**Objetos:**

class player : public QGraphicsItem

**Atributos:**

-float x, y, vx, vy, ax, ay;

posición, velocidad y aceleración para calcular las fisicas de movimiento

-double salud;

Salud del jugador

-int armaID;

Arma que posee el jugador

-QTimer \*timer;

Timer para controlar la animación del objeto

**Métodos:**

-QRectF boundingRect() const;

Tamaño del objeto

-void paint(QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget);

dibuja el objeto

-void animacion();

Controla la animación del objeto en pantalla

-void aceleracion();

Cambia la aceleración del objeto de acuerdo a la interacción del jugador

-void velocidades();

Calcula la velocidad del objeto

-void posiciones();

Calcula la posición del objeto

-void muerte();

Comprueba si el jugador no tiene salud y en tal caso lo hace reaparecer

-void modSalud();

Modifica la salud del jugador

-void interaccion();

Controla la interacción del jugador con objetos del escenario

-void atacar();

Realiza la acción de atacar, si es un arma de proyectiles dispara

-void lanzar();

Realiza el lanzamiento de un arma

class enemy : public QGraphicsItem

**Atributos:**

-float x, y, vx, vy, ax, ay;

posición, velocidad y aceleración para calcular las fisicas de movimiento

-double salud

Salud del enemigo

-QTimer \*timer;

Timer para controlar la animación del objeto

**Métodos:**

-QRectF boundingRect() const;

Tamaño del objeto

-void paint(QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget);

dibuja el objeto

-void animacion();

Controla la animación del objeto en pantalla

-void aceleracion();

Cambia la aceleración del objeto

-void velocidades();

Calcula la velocidad del objeto

-void posiciones();

Calcula la posición del objeto

-void muerte();

Comprueba si el enemigo no tiene salud y en tal caso lo elimina de la escena

-void modSalud();

Modifica la salud del enemigo

-void atacar();

Realiza la acción de atacar

Class alienmelee : public enemy

**Atributos:**

**Métodos:**

-void atacar();

Realiza la acción de atacar

-void paint(QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget);

dibuja el objeto

Class aliengun : public enemy

**Atributos:**

**Métodos:**

-void atacar();

Realiza la acción de disparar

-void paint(QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget);

dibuja el objeto

Class alienarm: public enemy

**Atributos:**

**Métodos:**

-void atacar();

Realiza la acción de lanzar el brazo

-void paint(QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget);

dibuja el objeto

Class aliensexplosive : public enemy

**Atributos:**

**Métodos:**

-void atacar();

Realiza la acción de atacar

-void paint(QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget);

dibuja el objeto

Class alienoctopus: public enemy

**Atributos:**

**Métodos:**

-void atacar();

Realiza la acción de atacar

-void paint(QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget);

dibuja el objeto

class weapon : public QGraphicsItem

**Atributos:**

-float x, y, vx, vy, ax, ay;

posición, velocidad y aceleración para calcular las fisicas de movimiento

-float daño;

Daño que proporciona el arma

-int ID;

Id del arma

-bool usedbyplayer;

Indica si está siendo usada por el jugador

**Métodos:**

-void aceleracion();

Cambia la aceleración del objeto de acuerdo a la interacción del jugador

-void velocidades();

Calcula la velocidad del objeto

-void posiciones();

Calcula la posición del objeto, si la posee el jugador seguirá la posición del jugador

class projectile : public QGraphicsItem

**Atributos:**

-float x, y, vx, vy, ax, ay;

posición, velocidad y aceleración para calcular las fisicas de movimiento

**Métodos:**

-void aceleracion();

Cambia la aceleración del objeto de acuerdo a la interacción del jugador

-void velocidades();

Calcula la velocidad del objeto

-void posiciones();

Calcula la posición del objeto