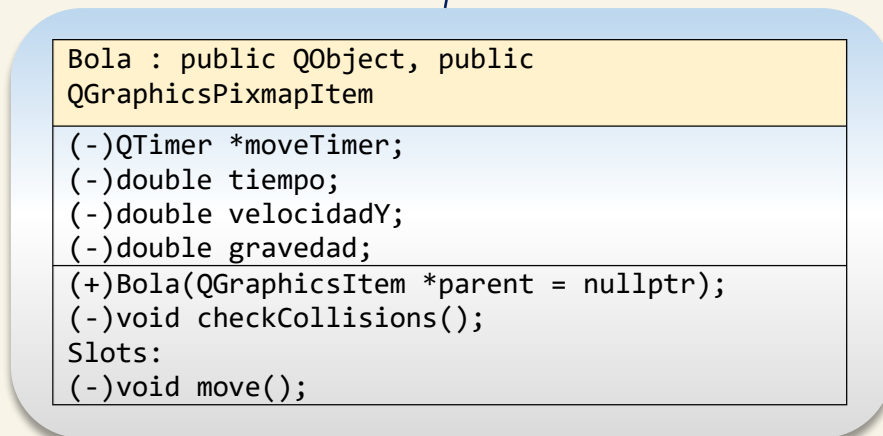
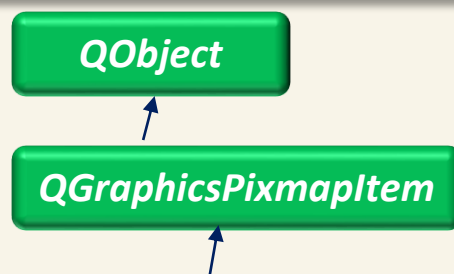
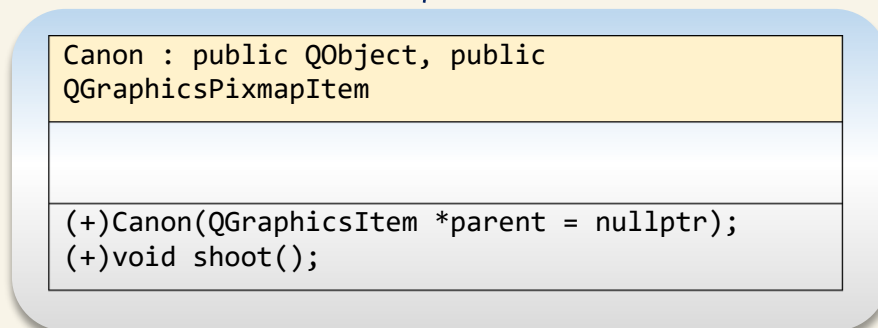
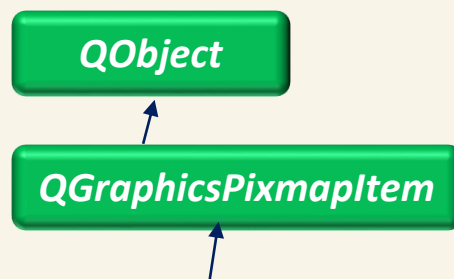
El juego lo hemos decidido dividir en dos niveles, los cuales están ambientados en dos momentos cruciales del asedio otomano a Constantinopla:

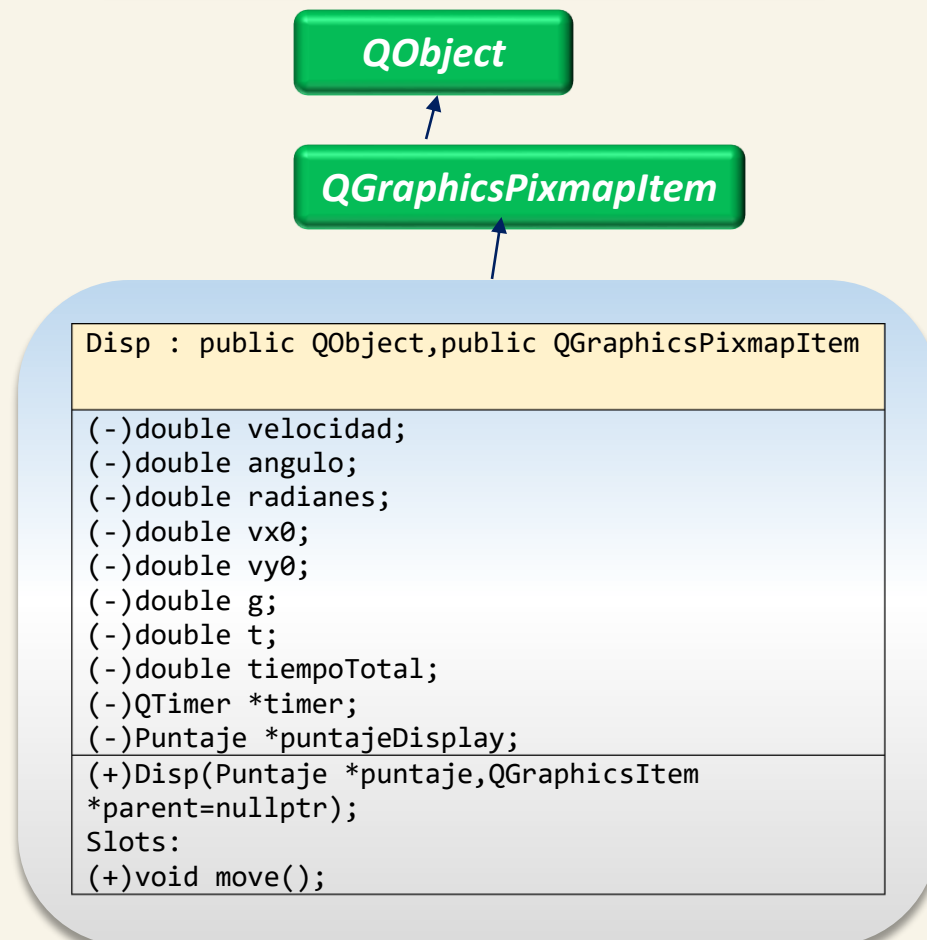
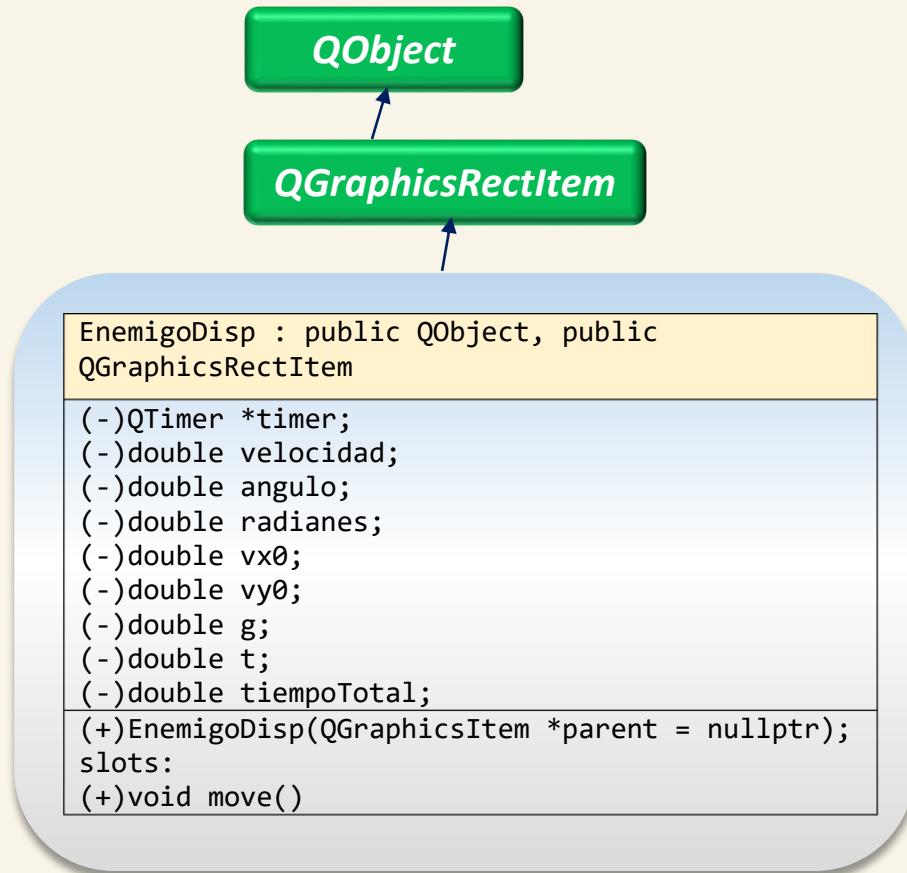
- 1) La batalla naval que tuvo lugar en las aguas de la metrópolis en el estrecho del Bósforo, el mar Negro y el mar del Mármara, así como la infiltración de galeras otomanas que moviéndose por tierra a través de la península del Gálata llegaron hasta las aguas del Cuerno de Oro, para luego desembarcar sus tropas y municiones cerca de la entrada de San Romano.
- 2) La batalla campal final en tierra que tuvo lugar en varios flancos de la muralla de Constantinopla, en donde se usó un famoso cañón gigante para derribar sus muros llamado “cañón de Orbán”, diseñado por un ingeniero húngaro. Finalmente los otomanos consiguieron la victoria luego de entrar a la ciudad por la entrada de San Romano.

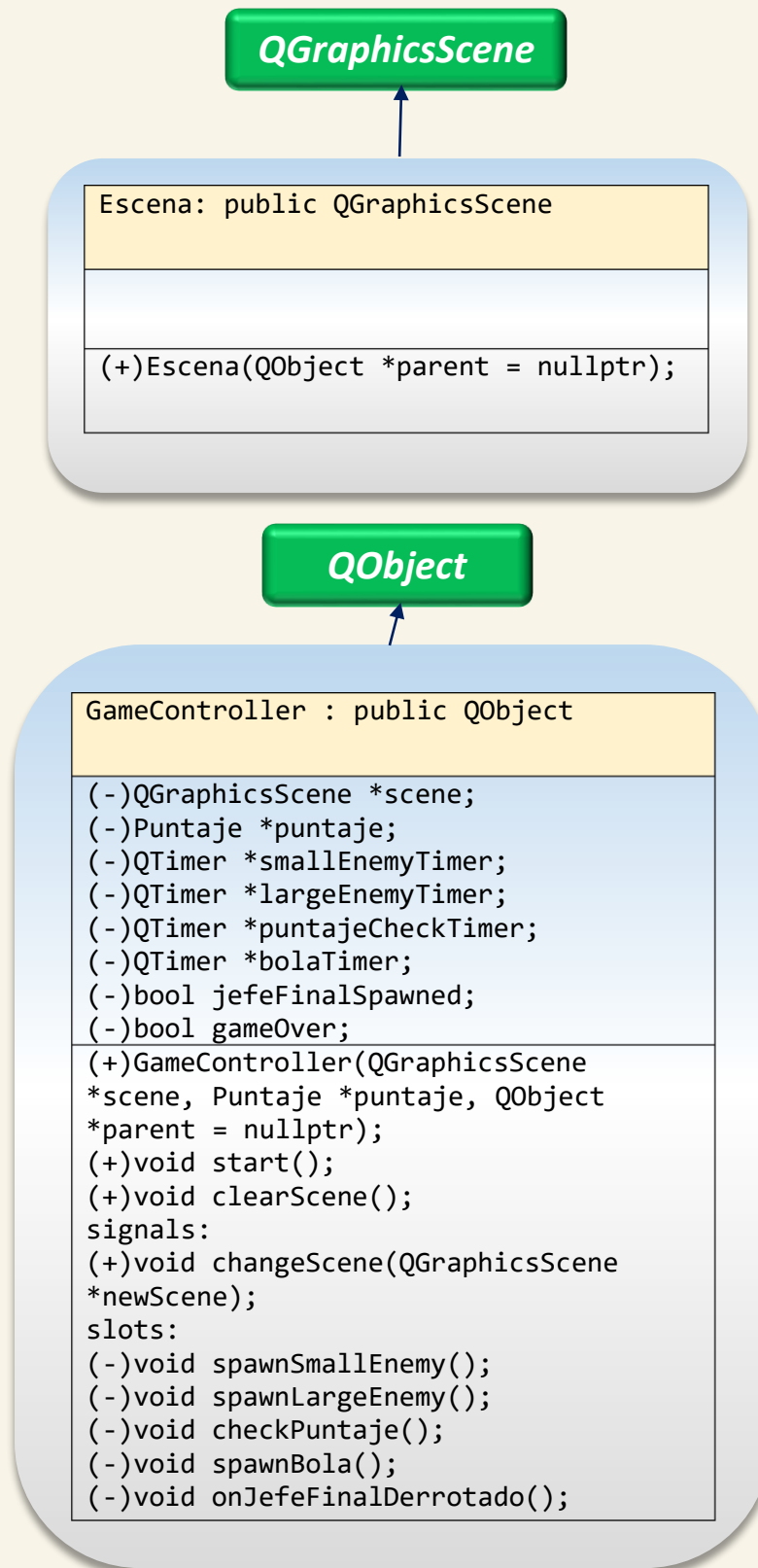


DIAGRAMA DE CLASES:









QObject

GameControllerNivel2 : public QObject

(-)QGraphicsScene *scene;
 (-)QTimer *spawnTimer;
 (-)PolvoraDisplay *polvoraDisplay;
 (-)PuertaDisplay *puertaDisplay;
 (-)VidaDisplay *vidaDisplay;
 (-)soldado_otomano *jugador;
 (-)Canon *canon;
 (-)Puerta *puerta;

(+)explicit

GameControllerNivel2(QGraphicsScene
 *scene,soldado_otomano *jugador,Canon
 *canon,Puerta *puerta, QObject *parent
 = nullptr);

(+)~GameControllerNivel2();

Slots:

(-)void spawnEnemigo();

(+)void recolectarPolvora();

QObject

QGraphicsPixmapItem

JefeFinal : public QObject, public
 QGraphicsPixmapItem

(-)int direccion;
 (-)int vida;
 (-)void checkCollisions();
 (-)QTimer *collisionTimer;

(+)JefeFinal(QGraphicsItem *parent =
 nullptr);

(+)void recibirDisparo();

(+)int getVida() const;

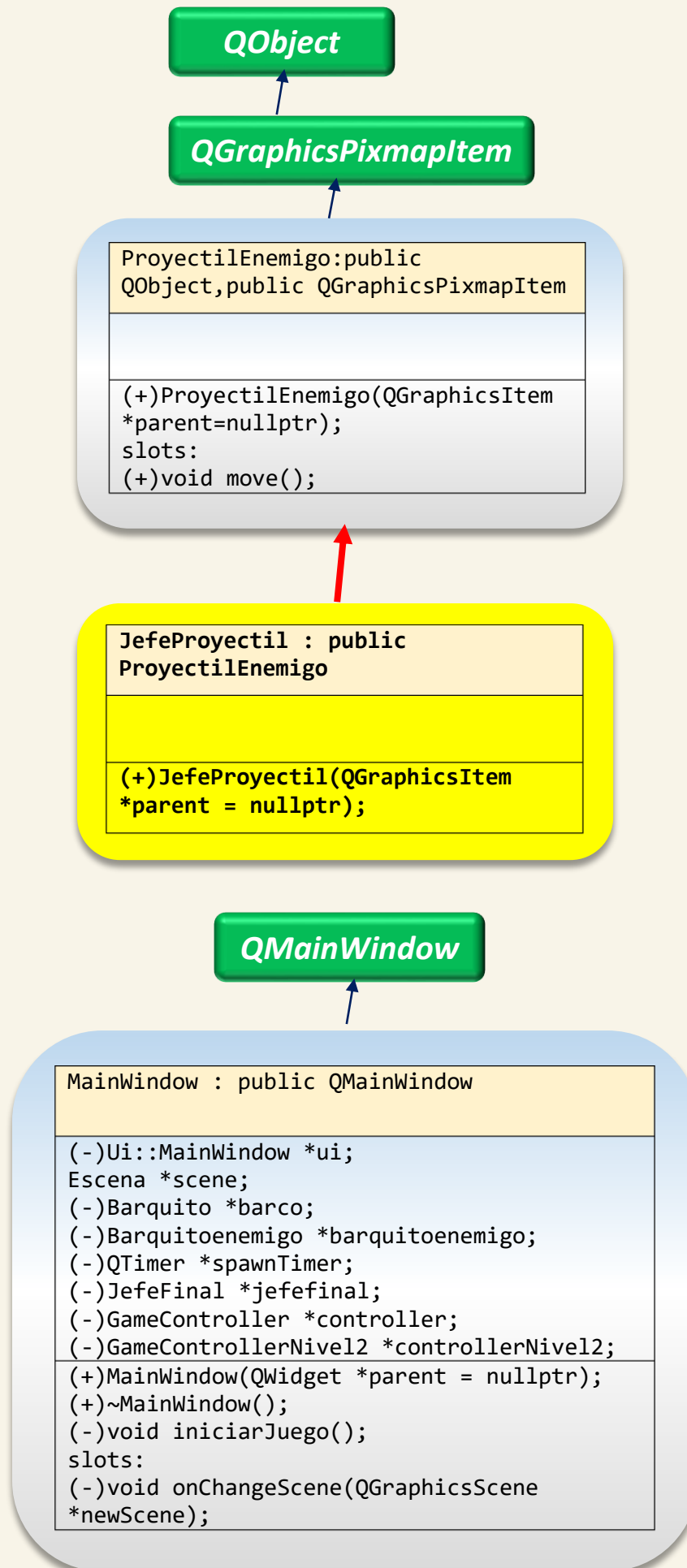
Signals:

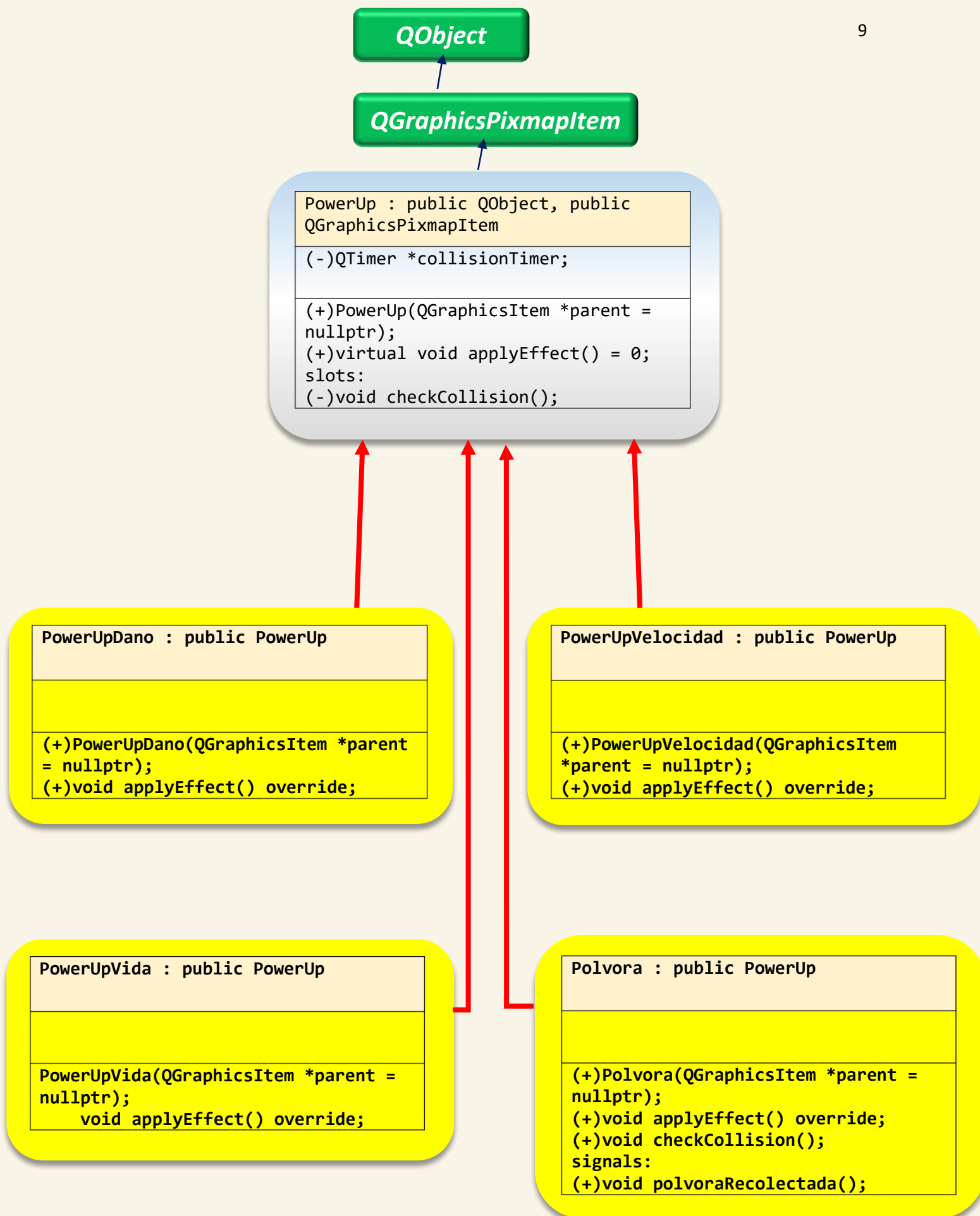
(+)void jefeDerrotado();

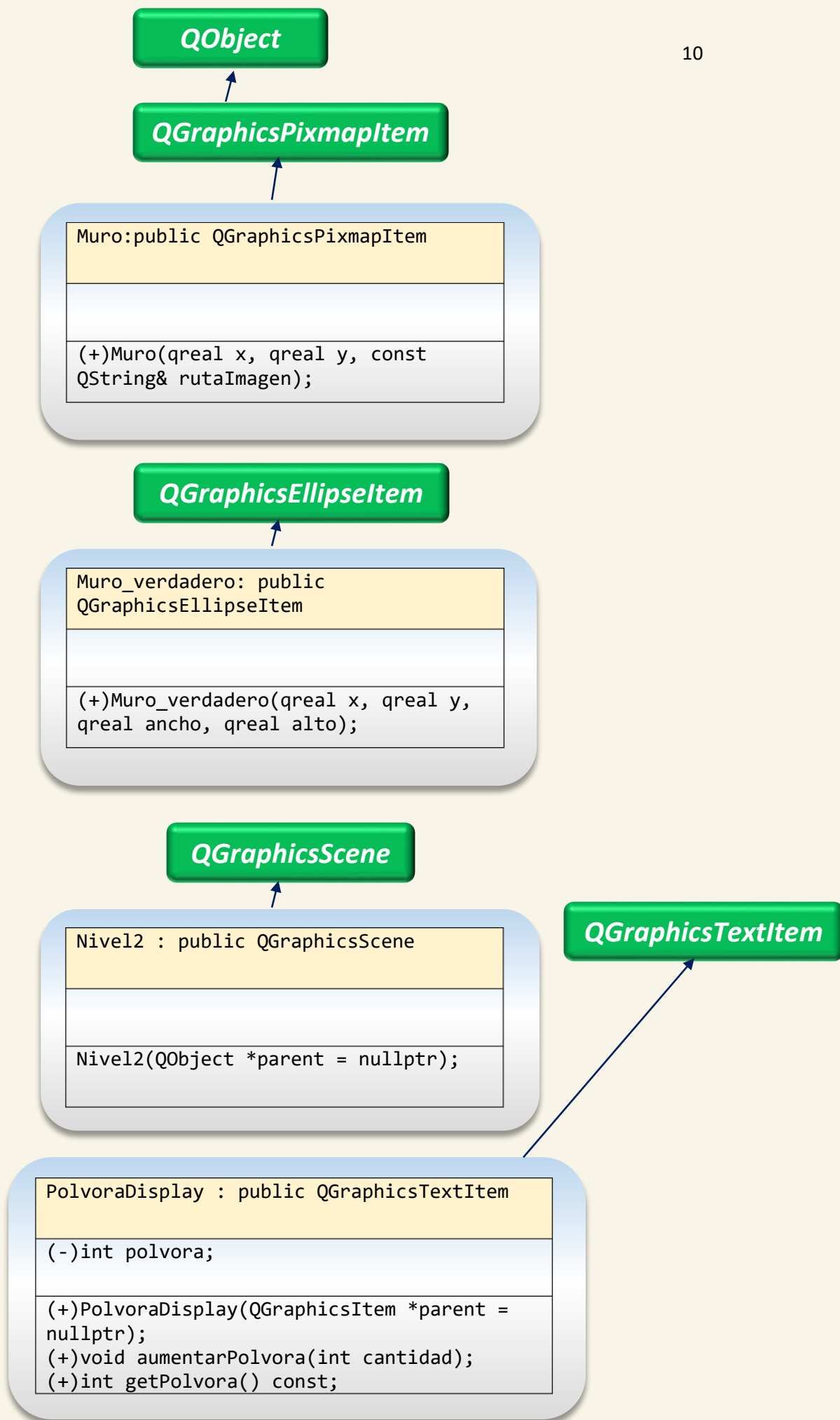
Slots:

(-)void move();

(-)void shoot();







QObject**QGraphicsPixmapItem**

```
Proyectil : public QObject, public
QGraphicsPixmapItem
```

```
(-)QTimer *moveTimer;
(-)qreal angle;
(-)qreal initialSpeed;
(-)qreal currentSpeed;
(-)qreal deceleration;
(-)QElapsedTimer timer;
```

```
(+)Proyectil(qreal angle, QGraphicsItem
*parent = nullptr);
slots:
(-)void move();
```

QObject**QGraphicsPixmapItem**

```
ProyectilEnemigoCanon : public QObject,
public QGraphicsPixmapItem
```

```
(+)explicit
ProyectilEnemigoCanon(QGraphicsItem
*parent = nullptr);
slots:
(+)void move2();
```

QObject**QGraphicsPixmapItem**

```
Puerta : public QObject, public QGraphicsPixmapItem
```

```
(-)int vida;
```

```
(+)Puerta(QGraphicsItem *parent = nullptr);
(+)void recibirDisparo();
(+)int getVida() const;
signals:
(+)void vidaCambiada(int nuevaVida);
```

QGraphicsTextItem

PuertaDisplay : public QGraphicsTextItem

```
(+)PuertaDisplay(QGraphicsItem *parent = nullptr);
(+)void actualizarVida(int vida);
```

QGraphicsTextItem

Puntaje : public QGraphicsTextItem

```
(-)int puntaje;
```

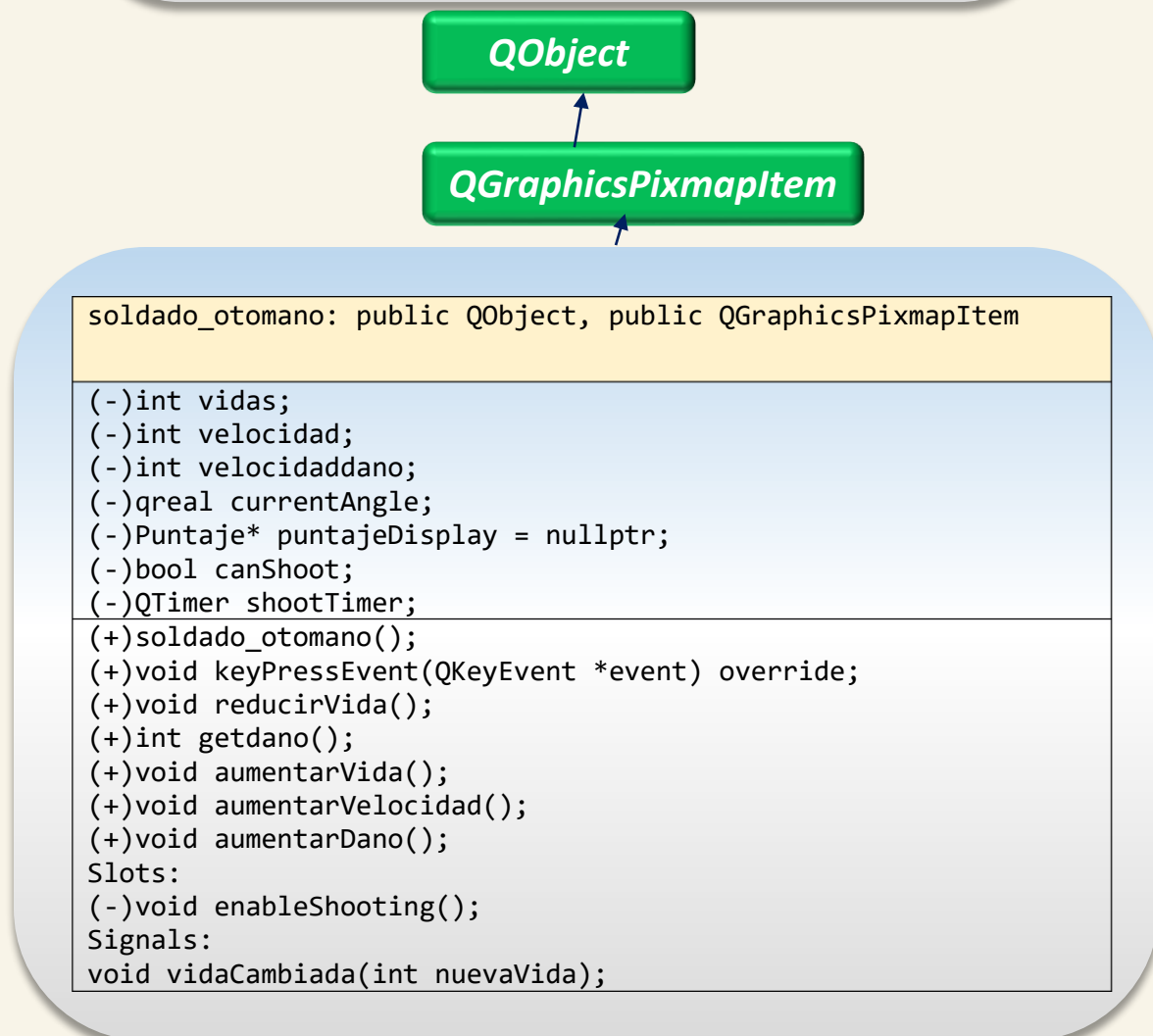
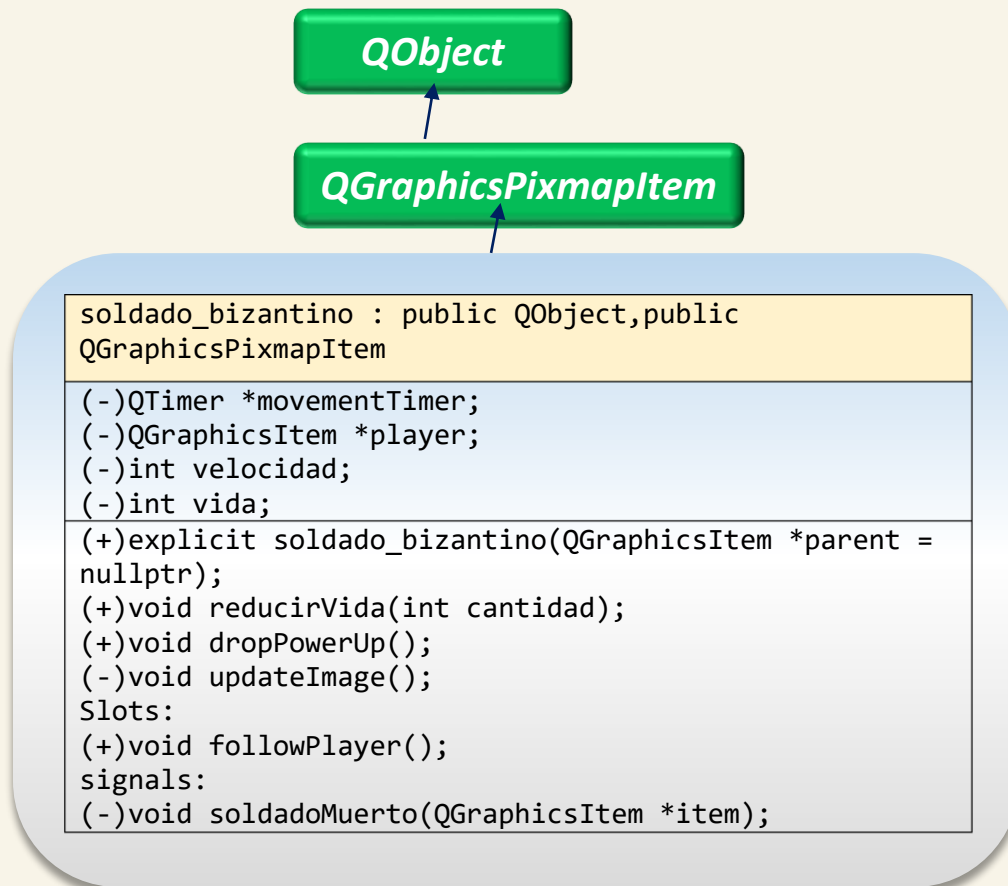
```
(+)Puntaje(QGraphicsItem *parent = nullptr);
(+)void incrementar(int puntos);
(+)int getPuntaje() const;
(+)bool hasReached(int target) const;
```

QDialog

Registro : public QDialog

```
(-)Ui::registro *ui;
(-)QMap<QString, QString> usuarios;

(+)explicit Registro(QWidget *parent = nullptr);
(+)~Registro();
(-)void registrarUsuario(const QString &nombreUsuario,
const QString &contrasena);
(-)void cargarUsuarios();
(-)void guardarUsuarios();
(-)bool autenticarUsuario(const QString
&nombreUsuario, const QString &contrasena);
signals:
(+)void usuarioAutenticado();
slots:
(-)void on_registrarButton_clicked();
(-)void on_iniciarSesionButton_clicked();
```



QGraphicsTextItem



VidaDisplay: public QGraphicsTextItem

```
(+)VidaDisplay(QGraphicsItem *parent= nullptr);  
(+)void actualizarVida(int vida);
```

QGraphicsTextItem



Vidas : public QGraphicsTextItem

```
(-)int vida;
```

```
(+)Vidas(int vid, QGraphicsItem *parent = nullptr);  
(+)void decrementar();  
(+)int getPuntaje();  
(+)int getVidas();
```