

## Clase Personaje

Atributos:

- # QString ultimaDireccionHorizontal
- # QPointF posicionInicialQuieto
- # int kiActual
- # int kiMaximo
- # QString nombre
- # QString carpetaSprites
- # int vida
- # int vidaMaxima
- # qreal velocidadMovimiento
- # qreal escalaSprite
- # bool animacionEntradaActiva
- # int frameEntradaActual
- # QTimer\* timerEntrada
- # int frameActual
- # int frameMaximo
- # QTimer \*animacionTimer
- # bool moviendose
- # QString baseFase
- # bool saltando
- # qreal alturaOriginal
- # qreal velln
- # qreal tiempo
- # QTimer \*jumpTimer
- # int frameSaltoActual
- # qreal alturaMaximaAlcanzada
- # qreal velocidadVertical
- # qreal aceleracionVertical
- # qreal velocidadHorizontalSalto
- # qreal coeficienteResistencia
- # qreal masaPersonaje
- # qreal deltaT
- # qreal posicionXSalto
- # qreal velocidadHorizontal
- # qreal offsetYSalto
- # qreal posicionBaseSalto
- # qreal hitboxAncho
- # qreal hitboxAlto
- # qreal hitboxOffsetX
- # qreal hitboxOffsetY
- # qreal hitboxSaltoAncho
- # qreal hitboxSaltoAlto
- # qreal hitboxSaltoOffsetX
- # qreal hitboxSaltoOffsetY

- # qreal hitboxNormalAncho
- # qreal hitboxNormalAlto
- # qreal hitboxNormalOffsetX
- # qreal hitboxNormalOffsetY
- # QRectF limitesEscena
- # bool hitboxVisible
- # QGraphicsRectItem\* hitboxVisual
- # QList<QPointF> trayectoriaSalto
- # int maxPuntosTrayectoriaSalto

#### Métodos:

- + explicit Personaje(QObject \*parent = nullptr)
- + virtual ~Personaje()
- + virtual void moverDerecha()
- + virtual void moverIzquierda()
- + virtual void moverArriba()
- + virtual void moverAbajo()
- + virtual void atacar()
- + virtual void recibirDanio(int danio)
- + virtual void saltar()
- + virtual void cambiarSprite(const QString& direccion)
- + virtual void cambiarSpriteCentrado(const QString& direccion)
- + virtual void cambiarSpriteConOffset(const QString& direccion, qreal offsetX, qreal offsetY)
- + virtual void iniciarAnimacionIdle()
- + virtual void establecerEscala(qreal escala)
- + virtual void morir()
- + virtual void iniciarAnimacionEntrada()
- + virtual void kiCambiado(int kiActual, int kiMaximo)
- + bool estaEnAnimacionEntrada() const
- + void paint(QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget = nullptr) override
- + void establecerVida(int vidaMaxima)
- + void establecerVelocidad(qreal velocidad)
- + void establecerNombre(const QString& nombre)
- + void establecerCarpetaSprites(const QString& carpeta)
- + void establecerFisicaSalto(qreal masa, qreal resistencia = 0.1)
- + void establecerVelocidadSalto(qreal velocidadInicial)
- + void establecerVelocidadHorizontalSalto(qreal velocidadH)
- + void aplicarMovimientoHorizontal(qreal deltaX)
- + void setPos(qreal x, qreal y)
- + void pausarTimerSalto()
- + void reanudarTimerSalto()
- + void establecerHitbox(qreal ancho, qreal alto, qreal offsetX = 0, qreal offsetY = 0)
- + void establecerHitboxSalto(qreal ancho, qreal alto, qreal offsetX = 0, qreal offsetY = 0)
- + void aplicarHitboxSalto()

- + void restaurarHitboxNormal()
- + QRectF obtenerHitbox() const
- + QRectF obtenerHitboxGlobal() const
- + bool colisionaCon(Personaje\* otroPersonaje) const
- + bool colisionaCon(const QRectF& rectangulo) const
- + void verificarLimitesPantalla(const QRectF& limitesEscena)
- + void establecerLimitesEscena(const QRectF& limites)
- + void mostrarHitbox(bool mostrar = true)
- + void ocultarHitbox()
- + bool estaHitboxVisible() const
- + int getVida() const
- + int getVidaMaxima() const
- + qreal getVelocidad() const
- + QString getNombre() const
- + bool estaVivo() const
- + bool estaSaltando() const
- # virtual void actualizarSalto()
- # virtual void actualizarAnimacion()
- # void actualizarVisualizacionHitbox()
- # virtual void actualizarAnimacionEntrada()
- signals: void personajeMuerto(Personaje\* personaje)
- signals: void vidaCambiada(int vidaActual, int vidaMaxima)
- signals: void personajeAtaco(Personaje\* atacante)

## Clase Piccolo

Atributos:

- - bool fase
- - QTransform transform
- - bool animacionRayoActiva
- - int frameRayoActual
- - QTimer\* timerRayo
- - bool animacionKickActiva
- - int frameKickActual
- - bool kickAlta
- - bool animacionGravityBlastActiva
- - int frameGravityBlastActual
- - QTimer\* timerGravityBlast
- - Goku\* objetivoActual

Métodos:

- + Piccolo(QObject \*parent = nullptr)
- + bool getFase() const
- + void setFase(bool newFase)
- + void alternarFase()
- + bool getkickAlta() const

- + void setkickAlta(bool newkickAlta)
- + void moverDerecha() override
- + void moverIzquierda() override
- + void moverArriba() override
- + void moverAbajo() override
- + void atacar() override
- + void recibirDanio(int danio) override
- + void morir() override
- + void iniciarAnimacionIdle() override
- + void actualizarAnimacion() override
- + void cambiarSprite(const QString& direccion) override
- + void cambiarSpriteCentrado(const QString& direccion) override
- + void iniciarAnimacionEntrada() override
- + void iniciarCargaRayo()
- + void detenerCargaRayo()
- + void lanzarRayo()
- + bool estaCargandoRayo() const
- + void iniciarCargaKick()
- + void detenerCargaKick()
- + void lanzarKick()
- + bool estaCargandoKick() const
- + void lanzarGravityBlast(Goku\* gokuTarget)
- + void iniciarCargaGravityBlast()
- + void detenerCargaGravityBlast()
- + bool estaCargandoGravityBlast() const
- + void establecerObjetivo(Goku\* objetivo)
- - void actualizarAnimacionEntrada() override
- - void actualizarAnimacionRayo()
- - void actualizarAnimacionKick()
- - void actualizarAnimacionGravityBlast()
- signals: void kiCambiado(int kiActual, int kiMaximo) override

## Clase Goku

Atributos:

- - bool animacionKiActiva
- - int frameKiActual
- - QTimer\* timerKi
- - QTimer\* timerRecargaKi
- - QPointF posicionOriginalKi
- - bool animacionKamehamehaActiva
- - int frameKamehamehaActual
- - QTimer\* timerKamehameha
- - QPointF posicionFijaKamehameha
- - bool animacionRafagaActiva
- - int frameRafagaActual
- - QTimer\* timerRafaga

- - EstadoKi estadoKiActual
- - int frameDentroEstado
- - int velocidadRecargaKi
- - qreal offsetKiX
- - qreal offsetKiY
- - int frameMuerteActual
- - QTimer\* timerMuerte
- - int frameGolpe
- - int framePatada
- - QTimer \*timerGolpe
- - QTimer \*timerPatada
- - int frameTeleport
- - QTimer \*timerTeleport
- - bool animacionTeleportActiva

#### Métodos:

- + Goku(QObject \*parent = nullptr)
- + void moverDerecha() override
- + void moverIzquierda() override
- + void moverArriba() override
- + void moverAbajo() override
- + void recibirDanio(int danio) override
- + void iniciarAnimacionIdle() override
- + void iniciarAnimacionEntrada() override
- + void iniciarRecargaKi()
- + void detenerRecargaKi()
- + bool estaRecargandoKi() const
- + void iniciarCargaKamehameha()
- + void detenerCargaKamehameha()
- + void lanzarKamehameha()
- + bool estaCargandoKamehameha() const
- + void cambiarSpriteKamehamehaFijo(const QString& direccion)
- + void iniciarAnimacionRafaga()
- + void detenerAnimacionRafaga()
- + bool estaEnAnimacionRafaga() const
- + void lanzarBlastB()
- + void establecerKi(int ki, int kiMax = 100)
- + int obtenerKi() const
- + int obtenerKiMaximo() const
- + float obtenerPorcentajeKi() const
- + void morir() override
- + void iniciarAnimacionTeleport()
- + void golpear()
- + void patear()
- + void animarGolpe()
- + void animarPatada()
- + void tp()

- - void actualizarAnimacionEntrada() override
- - void actualizarAnimacionKi()
- - void recargarKi()
- - void actualizarAnimacionKamehameha()
- - void actualizarAnimacionRafaga()
- - void animarTp()
- signals: void kiCambiado(int kiActual, int kiMaximo) override

## Clase game

Atributos:

- + Piccolo\* pic
- + Goku\* p
- + QGraphicsView \*view
- + QGraphicsScene \*scene
- - Ui::game \*ui
- - int nivelActual
- - bool piccoloDerrotado
- - QTimer\* movimientoTimer
- - QTimer\* piccoloMovTimer
- - QTimer\* piccoloIATimer
- - unsigned int cntPiccolo
- - QTransform transform
- - qreal POSICION\_ORIGINAL\_X
- - qreal POSICION\_ORIGINAL\_Y
- - bool teclaD\_presionada
- - bool teclaA\_presionada
- - bool teclaW\_presionada
- - bool teclaS\_presionada
- - bool teclaK\_presionada
- - bool teclaJ\_presionada
- - bool teclaL\_presionada
- - bool piccoloD\_presionada
- - bool piccoloA\_presionada
- - bool piccoloW\_presionada
- - bool piccoloS\_presionada
- - bool piccoloK\_presionada
- - bool piccoloJ\_presionada
- - bool piccoloL\_presionada
- - QGraphicsPixmapItem\* fondoItem
- - int fondoActual
- - QStringList fondosDisponibles
- - QGraphicsPixmapItem\* barraVida
- - QGraphicsPixmapItem\* carapersonaje
- - QGraphicsPixmapItem\* barraKi
- - QGraphicsPixmapItem\* barraVida2
- - QGraphicsPixmapItem\* carapersonaje2

- - QGraphicsPixmapItem\* barraKi2
- - QMediaPlayer\* musicPlayer
- - QAudioOutput\* audioOutput

Métodos:

- + game(QWidget \*parent = nullptr)
- + ~game()
- + void keyPressEvent(QKeyEvent \*e) override
- + void keyReleaseEvent(QKeyEvent \*e) override
- + void resizeEvent(QResizeEvent \*event) override
- + void saltoDireccional()
- slots: void actualizarBarraVida(int vidaActual, int vidaMaxima)
- slots: void actualizarBarraVida2(int vidaActual, int vidaMaxima)
- slots: void actualizarBarraKi(int kiActual, int kiMaximo)
- - void actualizarMovimiento()
- - void piccoloActualizarMovimiento()
- - void manejarDerrotaPiccolo()
- - void manejarDerrotaGoku()
- - void configurarFondo()
- - void cambiarFondo()
- - void alternarHitboxKamehameha()
- Consulta realizada correctamente

## Clase MainMenu

Atributos:

Métodos:

- + explicit MainMenu(QWidget \*parent = nullptr)

## Clase Habilidad

Atributos:

- # qreal velocidadHabilidad
- # int danoHabilidad
- # qreal alcanceMaximo
- # qreal distanciaRecorrida
- # bool activa
- # QPointF posicionInicial
- # QPointF direccionMovimiento
- # QTimer\* timerActualizacion
- # QGraphicsScene\* escenaJuego

Métodos:

- + explicit Habilidad(QObject \*parent = nullptr)

- + virtual ~Habilidad()
- + virtual void iniciar(QPointF posicionInicial, QPointF direccion) = 0
- + virtual void actualizar() = 0
- + virtual void detener() = 0
- + void establecerVelocidad(qreal velocidad)
- + void establecerDano(int dano)
- + void establecerAlcance(qreal alcance)
- + void establecerEscena(QGraphicsScene\* escena)
- + qreal obtenerVelocidad() const
- + int obtenerDano() const
- + qreal obtenerAlcance() const
- + bool estaActiva() const
- + virtual QRectF boundingRect() const override = 0
- + virtual void paint(QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget = nullptr) override = 0
- # void verificarLmites()
- signals: void habilidadTerminada(Habilidad\* habilidad)
- signals: void colisionDetectada(Habilidad\* habilidad, QGraphicsItem\* objetivo)
- # slots: virtual void actualizarFisica()

## Clase BlastB

Atributos:

- - QPixmap spriteBlastB1
- - QPixmap spriteBlastB2
- - QPixmap spriteBlastB3
- - QPixmap spriteBlastB4
- - QPixmap spriteActual
- - QTimer\* timerAnimacion
- - int frameActual
- - QPointF posicionBase
- - QPointF direccionBase
- - qreal tiempoVida
- - qreal tiempoMaximo
- - qreal velocidad
- - qreal alcance
- - qreal x
- - qreal y
- - qreal z
- - qreal dx
- - qreal dy
- - qreal dz
- - qreal sigma
- - qreal rho
- - qreal beta
- - qreal dt
- - qreal escala



- - QList<QPointF> trayectoria
- - int maxPuntosTrayectoria
- - TipoTrayectoria tipoTrayectoria
- - QRectF hitbox
- - int daño
- - bool hitboxActivo
- - QList<QGraphicsItem\*> objetosGolpeados
- - static bool mostrarHitbox
- - QString carpetaSprites

#### Métodos:

- + explicit BlastB(QObject \*parent = nullptr)
- + virtual ~BlastB()
- + void iniciar(QPointF posicionInicial, QPointF direccion) override
- + void actualizar() override
- + void detener() override
- + void destruir()
- + void crear(qreal x, qreal y, qreal dirX, qreal dirY, qreal velocidad, qreal alcance)
- + void configurarCaos(qreal sigma = 10.0, qreal rho = 28.0, qreal beta = 8.0 / 3.0)
- + void setTipoTrayectoria(TipoTrayectoria tipo)
- + void setCarpetaSprites(const QString& carpeta)
- + void verificarColisiones()
- + void actualizarHitbox()
- + QRectF obtenerHitbox() const
- + static void alternarVisualizacionHitbox()
- + QRectF boundingRect() const override
- + void paint(QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget = nullptr) override
- + void cargarSprites()
- - void actualizarAnimacion()
- - void actualizarFisica() override
- - void calcularSiguientePosicion()
- - void actualizarAtractorLorenz()
- - QPointF obtenerPosicionCaotica()

#### Clase GravityBlast

##### Atributos:

- - QPixmap spriteActual
- - QPixmap spriteGravityBlast1
- - QPixmap spriteGravityBlast2
- - QPixmap spriteGravityBlast3
- - QPixmap spriteGravityBlast4
- - QTimer\* timerAnimacion
- - int frameActual
- - Goku\* objetivoGoku

- - qreal fuerzaGravitacional
- - qreal velocidadMaxima
- - QVector2D velocidadActual
- - QVector2D direccionMovimientoQVector
- - QRectF hitbox
- - int dano
- - bool hitboxActivo
- - QList<QGraphicsItem\*> objetosGolpeados
- - qreal tiempoVida
- - qreal tiempoMaximo
- - qreal dt
- - static bool mostrarHitbox

#### Métodos:

- + explicit GravityBlast(QObject \*parent = nullptr)
- + ~GravityBlast()
- + void iniciar(QPointF posicionInicial, QPointF direccion) override
- + void actualizar() override
- + void actualizarFisica() override
- + void detener() override
- + void destruir()
- + QRectF boundingRect() const override
- + void paint(QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget = nullptr) override
- + void establecerObjetivo(Goku\* objetivo)
- + static void alternarVisualizacionHitbox()
- - void cargarSprites()
- - void actualizarAnimacion()
- - void actualizarHitbox()
- - void verificarColisiones()

#### Clase Kamehameha

##### Atributos:

- - float x
- - float y
- - float dirX
- - float dirY
- - float velocidad
- - float alcance
- - float distanciaRecorrida
- - bool activo
- - QPixmap hame1
- - QPixmap hame2
- - QPixmap hame3
- - QPixmap hameha1

- - QPixmap hameha2
- - QPixmap hameha3
- - bool usarHameha
- - bool spritesValidos
- - QTimer \*animacionTimer
- - int anchoTotal
- - int altoTotal
- - QRectF hitbox
- - bool hitboxActivo
- - QList<QGraphicsItem\*> objetosGolpeados
- - int daño
- - static bool mostrarHitbox
- - QList<QPointF> trayectoria
- - int maxPuntosTrayectoria
- - QTimer \*timer

#### Métodos:

- + explicit Kamehameha(QObject \*parent = nullptr)
- + ~Kamehameha()
- + void iniciar(QPointF posicionInicial, QPointF direccion) override
- + void detener() override
- + void actualizar() override
- + void crear(float x, float y, float dirX, float dirY, float velocidad, float alcance)
- + QRectF boundingRect() const override
- + void paint(QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget = nullptr) override
- + bool estaActivo() const
- + QRectF getHitbox() const
- + void verificarColisiones()
- + bool colisionaCon(QGraphicsItem\* item)
- + void procesarColision(QGraphicsItem\* item)
- + static void alternarVisualizacionHitbox()
- - void mover()
- - void destruir()
- - void cargarSprites()
- - void iniciarAnimacion()
- - void detenerAnimacion()
- - void alternarSprites()
- private slots: void actualizar\_timeout()

#### Clase Kick

##### Atributos:

- - QTimer\* timerAnimacion
- - int frameActual
- - QRectF hitbox

- - int dano
- - bool hitboxActivo
- - QList<QGraphicsItem\*> objetosGolpeados
- - static bool mostrarHitbox

Métodos:

- + explicit Kick(QObject \*parent = nullptr)
- + ~Kick()
- + void iniciar(QPointF posicionInicial, QPointF direccion) override
- + void actualizar() override
- + void detener() override
- + void destruir()
- + QRectF boundingRect() const override
- + void paint(QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget = nullptr) override
- + static void alternarVisualizacionHitbox()
- - void verificarColisiones()

## Clase Rayo

Atributos:

- - QTransform transform
- - float x
- - float y
- - float dirX
- - float dirY
- - float velocidad
- - float alcance
- - float distanciaRecorrida
- - bool activo
- - QTimer \*timer
- - QPixmap hame1
- - QPixmap hame2
- - QPixmap hame3
- - QPixmap hameha1
- - QPixmap hameha2
- - QPixmap hameha3
- - bool spritesValidos
- - bool usarHameha
- - QTimer \*animacionTimer
- - int contadorAnimacion
- - int anchoTotal
- - int altoTotal
- - QRectF hitbox
- - int daño
- - bool hitboxActivo

- - QList<QGraphicsItem\*> objetosGolpeados
- - static bool mostrarHitbox
- - QList<QPointF> trayectoria
- - int maxPuntosTrayectoria

#### Métodos:

- + explicit Rayo(QObject \*parent = nullptr)
- + ~Rayo()
- + void crear(float x, float y, float dirX, float dirY, float velocidad, float alcance)
- + void iniciar(QPointF posicionInicial, QPointF direccion) override
- + void detener() override
- + void mover()
- + void actualizar() override
- + void actualizar\_timeout()
- + bool estaActivo() const
- + void destruir()
- + void dibujar(QPainter \*painter)
- + QRectF boundingRect() const override
- + void paint(QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget = nullptr) override
- + void establecerPotencia(int potencia)
- + void establecerSegmentos(int segmentos)
- + QRectF getHitbox() const
- + void verificarColisiones()
- + bool colisionaCon(QGraphicsItem\* item)
- + void procesarColision(QGraphicsItem\* item)
- + static void alternarVisualizacionHitbox()
- - void cargarSprites()
- - void construirRayo()
- - void iniciarAnimacion()
- - void detenerAnimacion()
- - void alternarSprites()