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.

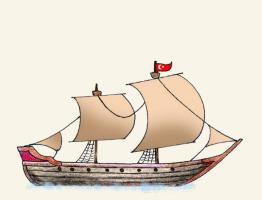
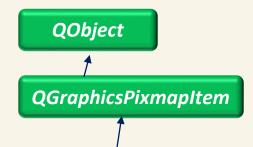




DIAGRAMA DE CLASES:



```
Barquito: public QObject,public
QGraphicsPixmapItem
```

(-)int velocidad=10;

```
(-)int vidas;
(-)Puntaje* puntajeDisplay = nullptr;
(-)Vidas* vidasDisplay=nullptr;

(+)Barquito(QGraphicsItem *parent
=nullptr);
(+)void keyPressEvent(QKeyEvent *event);
(+)void reducirVidas();
(+)void incrementarPuntaje(int puntos);
(+)void setPuntaje(Puntaje* p) {
  puntajeDisplay = p; }
(+)void setVidas(Vidas* v) { vidasDisplay
= v; }
(+)void reducirVelocidad();
```

QObject

QGraphicsPixmapItem

```
Barcoenemigo:public QObject,public QGraphicsPixmapItem
```

```
(-)int vida;
(-)QTimer *collisionTimer;
(+)Barcoenemigo(QGraphicsItem *parent=nullptr);
(+)void recibirDisparo();
Slots:
(+)void shoot();
(+)void checkCollisions();
```

```
QObject
                                QGraphicsPixmapItem
                    Canon : public QObject, public
                    QGraphicsPixmapItem
                    (+)Canon(QGraphicsItem *parent = nullptr);
                    (+)void shoot();
                                  QObject
                                QGraphicsPixmapItem
                   Bola : public QObject, public
                   QGraphicsPixmapItem
                   (-)QTimer *moveTimer;
                   (-)double tiempo;
                   (-)double velocidadY;
                   (-)double gravedad;
                   (+)Bola(QGraphicsItem *parent = nullptr);
                   (-)void checkCollisions();
                   Slots:
                   (-)void move();
QGraphicsPixmapItem
                    Barquitoenemigo : public QObject, public
                    QGraphicsPixmapItem
                    (-)QTimer *moveTimer;
```

(-)double velocidad; (-)double aceleracion; (-)double tiempoTotal; (-)void checkCollisions(); (+)Barquitoenemigo(QGraphicsItem *parent = nullptr); slots: (+)void move();

QObject

```
QObject
QGraphicsRectItem
```

```
EnemigoDisp : public QObject, public
QGraphicsRectItem

(-)QTimer *timer;
(-)double velocidad;
(-)double angulo;
(-)double radianes;
(-)double vx0;
(-)double vy0;
(-)double by0;
(-)double g;
(-)double t;
(-)double t;
(-)double tiempoTotal;
(+)EnemigoDisp(QGraphicsItem *parent = nullptr);
slots:
(+)void move()
```

QObject QGraphicsPixmapItem

```
Disp : public QObject,public QGraphicsPixmapItem

(-)double velocidad;
(-)double angulo;
(-)double radianes;
(-)double vx0;
(-)double vy0;
(-)double g;
(-)double t;
(-)double tiempoTotal;
(-)QTimer *timer;
(-)Puntaje *puntajeDisplay;

(+)Disp(Puntaje *puntaje,QGraphicsItem
*parent=nullptr);
Slots:
(+)void move();
```

QGraphicsScene

```
Escena: public QGraphicsScene
```

```
(+)Escena(QObject *parent = nullptr);
```

QObject

GameController : public QObject

```
(-)QGraphicsScene *scene;
(-)Puntaje *puntaje;
(-)QTimer *smallEnemyTimer;
(-)QTimer *largeEnemyTimer;
(-)QTimer *puntajeCheckTimer;
(-)QTimer *bolaTimer;
(-)bool jefeFinalSpawned;
(-)bool gameOver;
(+)GameController(QGraphicsScene
*scene, Puntaje *puntaje, QObject
*parent = nullptr);
(+)void start();
(+)void clearScene();
signals:
(+)void changeScene(QGraphicsScene
*newScene);
slots:
(-)void spawnSmallEnemy();
(-)void spawnLargeEnemy();
(-)void checkPuntaje();
(-)void spawnBola();
(-)void onJefeFinalDerrotado();
```

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 ProyectilEnemigo:public QObject,public QGraphicsPixmapItem (+)ProyectilEnemigo(QGraphicsItem *parent=nullptr); slots: (+)void move();

JefeProyectil : public ProyectilEnemigo

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

QMainWindow

```
MainWindow : public QMainWindow

(-)Ui::MainWindow *ui;
Escena *scene;
(-)Barquito *barco;
(-)Barquitoenemigo *barquitoenemigo;
(-)QTimer *spawnTimer;
(-)JefeFinal *jefefinal;
(-)GameController *controller;
(-)GameControllerNivel2 *controllerNivel2;
(+)MainWindow(QWidget *parent = nullptr);
(+)~MainWindow();
(-)void iniciarJuego();
slots:
(-)void onChangeScene(QGraphicsScene
*newScene);
```

QObject QGraphicsPixmapItem

```
PowerUp : public QObject, public QGraphicsPixmapItem
```

(-)QTimer *collisionTimer;

```
(+)PowerUp(QGraphicsItem *parent =
nullptr);
```

- (+)virtual void applyEffect() = 0;
 slots:
- (-)void checkCollision();

```
PowerUpDano : public PowerUp
```

- (+)PowerUpDano(QGraphicsItem *parent
 = nullptr);
- (+)void applyEffect() override;

PowerUpVelocidad : public PowerUp

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

(+)void applyEffect() override;

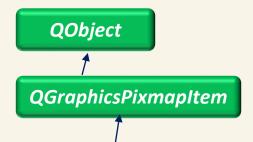
PowerUpVida : public PowerUp

PowerUpVida(QGraphicsItem *parent =
nullptr);

void applyEffect() override;

Polvora : public PowerUp

- (+)Polvora(QGraphicsItem *parent =
 nullptr);
- (+)void applyEffect() override;
- (+)void checkCollision();
- signals:
- (+)void polvoraRecolectada();



Muro:public QGraphicsPixmapItem

(+)Muro(qreal x, qreal y, const
QString& rutaImagen);

QGraphicsEllipseItem

Muro_verdadero: public QGraphicsEllipseItem

(+)Muro_verdadero(qreal x, qreal y,
qreal ancho, qreal alto);

QGraphicsScene

Nivel2 : public QGraphicsScene

Nivel2(QObject *parent = nullptr);

QGraphicsTextItem

PolvoraDisplay : public QGraphicsTextItem

(-)int polvora;

- (+)PolvoraDisplay(QGraphicsItem *parent =
 nullptr);
- (+)void aumentarPolvora(int cantidad);
- (+)int getPolvora() const;

QObject QGraphicsPixmapItem Proyectil: public QObject, public QGraphicsPixmapItem (-)QTimer *moveTimer; (-)qreal angle; (-)qreal initialSpeed; (-)qreal currentSpeed; (-)qreal deceleration; (-)QElapsedTimer timer; (+)Proyectil(greal angle, QGraphicsItem *parent = nullptr); slots: (-)void move(); **QObject QGraphicsPixmapItem** ProyectilEnemigoCanon : public QObject, public QGraphicsPixmapItem (+)explicit ProyectilEnemigoCanon(QGraphicsItem *parent = nullptr); slots: (+)void move2(); **QGraphicsPixmapItem** Puerta : public QObject, public QGraphicsPixmapItem (-)int vida;

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

QObject

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();

QObject QGraphicsPixmapItem

```
soldado_bizantino : public QObject,public
QGraphicsPixmapItem

(-)QTimer *movementTimer;
(-)QGraphicsItem *player;
(-)int velocidad;
(-)int vida;

(+)explicit soldado_bizantino(QGraphicsItem *parent = nullptr);
(+)void reducirVida(int cantidad);
(+)void dropPowerUp();
(-)void updateImage();
Slots:
(+)void followPlayer();
signals:
(-)void soldadoMuerto(QGraphicsItem *item);
```

QObject QGraphicsPixmapItem

```
soldado_otomano: public QObject, public QGraphicsPixmapItem
(-)int vidas;
(-)int velocidad;
(-)int velocidaddano;
(-)qreal currentAngle;
(-)Puntaje* puntajeDisplay = nullptr;
(-)bool canShoot;
(-)QTimer shootTimer;
(+)soldado_otomano();
(+)void keyPressEvent(QKeyEvent *event) override;
(+)void reducirVida();
(+)int getdano();
(+)void aumentarVida();
(+)void aumentarVelocidad();
(+)void aumentarDano();
Slots:
(-)void enableShooting();
Signals:
void vidaCambiada(int nuevaVida);
```

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();
```