This file was generated with Violet UML Editor 2.1.0. (View Source / Download Violet)

Start

+_main(args: String[])

Window

<<constructor>> Window(w: int, h: int, title: String, game: Game)

Model

Bat
BufferedImageLoader
GameObject
ObjectId
Player
Block
MovingBlock
Spikes
Flag

View

Animation
Credits
Death
SpriteSheet
Start
Texture
Victory
Window

Controller

Camera
Game
Handler
KeyInput
Music

<>
GameObject

#x: float
#y: float
#velX: float
#velY: float
#id: ObjectId
#falling: boolean
#jumping: boolean
#crounching: boolean
#attacking: boolean
#hit: boolean
#facing: int
#health: int

<<constructor>> GameObject(x: float, y: float, id: ObjectId)
+tick(object: LinkedList<GameObject>)
+render(g: Graphics)
+getBounds(): Rectangle
getAttackBounds(): Rectangle
getX(): float
getY(): float
SetX(x: float)
SetY(y: float)
getVelX(): float
getVelY(): float
SetVelX(velX: float)
SetVelY(velY: float)
+isFalling(): boolean
+setFalling(falling: boolean)
+isJumping(): boolean
+setJumping(jumping: boolean)
+getId(): ObjectId
+isCrouching(crouching: boolean)
+getCrouching(): boolean
+isAttacking(attacking: boolean)
+getAttacking(): boolean
+setFacing(facing: int)
+getHit(): boolean

Spikes

-tex: Texture
-type: int

<<constructor>> Spikes(x: float, y: float, type: int, id: ObjectId)
+render(g: Graphics)
+getBounds(): Rectangle

Block

-tex: Texture
-type: int

<<constructor>> Block(x: float, y: float, type: int, id: ObjectId)
+render(g: Graphics)
+getBounds(): Rectangle

MovingBlock

-tex: Texture
-type: int
-xstart: float

<<constructor>> MovingBlock(x: float, y: float, type: int, id: ObjectId)
+tick(object: LinkedList<GameObject>)
+render(g: Graphics)
+getAttackBounds(): Rectangle
+getFacing(): int

Bat

-width: float
-height: float
-gravity: float
-MAX_SPEED: float
-headmodtop: int
-headmodside: int
-facing: int
-handler: Handler
-tex: Texture

<<constructor>> Bat(x: float, y: float, handler: Handler, id: ObjectId)
+tick(object: LinkedList<GameObject>)
-PlayerDetect(object: LinkedList<GameObject>)
-Collision(object: LinkedList<GameObject>)
+render(g: Graphics)
+getBounds(): Rectangle
+getBoundsTop(): Rectangle
+getBoundsRight(): Rectangle
+getBoundsLeft(): Rectangle
+getAttackBounds() Rectangle

Flag

-tex: Texture
-type: int

<<constructor>> Flag(x: float, y: float, type: int, id: ObjectId)
+render(g: Graphics)
+getBounds(): Rectangle

Game

-serialVersionUID: long
-running: boolean
-thread: Thread
+WIDTH: int
+HEIGHT: int
-health: BufferedImage
-arrow: BufferedImage
-bc: BufferedImage
-arrowX: int
-arrowX1: int
-arrowY1: int
-arrowY1: int
-level: BufferedImage
-handler: Handler
-cam: Camera
-tex: Texture
-rand: Random
-menu: Credits
-victory: Victory
+state: STATE
+LEVEL: int

+STATE: enum
-init()
+start()
+run()
-tick()
+render()
+getInstance(): Texture

Death

+render(g Graphics)

Victory

+render(g Graphics)

Credits

+render(g Graphics)

Camera

-x: float
-y: float

<<constructor>> Camera(x: float, y: float)
+tick(player: GameObject)
+setX(x: float)
+getX(): float
+setY(y: float)
+getY(): float

Music

-clips: HashMap<String, Clip>
-gap: int
-mute: boolean

+init()
+load(s: String, n: String)
+play(s: String)
+play(s: String, i: int)
+stop(s: String)
+resume(s: String)
+loop(s: String)
+loop(s: String, frame: int)
+loop(s: String, start: int, end: int)
+loop(s: String, frame: int, start: int, end: int)
+setPosition(s: String, frame: int)
+getPosition(s: String)
+close(s: String)

BufferedImageLoader

image: BufferedImage

loadImage(path: String): BufferedImage

Handler

+object: LinkedList<GameObject>
-tempObject: GameObject
-cam: Camera
-level: BufferedImage

<<constructor>> Handler(cam: Camera)
+tick()
+render(g: Graphics)
-clearLevel()
+addObject(object: GameObject)
+removeObject(object: GameObject)
+LoadImageLevel(image: BufferedImage)
+resetLevel()

KeyInput

-handler: Handler

<<constructor>> KeyInput(handler: Handler)
+PlaySound(file: File)
+keyPressed(event: KeyEvent)
+keyReleased(event: KeyEvent)

ObjectId

+ObjectId: enum

Texture

#bs, pi, pil, pw, pwl, pj, pjl, pc, pcl, pa, pal, bi, pb: SpriteSheet
-block_sheet: BufferedImage
-player_idle: BufferedImage
-player_idle_left: BufferedImage
-player_walk: BufferedImage
-player_walk_left: BufferedImage
-player_jump: BufferedImage
-player_jump_left: BufferedImage
-player_crouch: BufferedImage
-player_crouch_left: BufferedImage
-player_attack: BufferedImage
-player_attack_left: BufferedImage
-player_blank: BufferedImage
-bat_idle: BufferedImage
+block: BufferedImage[]
+spikes: BufferedImage[]
+player: BufferedImage[]
+bat: BufferedImage[]

<<constructor>> Texture()
-getTextures()

Animation

-speed: int
-frames: int
-index: int
-count: int
-images: BufferedImage[]
-currentImg: BufferedImage

<<constructor>> Animation(speed: int, args: BufferedImage..)
+runAnimation()
-nextFrame()
+runAnimationOnce()
-nextFrameNR()
+setFirst()
+setCount(count: int)
+getCount(): int
+getFrames(): int
+setIndex(index: int)
+drawAnimation(g: Graphics, x: int, y: int)
+drawAnimation(g: Graphics, x: int, y: int, scaleX: int, scaleY: int)

Player

-width: float
-height: float
-headmodtop: int
-headmodside: int
-gravity: float
-MAX_SPEED: float
-hitcount: int
-handler: Handler
-cam: Camera
-iframes: int
-tex: Texture
-playerIdle: Animation
-playerIdleLeft: Animation
-playerWalk: Animation
-playerWalkLeft: Animation
-playerJump: Animation
-playerJumpLeft: Animation
-playerCrouch: Animation
-playerCrouchLeft: Animation
-playerAttack: Animation
-playerAttackLeft: Animation
-playerBlank: Animation

<<constructor>> Player(x: float, y: float, handler: Handler, cam: Camera, id: ObjectId)
-Collision(object: LinkedList<GameObject>)
+getWidth(): float
+getHeight(): float
+getHeadModTop(): int
+getHeadModSide(): int
+getPlayerAttackLeft(): Animation
+getPlayerAttack(): Animation
+getFacing(): int
+getBounds(): Rectangle
+getBoundsTop(): Rectangle
+getBoundsRight(): Rectangle
+getBoundsLeft(): Rectangle

SpriteSheet

-image: BufferedImage

<<constructor>> SpriteSheet(image: BufferedImage)
+grabImage(col: int, row: int, width: int, height: int)