

Modelamiento de Objetos

Santiago Pereira Ramirez

Departamento de Ingeniería Electrónica y
Telecomunicaciones
Universidad de Antioquia
Medellín
Octubre de 2021

Índice

1. Modelamiento de objetos	2
----------------------------	---

1. Modelamiento de objetos

//La clase MainCharacter sera el diseño del personaje principal, este tendra atributos privados, asi mismo metodos publicos .Ademas tendra herencia.La clase MainCharacter es fundamental ya que nos permitira interactuar con los enemigos y obtaculos en la escena.

```
class MainCharacter:public QObject,public QGraphicsItem
{
private :

    int alto;
    int ancho;
    double Posicionx,Posiciony;
    double Velocidadx,Velocidady;
    double Aceleracionx,Aceleraciony;
    double delta;
public:

    MainCharacter(int An, int Al);
    MainCharacter(int An, int Al,double Posx,double Posy,double Velx,double
Vely,double Acex,double Acey);
    QRectF boundingRect();
    void paint();
    void MoverDerecha();
    void MoverIzquierda();
    void MoverArriba();
    void MoverArribaPlataforma();

    public slots:

    void Calcular();
}
```

//Esta es la clase enemigo con atributos privados y metodos publicos, se generara el movimiento del enemigo apartir de la propia clase, este podra disparar a el personaje principal para que este no cumpla su objetivo.

```
class EnemigoInteligente:public QGraphicsItem.
{
private :

    double Posicionx;
    double Posiciony;
    double Posicionx2;
    double Posiciony2;
```

```

    double ancho;
    double alto;
    int Op;
    int LiS,LiI,Incremento;
    int CambioDerecha;
    int CambioLado;
public:

    EnemigoInteligente(float posx,float posy,int LimitInf, int LimitSup);
    QRectF boundingRect();
    void paint();
    void Disparo();
public slots:

    void move();
}

```

//La clase proyectil con atributos privados y metodos publicos, se podra modelar un proyectil con movimiento fisico caracteristico esto con el fin de destruir enemigos o el personaje principal.

```

class Proyectil:public QObject,public QGraphicsItem
{
    private :

    double velocidadinicial;
    double velocidadx;
    double velocidady;
    double angulo;
    double posicionx;
    double posicony;
    double delta;
    double gravedad;
    int radio;
public:

    Proyectil();
    Proyectil(double x,double y,double v,double a);
    QRectF boundingRect();
    void paint();
    void actualizarposicion();
    void calcularvelocidad();
public slots:

```

```

    void move();
}

```

//La clase Vista sera la encargada de mostrar el status del personaje. Recibirá a partir del personaje tanto el puntaje que vaya adquiriendo o la vida que pierda.

```

class Vista: public QGraphicsTextItem
{
public :

    Vista();
    void AumentarVida();
    void RestarVida();
    void AumentarPuntaje();
    void RestarPuntaje();
private:

    int Vida;
    int Puntaje;
}

```

//La clase escena con atributos privados, metodos publico,slots publicos, podremos mostrar los personajes y obstaculos que nos permitiran crear un juego. Así mismo interactuarán cada uno de los objetos creados por medio de sistemas fisicos.

```

class MainWindow :public QMainWindow
{
private :

    Ui::MainWindow *ui;
    QGraphicsScene *ventana;
    EnemigoInteligente *Smart;
    Proyectil *ball;
    MainCharacter *player;
    Vista *score;
    int Nivel;
public:

    MainWindow(QWidget *parent = nullptr);
    MainWindow();
    void keyPressEvent();
    void Niveles();
public slots:

```

```
void spawn();  
void disparar();  
}
```