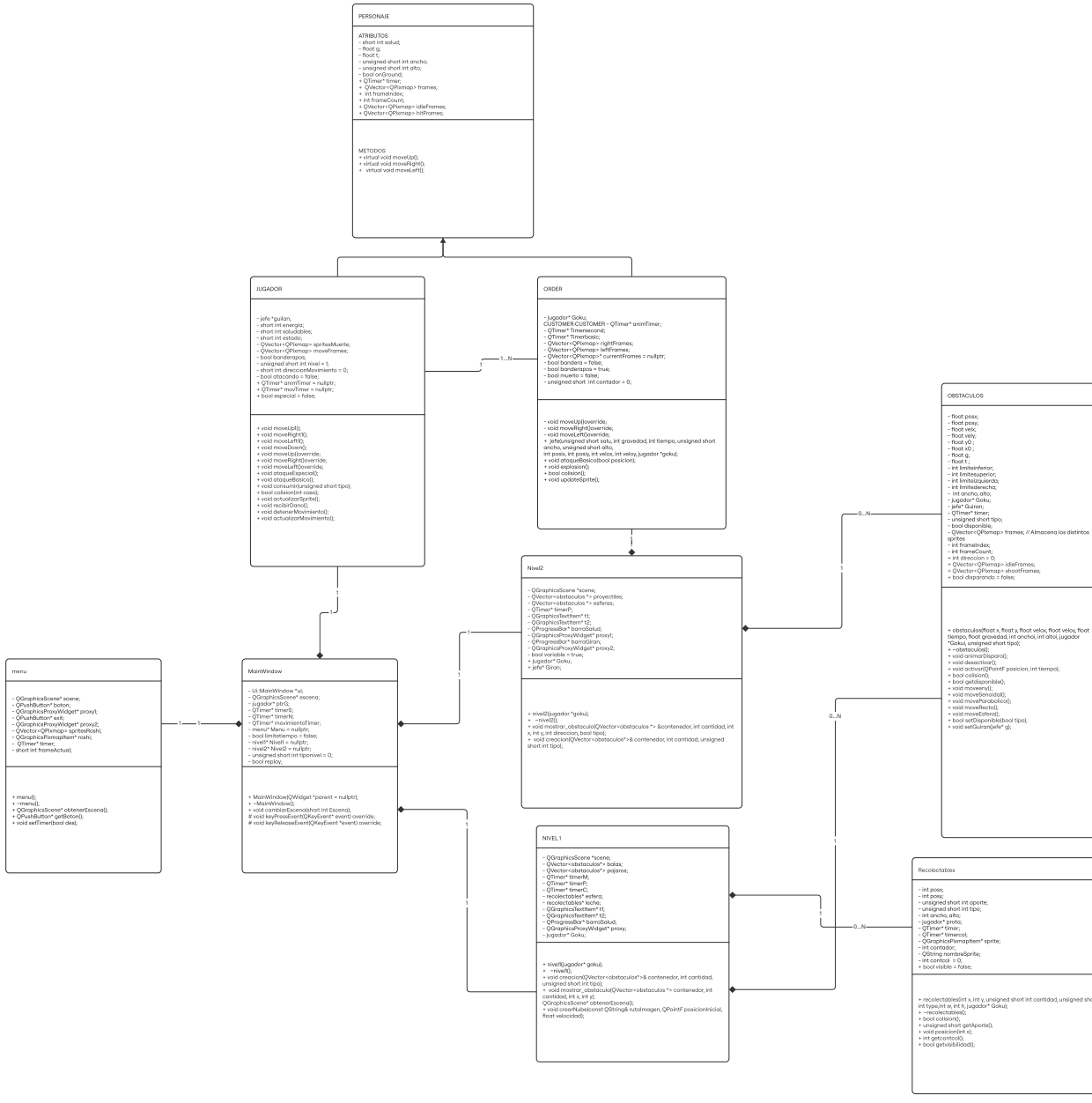
PERSONAJE
ATRIBUTOS
- short int salud;
- float g;
- float t;
- unsigned short int ancho;
- unsigned short int alto;
- bool onGround;
+ QTimer* timer;
+ QVector<QPixmap> frames;
+ int frameIndex;
+ int frameCount;
+ QVector<QPixmap> idleFrames;
+ QVector<QPixmap> hitFrames;
METODOS
+ virtual void moveUp();
+ virtual void moveRight();
+ virtual void moveLeft();
JUGADOR
- jefe *guilan;
- short int energia;
- short int soluddbles;
- short int estado;
- QVector<QPixmap> spritesMuerte;
- QVector<QPixmap> moveFrames;
- bool banderapos;
- unsigned short int nivel = 1;
- short int direccionMovimiento = 0;
- bool atacando = false;
+ QTimer* animTimer = nullptr;
+ QTimer* movTimer = nullptr;
+ bool especial = false;
+ void moveUp();
+ void moveRight();
+ void moveLeft();
+ void moveDown();
+ void moveUp()override;
+ void moveRight()override;
+ void moveLeft()override;
+ void ataqueEspecial();
+ void ataqueBasico();
+ void consumir(unsigned short tipo);
+ bool colision(int cosa);
+ void actualizarSprite();
+ void recibirDano();
+ void detenerMovimiento();
+ void actualizarMovimiento();
1...N
1
ORDER
- jugador* Goku;
CUSTOMER CUSTOMER - QTimer* animTimer;
- QTimer* Timerescena;
- QTimer* Timerbasic;
- QVector<QPixmap> rightFrames;
- QVector<QPixmap> leftFrames;
- QVector<QPixmap>* currentFrames = nullptr;
- bool bandera = false;
- bool banderapos = true;
- bool muerto = false;
- unsigned short int contador = 0;
- void moveUp()override;
- void moveRight()override;
- void moveLeft()override;
+ jefe(unsigned short saludd, int gravedad, int tiempo, unsigned short ancho, unsigned short alto, int posx, int posy, int velox, int velay, jugador *goku);
+ void ataqueBasico(bool posicion);
+ void explosion();
+ bool colision();
+ void updateSprite();
1
1
1
1
Nivel2
- QGraphicsScene *scene;
- QVector<obstaculos *> proyectiles;
- QVector<obstaculos *> esferas;
- QTimer* timerP;
- QGraphicsTextItem* t1;
- QGraphicsTextItem* t2;
- QProgressBar* barraSalud;
- QGraphicsProxyWidget* proxy1;
- QProgressBar* barraGiran;
- QGraphicsProxyWidget* proxy2;
- bool variable = true;
+ jugador* Goku;
+ jefe* Giran;
+ nivel2(jugador *goku);
+ ~nivel2();
+ void mostrar_obstaculo(QVector<obstaculos *> &contenedor, int cantidad, int x, int y, int direccion, bool tipo);
+ void creacion(QVector<obstaculos*>& contenedor, int cantidad, unsigned short int tipo);
0...N
1
OBSTACULOS
- float posx;
- float posy;
- float velx;
- float vely;
- float x0 ;
- float y0 ;
- float g;
- float t ;
- int limiteinferior;
- int limiteseuperior;
- int limiteizquierda;
- int limitederecha;
- int ancho, alto;
- jugador* Goku;
- jefe* Giran;
- QTimer* timer;
- unsigned short tipo;
- bool disponible;
- QVector<QPixmap> frames; // Almacena los distintos sprites
- int frameIndex;
- int frameCount;
+ int direccion = 0;
+ QVector<QPixmap> idleFrames;
+ QVector<QPixmap> shootFrames;
+ bool disparando = false;
+ obstaculos(float x, float y, float velox, float velay, float tiempo, float gravedad, int ancho, int alto, jugador *Goku, unsigned short tipo);
+ ~obstaculos();
+ void animarDisparo();
+ void desactivar();
+ void activar(QPointF posicion, int tiempo);
+ bool colision();
+ bool getdisponible();
+ void movenx();
+ void moveSenoidal();
+ void movePaboloide();
+ void movellecto();
+ void moveEsfera();
+ bool setDisponible(bool tipo);
+ void setGuirang(jefe* g);
menu
- QGraphicsScene* scene;
- QPushButton* boton;
- QGraphicsProxyWidget* proxy1;
- QPushButton* exit;
- QGraphicsProxyWidget* proxy2;
- QVector<QPixmap> spritesRoshi;
- QGraphicsPixmapItem* roshi;
- QTimer* timer;
- short int frameActual;
+ menu();
+ ~menu();
+ QGraphicsScene* obtenerEscena();
+ QPushButton* getBoton();
+ void setTimer(bool dea);
1
1
MainWindow
- Ui::MainWindow *ui;
- QGraphicsScene* escena;
- jugador* pirG;
- QTimer* timerS;
- QTimer* timerN;
- QTimer* movimientoTimer;
- menu* Menu = nullptr;
- bool limitetiempo = false;
- nivel1* Nivel1 = nullptr;
- nivel2* Nivel2 = nullptr;
- unsigned short int tipoNivel = 0;
- bool replay;
+ MainWindow(QWidget *parent = nullptr);
+ ~MainWindow();
+ void cambiarEscena(short int Escena);
# void keyPressEvent(QKeyEvent* event) override;
# void keyReleaseEvent(QKeyEvent *event) override;
1
1
1
1
NIVEL 1
- QGraphicsScene *scene;
- QVector<obstaculos*> balas;
- QVector<obstaculos*> pajaros;
- QTimer* timerM;
- QTimer* timerP;
- QTimer* timerC;
- recolectables* esfera;
- recolectables* leche;
- QGraphicsTextItem* t1;
- QGraphicsTextItem* t2;
- QProgressBar* barraSalud;
- QGraphicsProxyWidget* proxy;
- jugador* Goku;
+ nivel1(jugador* goku);
+ ~nivel1();
+ void creacion(QVector<obstaculos*>& contenedor, int cantidad, unsigned short int tipo);
+ void mostrar_obstaculo(QVector<obstaculos *> contenedor, int cantidad, int x, int y);
QGraphicsScene* obtenerEscena();
+ void crearNube(const QString& rutaImagen, QPointF posicionInicial, float velocidad);
1
1
0...N
Recolectables
- int posx;
- int posy;
- unsigned short int aporte;
- unsigned short int tipo;
- int ancho, alto;
- jugador* prota;
- QTimer* timer;
- QTimer* timerrock;
- QGraphicsPixmapItem* sprite;
- int contador;
- QString nombreSprite;
- int control = 0;
+ bool visible = false;
+ recolectables(int x, int y, unsigned short int cantidad, unsigned short int type,int w, int h, jugador* Goku);
+ ~recolectables();
+ bool colision();
+ unsigned short getAporte();
+ void posicion(int a);
+ int getcontrol();
+ bool getvisibilidad();