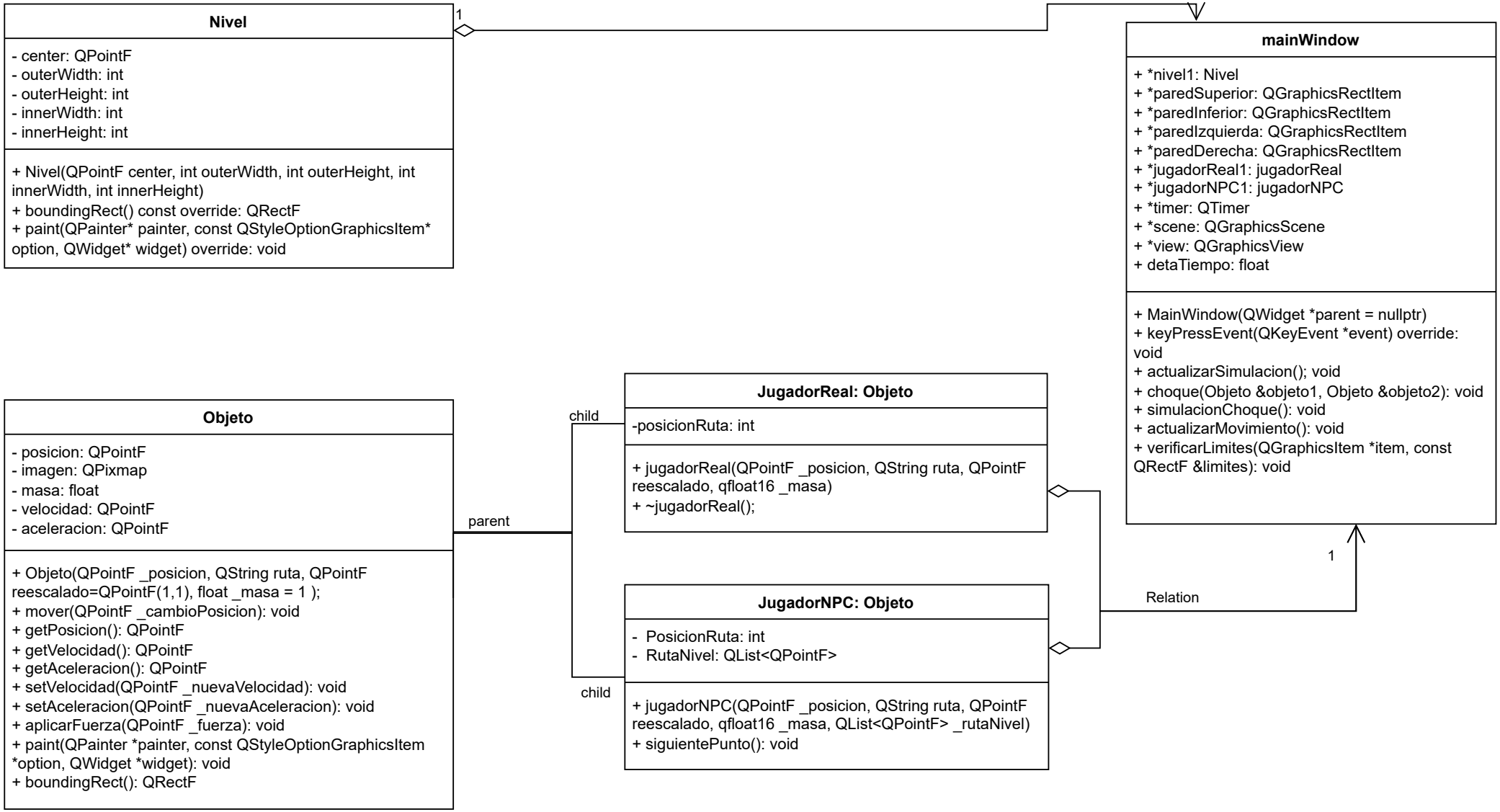
Nivel
- center: QPointF
- outerWidth: int
- outerHeight: int
- innerWidth: int
- innerHeight: int
+ Nivel(QPointF center, int outerWidth, int outerHeight, int innerWidth, int innerHeight)
+ boundingRect() const override: QRectF
+ paint(QPainter* painter, const QStyleOptionGraphicsItem* option, QWidget* widget) override: void
1
mainWindow
+ *nivel1: Nivel
+ *paredSuperior: QGraphicsRectItem
+ *paredInferior: QGraphicsRectItem
+ *paredIzquierda: QGraphicsRectItem
+ *paredDerecha: QGraphicsRectItem
+ *jugadorReal1: jugadorReal
+ *jugadorNPC1: jugadorNPC
+ *timer: QTimer
+ *scene: QGraphicsScene
+ *view: QGraphicsView
+ detaTiempo: float
+ MainWindow(QWidget *parent = nullptr)
+ keyPressEvent(QKeyEvent *event) override: void
+ actualizarSimulacion(); void
+ choque(Objeto &objeto1, Objeto &objeto2): void
+ simulacionChoque(): void
+ actualizarMovimiento(): void
+ verificarLimites(QGraphicsItem *item, const QRectF &limites): void
Objeto
- posicion: QPointF
- imagen: QPixmap
- masa: float
- velocidad: QPointF
- aceleracion: QPointF
+ Objeto(QPointF _posicion, QString ruta, QPointF reescalado=QPointF(1,1), float _masa = 1);
+ mover(QPointF _cambioPosicion): void
+ getPosicion(): QPointF
+ getVelocidad(): QPointF
+ getAceleracion(): QPointF
+ setVelocidad(QPointF _nuevaVelocidad): void
+ setAceleracion(QPointF _nuevaAceleracion): void
+ aplicarFuerza(QPointF _fuerza): void
+ paint(QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget): void
+ boundingRect(): QRectF
parent
child
child
JugadorReal: Objeto
-posicionRuta: int
+ jugadorReal(QPointF _posicion, QString ruta, QPointF reescalado, qfloat16 _masa)
+ ~jugadorReal();
JugadorNPC: Objeto
- PosicionRuta: int
- RutaNivel: QList<QPointF>
+ jugadorNPC(QPointF _posicion, QString ruta, QPointF reescalado, qfloat16 _masa, QList<QPointF> _rutaNivel)
+ siguientePunto(): void
1
Relation