JEFFREY EL PURGADOR

Jeffrey ha decidido acabar con todos los corruptos de este planeta, infectandolos con un virus letal el cual el tiene en su sangre siendo inmune a este, pero no será una tarea fácil ya que deberá evitar a toda costa los militares que defienden al corrupto.

1. Descripción y objetivo:

* El juego es un plataformero, tiene vista lateral.
* Se podrá saltar, empujar a los militares y un botón para toser así infectando al político(se debe estar a un cierto rango del político para infectarlo).
* En el escenario habrá un político el cual será el objetivo a infectar, en ciertas partes del mapa habrá diferentes militares que atacan al jugador apenas lo vean(el jugador tiene la posibilidad de empujarlos al vacío, eliminandolos, para así reducir la cantidad de militares en el escenario).
* Tendrá un tiempo límite para infectar a cada político, el tiempo se reinicia cuando se infecta un político.
* Cada vez que se infecte un político aparecerá otro en algún lugar random del escenario, en consecuencia los militares se vuelven más agresivos por cada político infectado, además por cada cierta cantidad de políticos infectados se agrega un militar al escenario.

El objetivo es ganar la mayor cantidad de puntos, por cada político infectado se gana 2 puntos, por cada guardia enviado al vacío se ganan 3, el jugador solo tendrá una vida.

1. Visualización simplificada:



1. Clases:

* MAINWINDOW:

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

#include <QGraphicsScene>

#include <QKeyEvent>

#include <QTimer>

#include <QGraphicsRectItem>

#include <QDebug>

#include "jeffrey.h"

#include "suelo.h"

#include "bala.h"

#include "soldado.h"

#include "politico.h"

QT\_BEGIN\_NAMESPACE

namespace Ui { class MainWindow; }

QT\_END\_NAMESPACE

class MainWindow : public QMainWindow

{

Q\_OBJECT

public:

MainWindow(QWidget \*parent = nullptr);

~MainWindow();

bool col\_est();

void crear\_orilla();

void keyPressEvent(QKeyEvent \*event);

void caida();

private slots:

void bala\_mov();

void mov\_sold();

private:

Ui::MainWindow \*ui;

QGraphicsScene \*scene;

suelo \*piso;

jeffrey \*jugador;

int vel;

int vel\_auto;

QList<suelo\*> orilla;

QList<suelo\*> lista\_piso;

QList<bala\*> balas;

QTimer \*timer\_sold;

QTimer \*timer\_bala;

};

#endif // MAINWINDOW\_H

* BALA:

#ifndef BALA\_H

#define BALA\_H

#include <QPainter>

#include <QGraphicsItem>

class bala: public QGraphicsItem

{

private:

int x;

int y;

int radio;

public:

bala(int rad);

~bala();

int getX() const;

void setX(int newX);

int getY() const;

void setY(int newY);

void posicion();

void posicion(int newX, int newY);

void disparo(int);

QRectF boundingRect() const;

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

};

#endif // BALA\_H

* JEFFREY:

#ifndef JEFFREY\_H

#define JEFFREY\_H

#include <QObject>

#include <QGraphicsPixmapItem>

#include <QGraphicsItem>

class jeffrey : public QGraphicsPixmapItem {

public:

jeffrey(QGraphicsItem \*carr = nullptr);

void setScale(qreal scale);

void posicion();

void posicion(int newX, int newY);

int getX() const;

void setX(int value);

int getY() const;

void setY(int value);

QRectF boundingRect() const override;

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

void empujar();

private slots:

void tos();

private:

int x,y;

};

#endif // JEFFREY\_H

* POLITICO:

#ifndef POLITICO\_H

#define POLITICO\_H

#include <QObject>

#include <QGraphicsPixmapItem>

#include <QGraphicsItem>

class politico : public QGraphicsPixmapItem {

public:

politico(QGraphicsItem \*carr = nullptr);

void setScale(qreal scale);

void posicion();

void posicion(int newX, int newY);

int getX() const;

void setX(int value);

int getY() const;

void setY(int value);

QRectF boundingRect() const override;

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

private:

int x,y;

};

#endif // POLITICO\_H

* SOLDADO:

#ifndef SOLDADO\_H

#define SOLDADO\_H

#include <QObject>

#include <QGraphicsPixmapItem>

#include <QGraphicsItem>

class soldado : public QGraphicsPixmapItem {

public:

soldado(QGraphicsItem \*carr = nullptr);

void setScale(qreal scale);

void posicion();

void posicion(int newX, int newY);

int getX() const;

void setX(int value);

int getY() const;

void setY(int value);

QRectF boundingRect() const override;

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

void friccion();

private:

int x,y;

};

#endif // SOLDADO\_H

* SUELO:

#ifndef SUELO\_H

#define SUELO\_H

#include <QObject>

#include <QGraphicsPixmapItem>

#include <QGraphicsItem>

class suelo : public QGraphicsPixmapItem {

public:

void setScale(qreal scale);

suelo(QGraphicsItem \*carr = nullptr);

void posicion();

void posicion(int newX, int newY);

int getX() const;

void setX(int value);

int getY() const;

void setY(int value);

QRectF boundingRect() const override;

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

private:

int x,y;

};

#endif // SUELO\_H