

# Modelamiento de Objetos

Proyecto final

Santiago Pereira Ramirez

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

# Índice

1. Introduccion	2
2. Modelamiento de objetos	2
3. Desarrollo	5

## 1. Introduccion

A continuacion se declararán los objetos que habra en el videojuego, esto con el fin de tener un vista a lo que se debera de implementar y tener una idea amplia de que elementos tendra cada clase, tambien se mostrara como los objetos interactuarian unos con los otros. igualmente se explicara como los usuarios podran desarrollar el videojuego para llegar al objetivo final.

## 2. 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 permitira al usuario una interaccion con los enemigos y obtaculos en la escena.

```
class MainCharacter: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();

}
```

–La clase enemigo con atributos privados y metodos publicos, se encargara de crear el enemigo u obstaculo, este podra disparar a el personaje principal para que este no cumpla su objetivo. Tenga en cuenta que seran varios tipos de enemigos por ende cada uno sera una clase diferente y que ademas de los atributos privados o metodos publicos que se encuentran en la clase Enemigo inteligente habran muchos mas atributos y metodos, ya que cada enemigo tendra

diferente movimiento ya sea gravitacional, amortiguado, armonico simple, etc.

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

        double Posicionx;
        double Posiciony;
        double ancho;
        double alto;
        double Amplitud;
        double ConstanteAmoriguador;
        int CambioDerecha;
        int CambioLado;
    public:

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

}
```

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

```
class Proyectil:public QGraphicsItem
{
    private :

        double velocidadinicial;
        double velocidadx;
        double velocidady;
        double angulo;
        double posicionx;
        double posiciony;
        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. Recibira 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 crear la interfaz grafica que nos permitira mostrar los personajes y obstaculos, asi mismo sera donde interactuarán cada uno de los objetos creados por medio de sistemas fisicos. Tambien habra un apartado donde se pondra cada uno de los niveles que el usuario o usuarios podran jugar.

```

class Escena :public QMainWindow
{
private :

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

        MainWindow();
        MainWindow();

```

```
void keyPressEvent();  
void Niveles();  
public slots:  
  
void spawn();  
}
```

### **3. Desarrollo**

Para el desarrollo del juego el usuario en cada nivel debera de esquivar los diferentes obstaculos, destruir a los enemigos y derrotar al jefe, de este ultimo podra obtener el nucleo y avanzar hasta ganar el juego. El jugador tendra cierta cantidad de vida, y se vera reducida cada vez que un enemigo lo golpee, el juego acabará cuando la vida del jugador llegue a cero. En caso de que sean dos jugadores, si almenos uno muere el juego acaba.