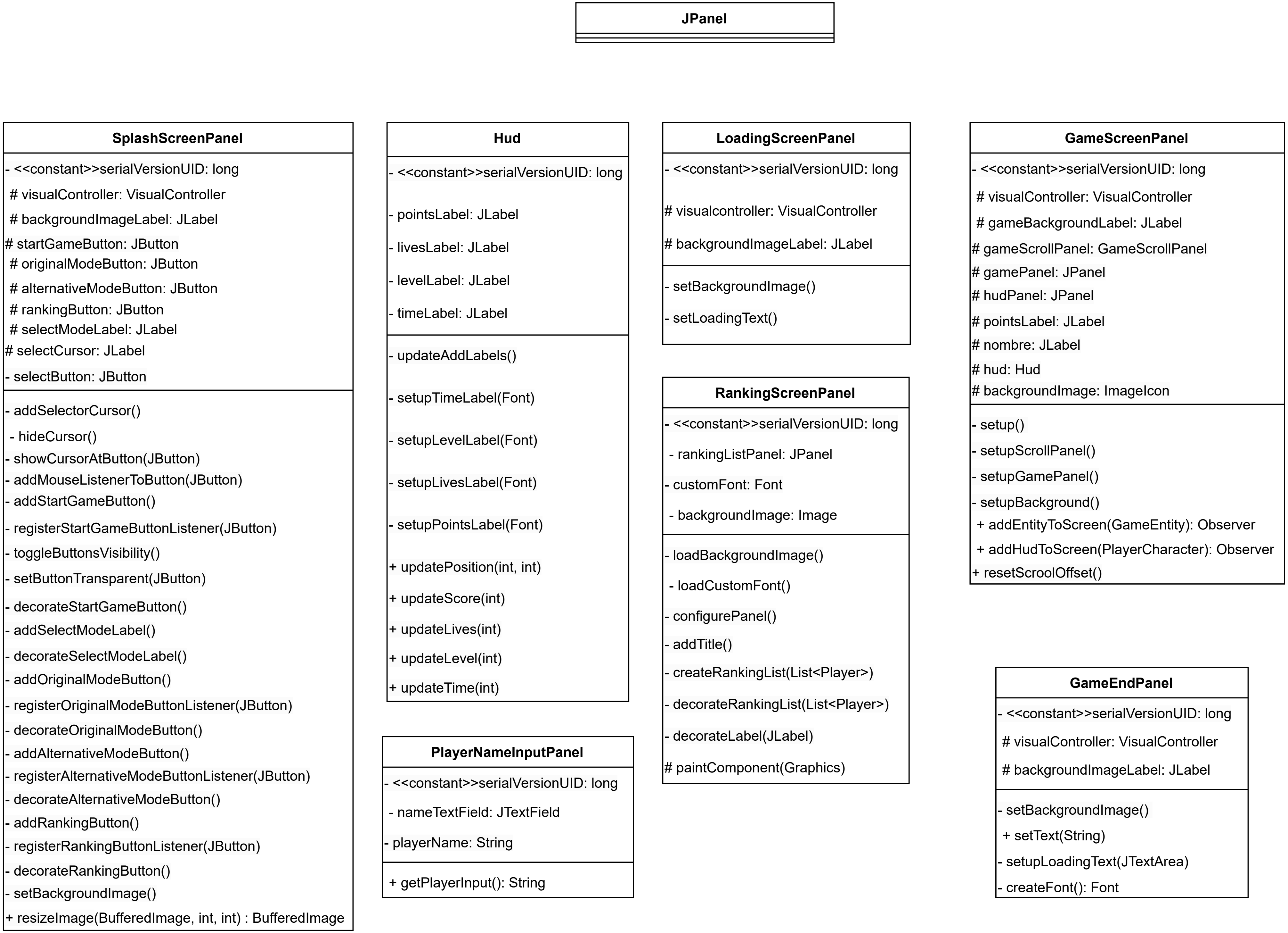


JPanel

SplashScreenPanel

- <<constant>>serialVersionUID: long
# visualController: VisualController
# backgroundImageLabel: JLabel
# startGameButton: JButton
# originalModeButton: JButton
# alternativeModeButton: JButton
# rankingButton: JButton
# selectModeLabel: JLabel
# selectCursor: JLabel
- selectButton: JButton

- addSelectorCursor()
- hideCursor()
- showCursorAtButton(JButton)
- addMouseListenerToButton(JButton)
- addStartGameButton()
- registerStartGameButtonListener(JButton)
- toggleButtonsVisibility()
- setButtonTransparent(JButton)
- decorateStartGameButton()
- addSelectModeLabel()
- decorateSelectModeLabel()
- addOriginalModeButton()
- registerOriginalModeButtonListener(JButton)
- decorateOriginalModeButton()
- addAlternativeModeButton()
- registerAlternativeModeButtonListener(JButton)
- decorateAlternativeModeButton()
- addRankingButton()
- registerRankingButtonListener(JButton)
- decorateRankingButton()
- setBackgroundImage()
+ resizeImage(BufferedImage, int, int) : BufferedImage

Hud

- <<constant>>serialVersionUID: long
- pointsLabel: JLabel
- livesLabel: JLabel
- levelLabel: JLabel
- timeLabel: JLabel

- updateAddLabels()
- setupTimeLabel(Font)
- setupLevelLabel(Font)
- setupLivesLabel(Font)
- setupPointsLabel(Font)
+ updatePosition(int, int)
+ updateScore(int)
+ updateLives(int)
+ updateLevel(int)
+ updateTime(int)

PlayerNameInputPanel

- <<constant>>serialVersionUID: long
- nameTextField: JTextField
- playerName: String

+ getPlayerInput(): String

LoadingScreenPanel

- <<constant>>serialVersionUID: long
# visualcontroller: VisualController
# backgroundImageLabel: JLabel

- setBackgroundImage()
- setLoadingText()

RankingScreenPanel

- <<constant>>serialVersionUID: long
- rankingListPanel: JPanel
- customFont: Font
- backgroundImage: Image

- loadBackgroundImage()
- loadCustomFont()
- configurePanel()
- addTitle()
- createRankingList(List<Player>)
- decorateRankingList(List<Player>)
- decorateLabel(JLabel)
# paintComponent(Graphics)

GameScreenPanel

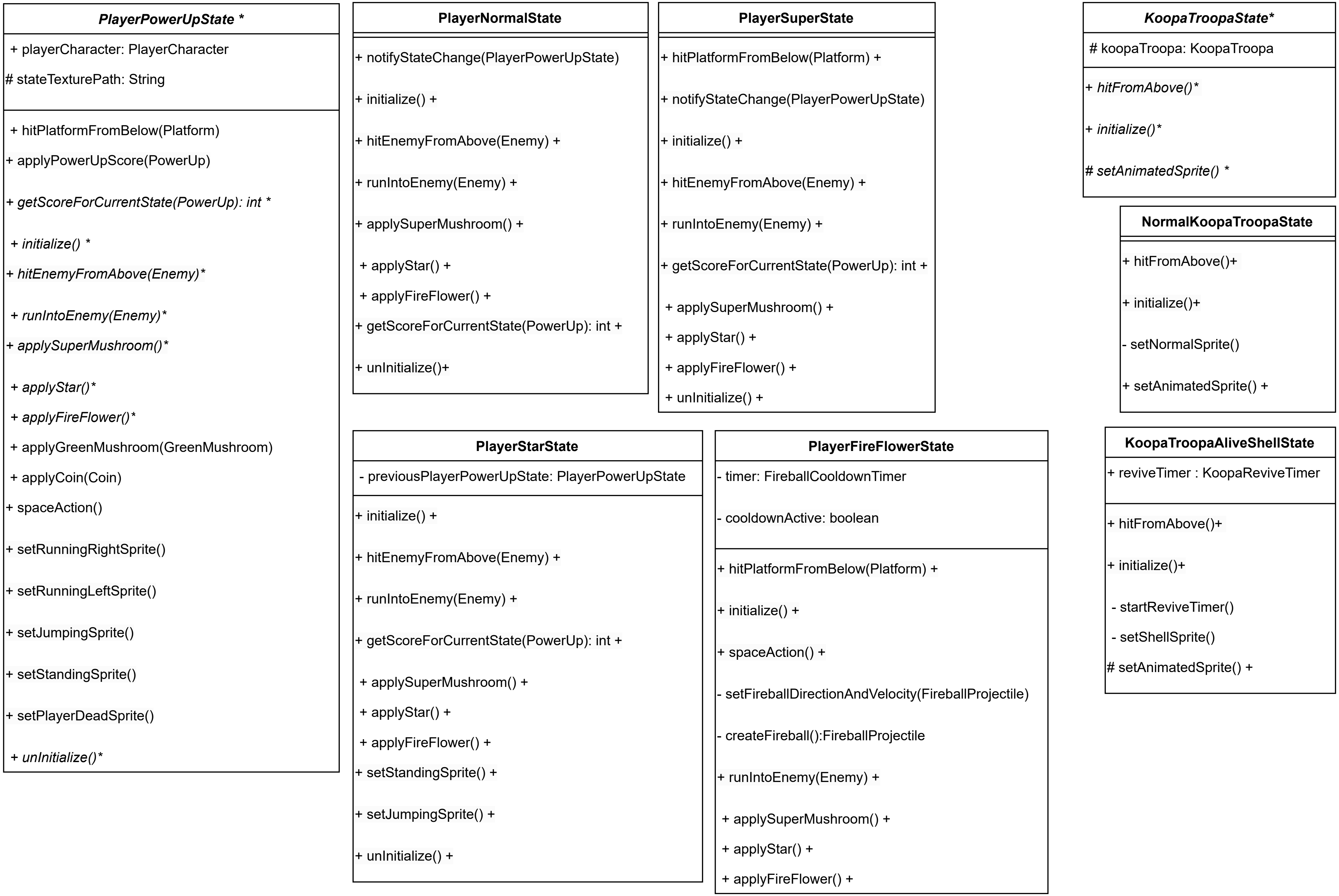
- <<constant>>serialVersionUID: long
# visualController: VisualController
# gameBackgroundLabel: JLabel
# gameScrollPanel: GameScrollPanel
# gamePanel: JPanel
# hudPanel: JPanel
# pointsLabel: JLabel
# nombre: JLabel
# hud: Hud
# backgroundImage: ImageIcon

- setup()
- setupScrollPanel()
- setupGamePanel()
- setupBackground()
+ addEntityToScreen(GameEntity): Observer
+ addHudToScreen(PlayerCharacter): Observer
+ resetScroolOffset()

GameEndPanel

- <<constant>>serialVersionUID: long
# visualController: VisualController
# backgroundImageLabel: JLabel

- setBackgroundImage()
+ setText(String)
- setupLoadingText(JTextArea)
- createFont(): Font



PlayerPowerUpState *

+ playerCharacter: PlayerCharacter
# stateTexturePath: String

+ hitPlatformFromBelow(Platform)
+ applyPowerUpScore(PowerUp)
+ _getScoreForCurrentState(PowerUp): int *_
+ _initialize()_ *
+ _hitEnemyFromAbove(Enemy)*_
+ _runIntoEnemy(Enemy)*_
+ _applySuperMushroom()*_
+ _applyStar()*_
+ _applyFireFlower()*_
+ applyGreenMushroom(GreenMushroom)
+ applyCoin(Coin)
+ spaceAction()
+ setRunningRightSprite()
+ setRunningLeftSprite()
+ setJumpingSprite()
+ setStandingSprite()
+ setPlayerDeadSprite()
+ _unInitialize()*_

PlayerNormalState

+ notifyStateChange(PlayerPowerUpState)
+ initialize() +
+ hitEnemyFromAbove(Enemy) +
+ runIntoEnemy(Enemy) +
+ applySuperMushroom() +
+ applyStar() +
+ applyFireFlower() +
+ getScoreForCurrentState(PowerUp): int +
+ unInitialize()+

PlayerSuperState

+ hitPlatformFromBelow(Platform) +
+ notifyStateChange(PlayerPowerUpState)
+ initialize() +
+ hitEnemyFromAbove(Enemy) +
+ runIntoEnemy(Enemy) +
+ getScoreForCurrentState(PowerUp): int +
+ applySuperMushroom() +
+ applyStar() +
+ applyFireFlower() +
+ unInitialize() +

KoopaTroopaState*

# koopaTroopa: KoopaTroopa

+ _hitFromAbove()*_
+ _initialize()*_
# _setAnimatedSprite()_ *

NormalKoopaTroopaState

+ hitFromAbove()+
+ initialize()+
- setNormalSprite()
+ setAnimatedSprite() +

PlayerStarState

- previousPlayerPowerUpState: PlayerPowerUpState

+ initialize() +
+ hitEnemyFromAbove(Enemy) +
+ runIntoEnemy(Enemy) +
+ getScoreForCurrentState(PowerUp): int +
+ applySuperMushroom() +
+ applyStar() +
+ applyFireFlower() +
+ setStandingSprite() +
+ setJumpingSprite() +
+ unInitialize() +

PlayerFireFlowerState

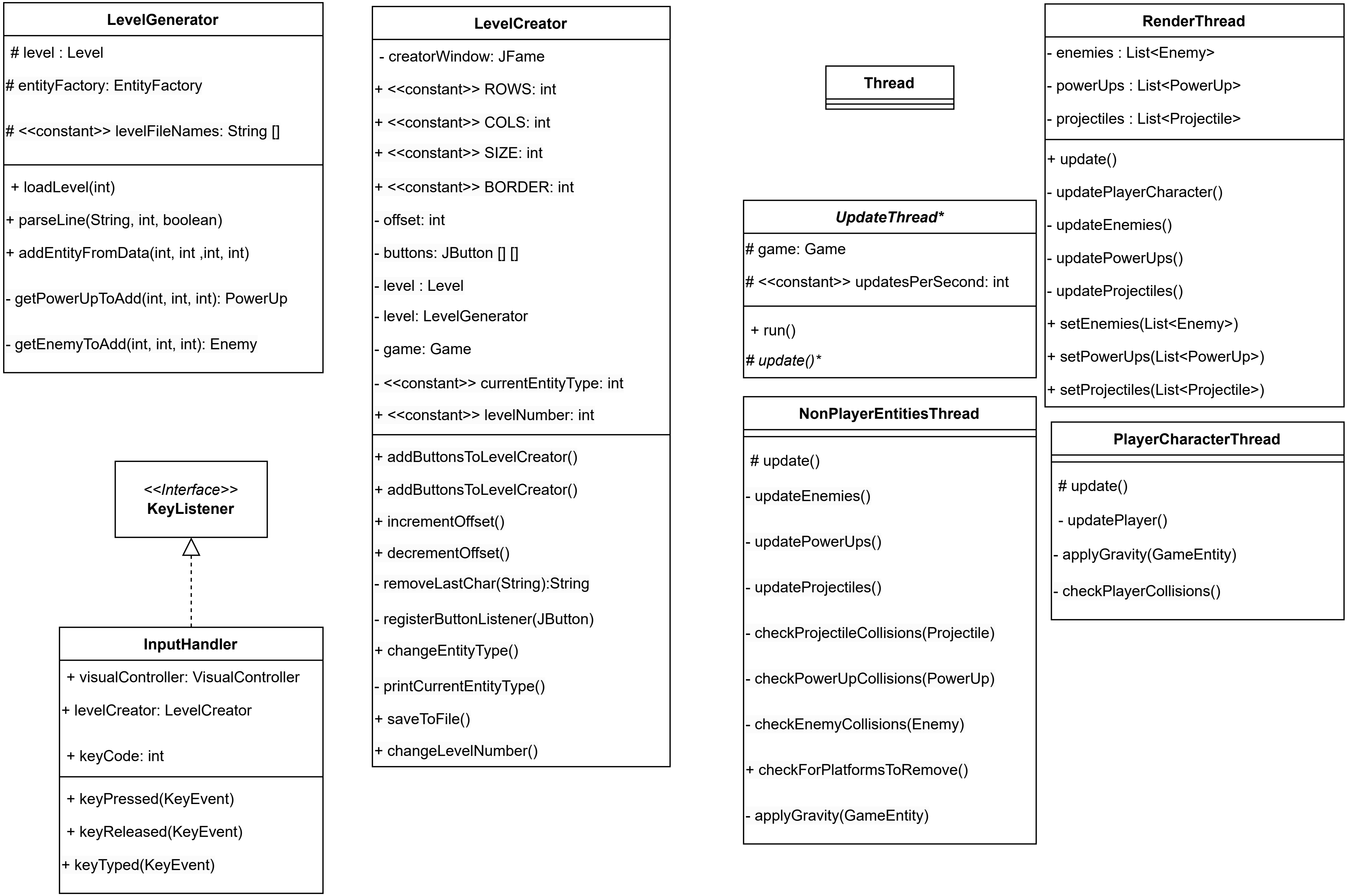
- timer: FireballCooldownTimer
- cooldownActive: boolean

+ hitPlatformFromBelow(Platform) +
+ initialize() +
+ spaceAction() +
- setFireballDirectionAndVelocity(FireballProjectile)
- createFireball():FireballProjectile
+ runIntoEnemy(Enemy) +
+ applySuperMushroom() +
+ applyStar() +
+ applyFireFlower() +

KoopaTroopaAliveShellState

+ reviveTimer : KoopaReviveTimer

+ hitFromAbove()+
+ initialize()+
- startReviveTimer()
- setShellSprite()
# setAnimatedSprite() +



LevelGenerator

# level : Level

# entityFactory: EntityFactory

# <<constant>> levelFileNames: String []

+ loadLevel(int)

+ parseLine(String, int, boolean)

+ addEntityFromData(int, int ,int, int)

- getPowerUpToAdd(int, int, int): PowerUp

- getEnemyToAdd(int, int, int): Enemy

<<Interface>>
KeyListener

InputHandler

+ visualController: VisualController

+ levelCreator: LevelCreator

+ keyCode: int

+ keyPressed(KeyEvent)

+ keyReleased(KeyEvent)

+ keyTyped(KeyEvent)

LevelCreator

- creatorWindow: JFame

+ <<constant>> ROWS: int

+ <<constant>> COLS: int

+ <<constant>> SIZE: int

+ <<constant>> BORDER: int

- offset: int

- buttons: JButton [] []

- level : Level

- level: LevelGenerator

- game: Game

- <<constant>> currentEntityType: int

+ <<constant>> levelNumber: int

+ addButtonsToLevelCreator()

+ addButtonsToLevelCreator()

+ incrementOffset()

+ decrementOffset()

- removeLastChar(String):String

- registerButtonListener(JButton)

+ changeEntityType()

- printCurrentEntityType()

+ saveToFile()

+ changeLevelNumber()

Thread

UpdateThread*

# game: Game

# <<constant>> updatesPerSecond: int

+ run()

*# update()**

NonPlayerEntitiesThread

# update()

- updateEnemies()

- updatePowerUps()

- updateProjectiles()

- checkProjectileCollisions(Projectile)

- checkPowerUpCollisions(PowerUp)

- checkEnemyCollisions(Enemy)

+ checkForPlatformsToRemove()

- applyGravity(GameEntity)

RenderThread

- enemies : List<Enemy>

- powerUps : List<PowerUp>

- projectiles : List<Projectile>

+ update()

- updatePlayerCharacter()

- updateEnemies()

- updatePowerUps()

- updateProjectiles()

+ setEnemies(List<Enemy>)

+ setPowerUps(List<PowerUp>)

+ setProjectiles(List<Projectile>)

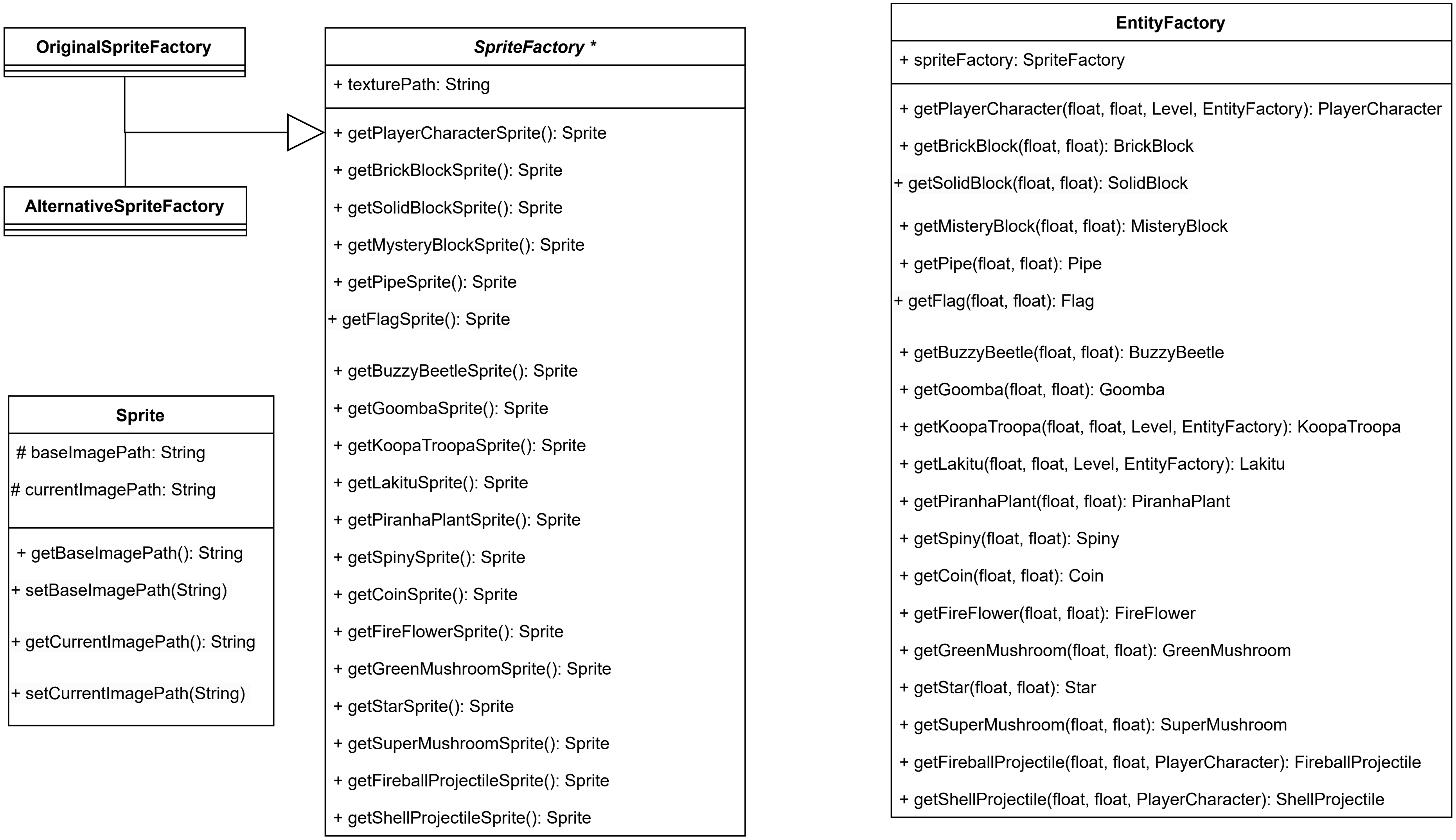
PlayerCharacterThread

# update()

- updatePlayer()

- applyGravity(GameEntity)

- checkPlayerCollisions()



OriginalSpriteFactory

AlternativeSpriteFactory

SpriteFactory *

+ texturePath: String

+ getPlayerCharacterSprite(): Sprite
+ getBrickBlockSprite(): Sprite
+ getSolidBlockSprite(): Sprite
+ getMysteryBlockSprite(): Sprite
+ getPipeSprite(): Sprite
+ getFlagSprite(): Sprite
+ getBuzzyBeetleSprite(): Sprite
+ getGoombaSprite(): Sprite
+ getKoopaTroopaSprite(): Sprite
+ getLakituSprite(): Sprite
+ getPiranhaPlantSprite(): Sprite
+ getSpinySprite(): Sprite
+ getCoinSprite(): Sprite
+ getFireFlowerSprite(): Sprite
+ getGreenMushroomSprite(): Sprite
+ getStarSprite(): Sprite
+ getSuperMushroomSprite(): Sprite
+ getFireballProjectileSprite(): Sprite
+ getShellProjectileSprite(): Sprite

EntityFactory

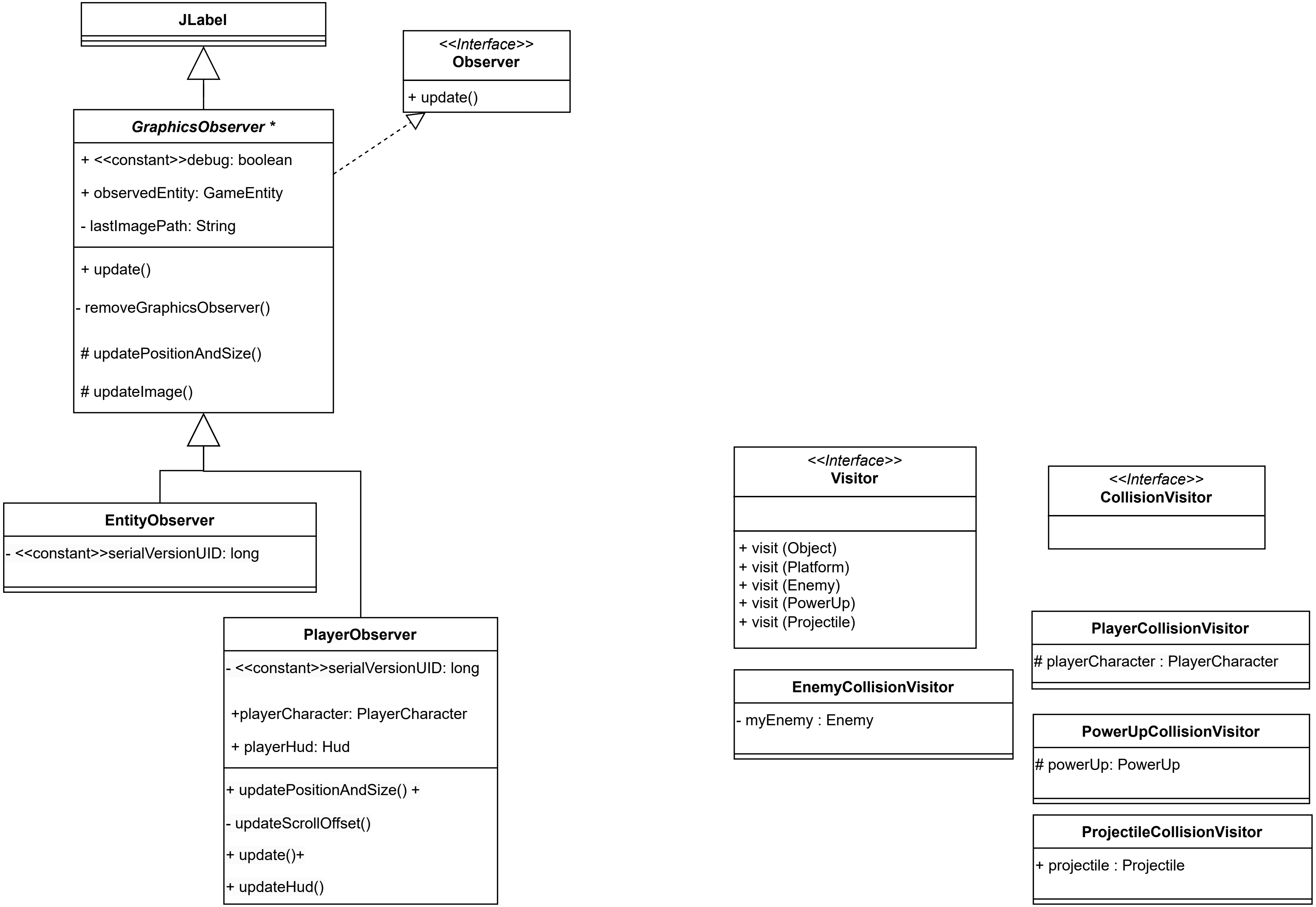
+ spriteFactory: SpriteFactory

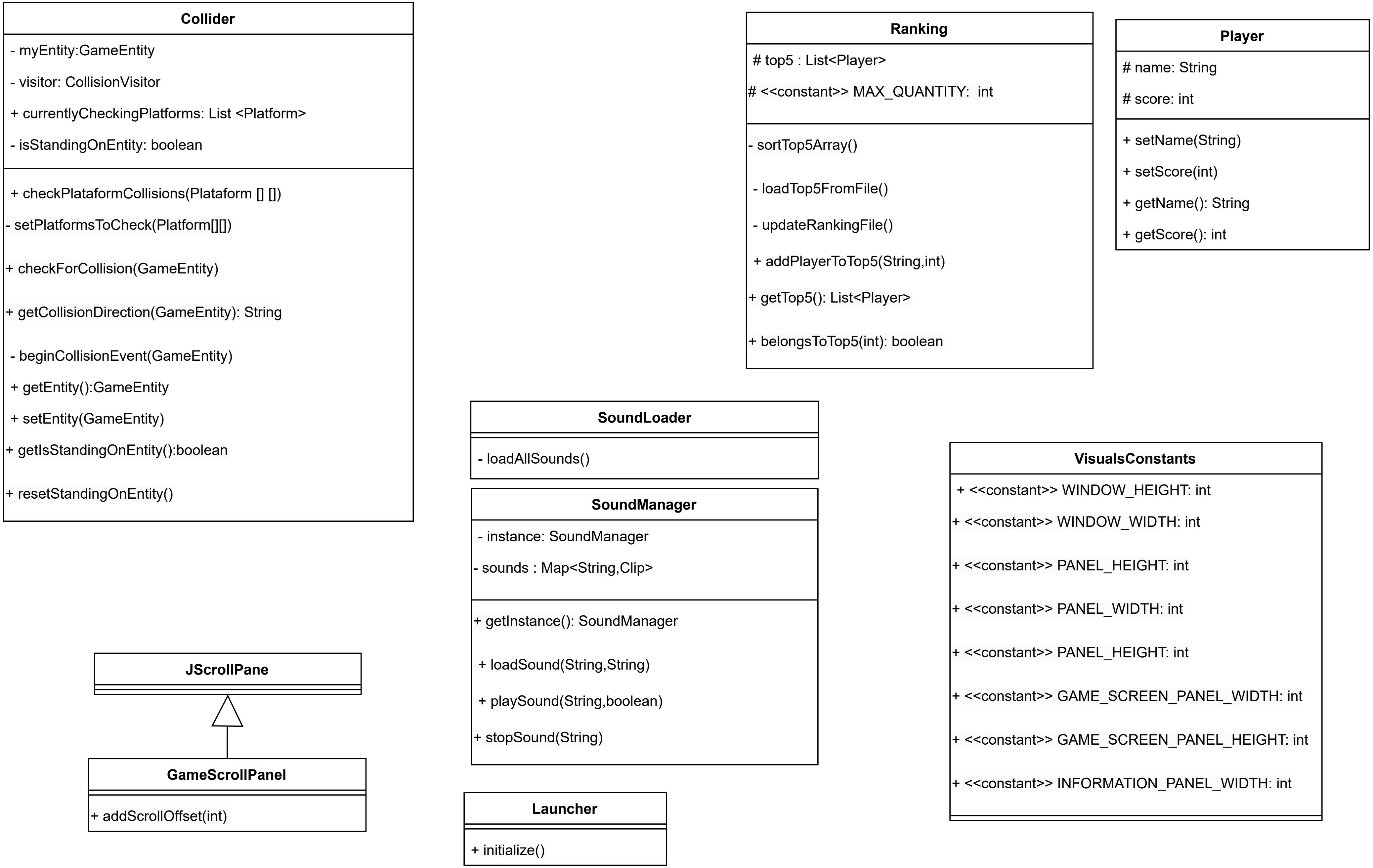
+ getPlayerCharacter(float, float, Level, EntityFactory): PlayerCharacter
+ getBrickBlock(float, float): BrickBlock
+ getSolidBlock(float, float): SolidBlock
+ getMisteryBlock(float, float): MisteryBlock
+ getPipe(float, float): Pipe
+ getFlag(float, float): Flag
+ getBuzzyBeetle(float, float): BuzzyBeetle
+ getGoomba(float, float): Goomba
+ getKoopaTroopa(float, float, Level, EntityFactory): KoopaTroopa
+ getLakitu(float, float, Level, EntityFactory): Lakitu
+ getPiranhaPlant(float, float): PiranhaPlant
+ getSpiny(float, float): Spiny
+ getCoin(float, float): Coin
+ getFireFlower(float, float): FireFlower
+ getGreenMushroom(float, float): GreenMushroom
+ getStar(float, float): Star
+ getSuperMushroom(float, float): SuperMushroom
+ getFireballProjectile(float, float, PlayerCharacter): FireballProjectile
+ getShellProjectile(float, float, PlayerCharacter): ShellProjectile

Sprite

# baseImagePath: String
# currentImagePath: String

+ getBaseImagePath(): String
+ setBaseImagePath(String)
+ getCurrentImagePath(): String
+ setCurrentImagePath(String)


JLabel
<<Interface>>
Observer
+ update()
GraphicsObserver *
+ <<constant>>debug: boolean
+ observedEntity: GameEntity
- lastImagePath: String
+ update()
- removeGraphicsObserver()
# updatePositionAndSize()
# updateImage()
EntityObserver
- <<constant>>serialVersionUID: long
PlayerObserver
- <<constant>>serialVersionUID: long
+playerCharacter: PlayerCharacter
+ playerHud: Hud
+ updatePositionAndSize() +
- updateScrollOffset()
+ update()+
+ updateHud()
<<Interface>>
Visitor
+ visit (Object)
+ visit (Platform)
+ visit (Enemy)
+ visit (PowerUp)
+ visit (Projectile)
<<Interface>>
CollisionVisitor
EnemyCollisionVisitor
- myEnemy : Enemy
PlayerCollisionVisitor
# playerCharacter : PlayerCharacter
PowerUpCollisionVisitor
# powerUp: PowerUp
ProjectileCollisionVisitor
+ projectile : Projectile



Collider

- myEntity:GameEntity
- visitor: CollisionVisitor
+ currentlyCheckingPlatforms: List <Platform>
- isStandingOnEntity: boolean

+ checkPlataformCollisions(Plataform [] [])
- setPlatformsToCheck(Platform[][])
+ checkForCollision(GameEntity)
+ getCollisionDirection(GameEntity): String
- beginCollisionEvent(GameEntity)
+ getEntity():GameEntity
+ setEntity(GameEntity)
+ getIsStandingOnEntity():boolean
+ resetStandingOnEntity()

Ranking

# top5 : List<Player>
# <<constant>> MAX_QUANTITY: int

- sortTop5Array()
- loadTop5FromFile()
- updateRankingFile()
+ addPlayerToTop5(String,int)
+ getTop5(): List<Player>
+ belongsToTop5(int): boolean

Player

# name: String
# score: int

+ setName(String)
+ setScore(int)
+ getName(): String
+ getScore(): int

SoundLoader

- loadAllSounds()

SoundManager

- instance: SoundManager
- sounds : Map<String,Clip>

+ getInstance(): SoundManager
+ loadSound(String,String)
+ playSound(String,boolean)
+ stopSound(String)

VisualsConstants

+ <<constant>> WINDOW_HEIGHT: int
+ <<constant>> WINDOW_WIDTH: int
+ <<constant>> PANEL_HEIGHT: int
+ <<constant>> PANEL_WIDTH: int
+ <<constant>> PANEL_HEIGHT: int
+ <<constant>> GAME_SCREEN_PANEL_WIDTH: int
+ <<constant>> GAME_SCREEN_PANEL_HEIGHT: int
+ <<constant>> INFORMATION_PANEL_WIDTH: int

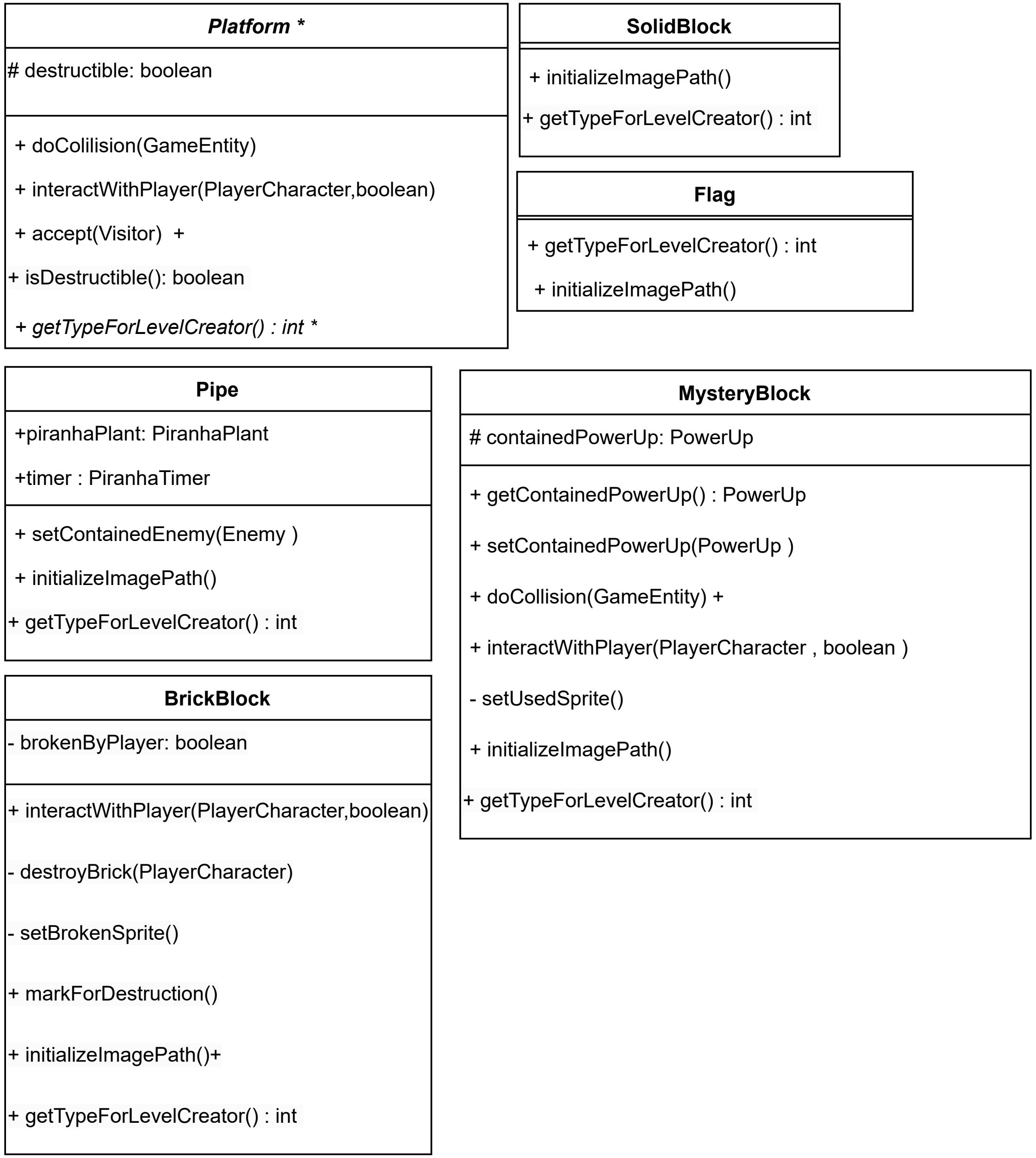
JScrollPane

GameScrollPanel

+ addScrollOffset(int)

Launcher

+ initialize()



Platform *

# destructible: boolean

+ doColilision(GameEntity)
+ interactWithPlayer(PlayerCharacter,boolean)
+ accept(Visitor) +
+ isDestructible(): boolean
+ _getTypeForLevelCreator() : int *_

SolidBlock

+ initializeImagePath()
+ getTypeForLevelCreator() : int

Flag

+ getTypeForLevelCreator() : int
+ initializeImagePath()

Pipe

+piranhaPlant: PiranhaPlant
+timer : PiranhaTimer

+ setContainedEnemy(Enemy)
+ initializeImagePath()
+ getTypeForLevelCreator() : int

BrickBlock

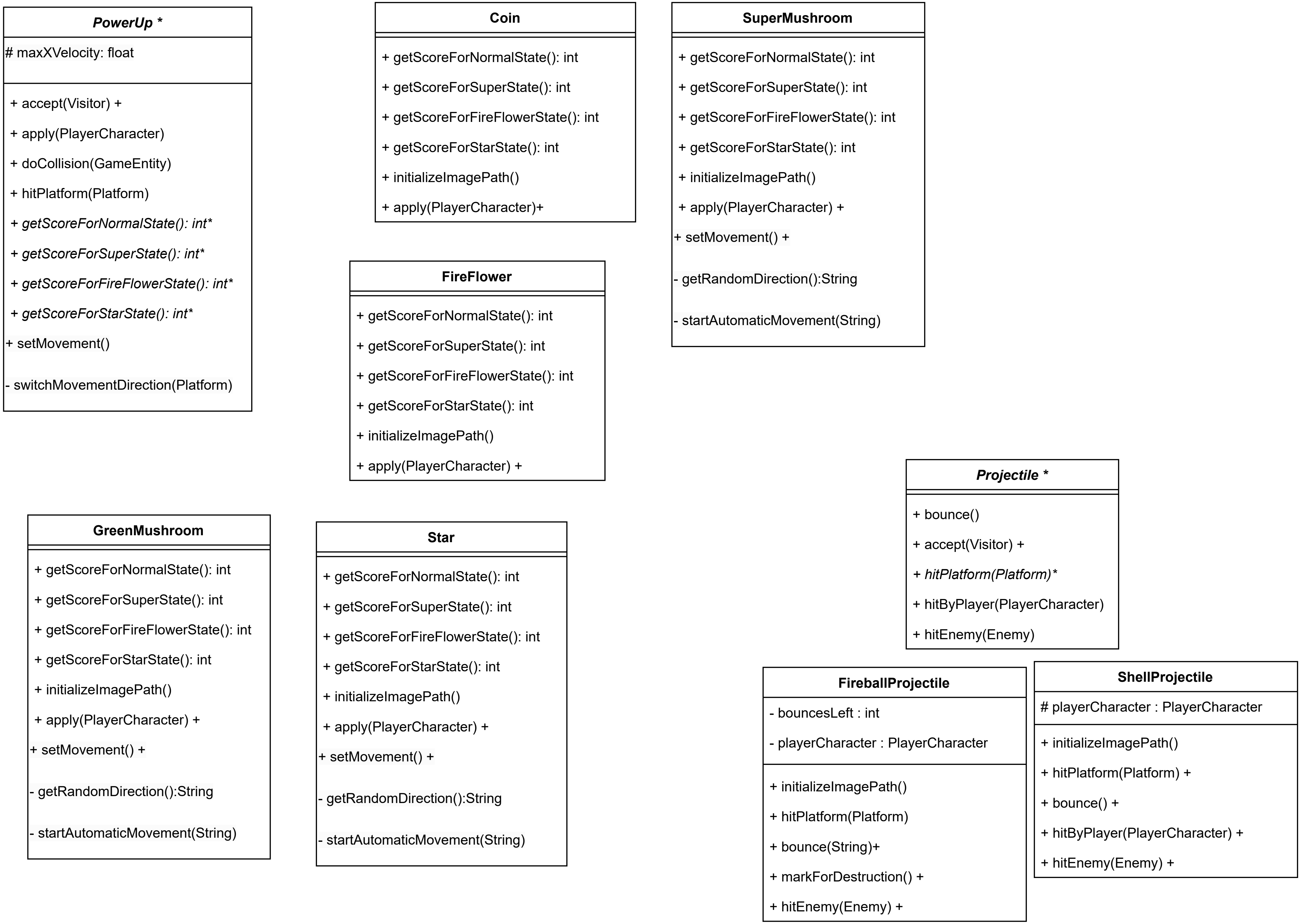
- brokenByPlayer: boolean

+ interactWithPlayer(PlayerCharacter,boolean)
- destroyBrick(PlayerCharacter)
- setBrokenSprite()
+ markForDestruction()
+ initializeImagePath()+
+ getTypeForLevelCreator() : int

MysteryBlock

# containedPowerUp: PowerUp

+ getContainedPowerUp() : PowerUp
+ setContainedPowerUp(PowerUp)
+ doCollision(GameEntity) +
+ interactWithPlayer(PlayerCharacter , boolean)
- setUsedSprite()
+ initializeImagePath()
+ getTypeForLevelCreator() : int



PowerUp *

# maxXVelocity: float

+ accept(Visitor) +

+ apply(PlayerCharacter)

+ doCollision(GameEntity)

+ hitPlatform(Platform)

+ getScoreForNormalState(): int*

+ getScoreForSuperState(): int*

+ getScoreForFireFlowerState(): int*

+ getScoreForStarState(): int*

+ setMovement()

- switchMovementDirection(Platform)

Coin

+ getScoreForNormalState(): int

+ getScoreForSuperState(): int

+ getScoreForFireFlowerState(): int

+ getScoreForStarState(): int

+ initializeImagePath()

+ apply(PlayerCharacter)+

SuperMushroom

+ getScoreForNormalState(): int

+ getScoreForSuperState(): int

+ getScoreForFireFlowerState(): int

+ getScoreForStarState(): int

+ initializeImagePath()

+ apply(PlayerCharacter) +

+ setMovement() +

- getRandomDirection():String

- startAutomaticMovement(String)

FireFlower

+ getScoreForNormalState(): int

+ getScoreForSuperState(): int

+ getScoreForFireFlowerState(): int

+ getScoreForStarState(): int

+ initializeImagePath()

+ apply(PlayerCharacter) +

Projectile *

+ bounce()

+ accept(Visitor) +

+ hitPlatform(Platform)*

+ hitByPlayer(PlayerCharacter)

+ hitEnemy(Enemy)

GreenMushroom

+ getScoreForNormalState(): int

+ getScoreForSuperState(): int

+ getScoreForFireFlowerState(): int

+ getScoreForStarState(): int

+ initializeImagePath()

+ apply(PlayerCharacter) +

+ setMovement() +

- getRandomDirection():String

- startAutomaticMovement(String)

Star

+ getScoreForNormalState(): int

+ getScoreForSuperState(): int

+ getScoreForFireFlowerState(): int

+ getScoreForStarState(): int

+ initializeImagePath()

+ apply(PlayerCharacter) +

+ setMovement() +

- getRandomDirection():String

- startAutomaticMovement(String)

FireballProjectile

- bouncesLeft : int

- playerCharacter : PlayerCharacter

+ initializeImagePath()

+ hitPlatform(Platform)

+ bounce(String)+

+ markForDestruction() +

+ hitEnemy(Enemy) +

ShellProjectile

# playerCharacter : PlayerCharacter

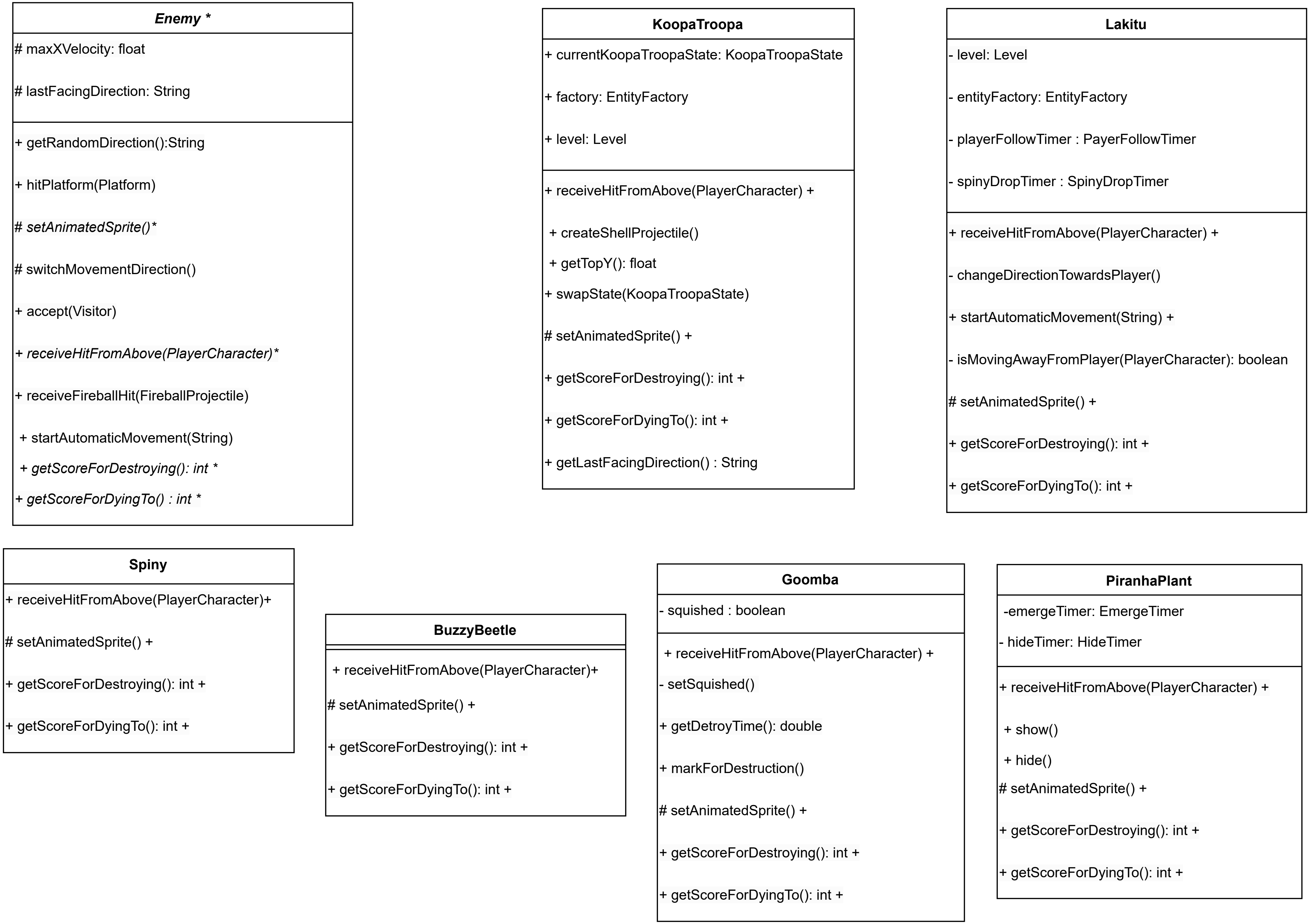
+ initializeImagePath()

+ hitPlatform(Platform) +

+ bounce() +

+ hitByPlayer(PlayerCharacter) +

+ hitEnemy(Enemy) +



Enemy *

# maxXVelocity: float

# lastFacingDirection: String

+ getRandomDirection():String

+ hitPlatform(Platform)

# setAnimatedSprite()*

# switchMovementDirection()

+ accept(Visitor)

+ receiveHitFromAbove(PlayerCharacter)*

+ receiveFireballHit(FireballProjectile)

+ startAutomaticMovement(String)

+ getScoreForDestroying(): int *

+ getScoreForDyingTo() : int *

KoopaTroopa

+ currentKoopaTroopaState: KoopaTroopaState

+ factory: EntityFactory

+ level: Level

+ receiveHitFromAbove(PlayerCharacter) +

+ createShellProjectile()

+ getTopY(): float

+ swapState(KoopaTroopaState)

# setAnimatedSprite() +

+ getScoreForDestroying(): int +

+ getScoreForDyingTo(): int +

+ getLastFacingDirection() : String

Lakitu

- level: Level

- entityFactory: EntityFactory

- playerFollowTimer : PayerFollowTimer

- spinyDropTimer : SpinyDropTimer

+ receiveHitFromAbove(PlayerCharacter) +

- changeDirectionTowardsPlayer()

+ startAutomaticMovement(String) +

- isMovingAwayFromPlayer(PlayerCharacter): boolean

# setAnimatedSprite() +

+ getScoreForDestroying(): int +

+ getScoreForDyingTo(): int +

Spiny

+ receiveHitFromAbove(PlayerCharacter)+

# setAnimatedSprite() +

+ getScoreForDestroying(): int +

+ getScoreForDyingTo(): int +

BuzzyBeetle

+ receiveHitFromAbove(PlayerCharacter)+

# setAnimatedSprite() +

+ getScoreForDestroying(): int +

+ getScoreForDyingTo(): int +

Goomba

- squished : boolean

+ receiveHitFromAbove(PlayerCharacter) +

- setSquished()

+ getDetroyTime(): double

+ markForDestruction()

# setAnimatedSprite() +

+ getScoreForDestroying(): int +

+ getScoreForDyingTo(): int +

PiranhaPlant

-emergeTimer: EmergeTimer

- hideTimer: HideTimer

+ receiveHitFromAbove(PlayerCharacter) +

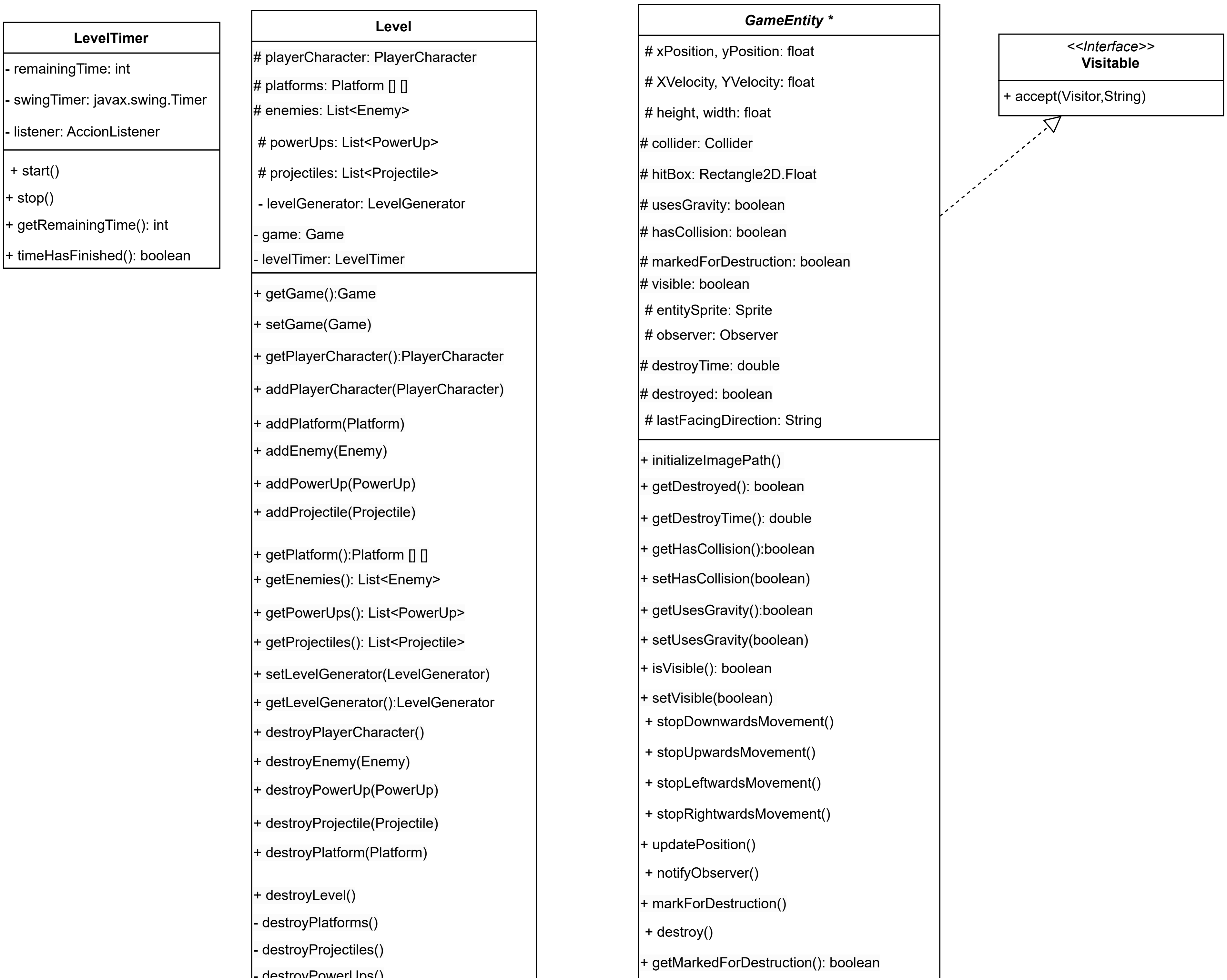
+ show()

+ hide()

# setAnimatedSprite() +

+ getScoreForDestroying(): int +

+ getScoreForDyingTo(): int +



LevelTimer

- remainingTime: int
- swingTimer: javax.swing.Timer
- listener: AccionListener

+ start()
+ stop()
+ getRemainingTime(): int
+ timeHasFinished(): boolean

Level

# playerCharacter: PlayerCharacter
# platforms: Platform [] []
# enemies: List<Enemy>
# powerUps: List<PowerUp>
# projectiles: List<Projectile>
- levelGenerator: LevelGenerator
- game: Game
- levelTimer: LevelTimer

+ getGame():Game
+ setGame(Game)
+ getPlayerCharacter():PlayerCharacter
+ addPlayerCharacter(PlayerCharacter)
+ addPlatform(Platform)
+ addEnemy(Enemy)
+ addPowerUp(PowerUp)
+ addProjectile(Projectile)
+ getPlatform():Platform [] []
+ getEnemies(): List<Enemy>
+ getPowerUps(): List<PowerUp>
+ getProjectiles(): List<Projectile>
+ setLevelGenerator(LevelGenerator)
+ getLevelGenerator():LevelGenerator
+ destroyPlayerCharacter()
+ destroyEnemy(Enemy)
+ destroyPowerUp(PowerUp)
+ destroyProjectile(Projectile)
+ destroyPlatform(Platform)
+ destroyLevel()
- destroyPlatforms()
- destroyProjectiles()
- destroyPowerUps()

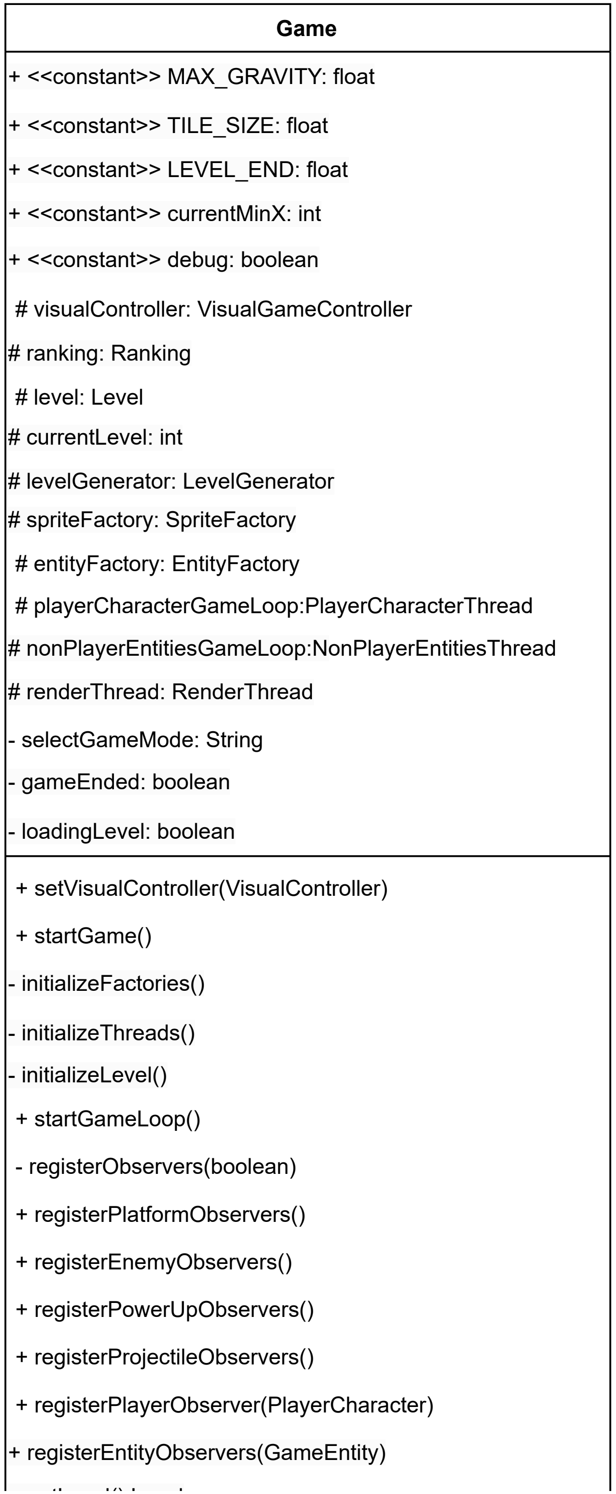
GameEntity *

# xPosition, yPosition: float
# XVelocity, YVelocity: float
# height, width: float
# collider: Collider
# hitBox: Rectangle2D.Float
# usesGravity: boolean
# hasCollision: boolean
# markedForDestruction: boolean
# visible: boolean
# entitySprite: Sprite
# observer: Observer
# destroyTime: double
# destroyed: boolean
# lastFacingDirection: String

+ initializeImagePath()
+ getDestroyed(): boolean
+ getDestroyTime(): double
+ getHasCollision():boolean
+ setHasCollision(boolean)
+ getUsesGravity():boolean
+ setUsesGravity(boolean)
+ isVisible(): boolean
+ setVisible(boolean)
+ stopDownwardsMovement()
+ stopUpwardsMovement()
+ stopLeftwardsMovement()
+ stopRightwardsMovement()
+ updatePosition()
+ notifyObserver()
+ markForDestruction()
+ destroy()
+ getMarkedForDestruction(): boolean

<<Interface>>
Visitable

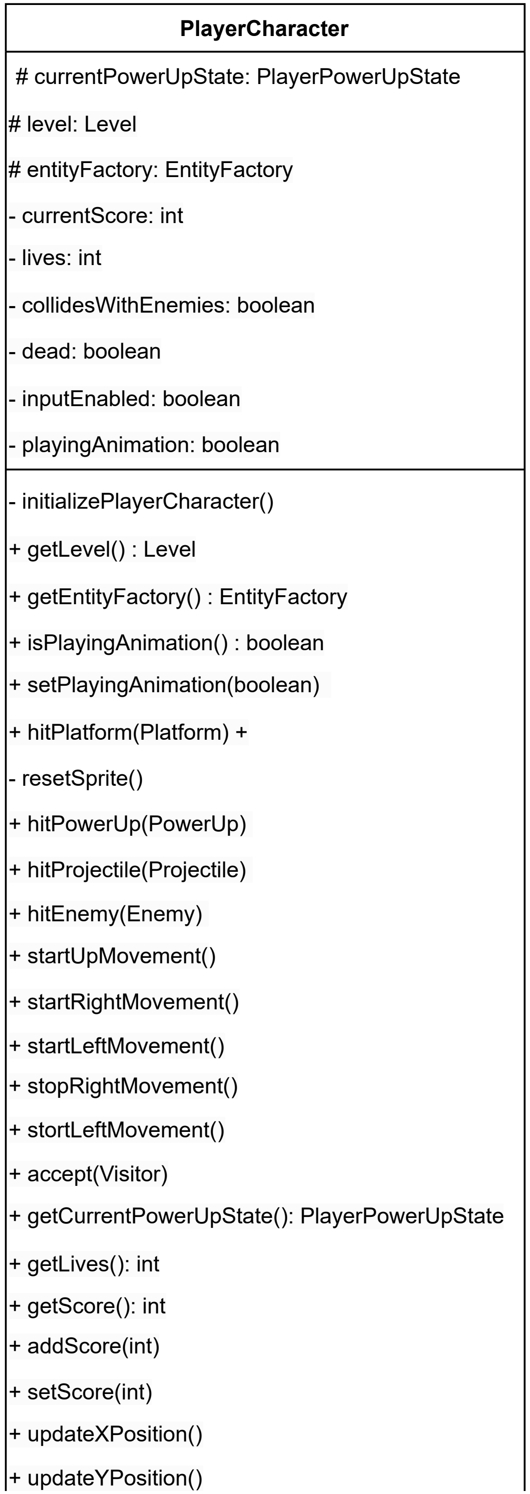
+ accept(Visitor,String)



Game

+ <<constant>> MAX_GRAVITY: float
+ <<constant>> TILE_SIZE: float
+ <<constant>> LEVEL_END: float
+ <<constant>> currentMinX: int
+ <<constant>> debug: boolean
# visualController: VisualGameController
# ranking: Ranking
# level: Level
# currentLevel: int
# levelGenerator: LevelGenerator
# spriteFactory: SpriteFactory
# entityFactory: EntityFactory
# playerCharacterGameLoop:PlayerCharacterThread
# nonPlayerEntitiesGameLoop:NonPlayerEntitiesThread
# renderThread: RenderThread
- selectGameMode: String
- gameEnded: boolean
- loadingLevel: boolean

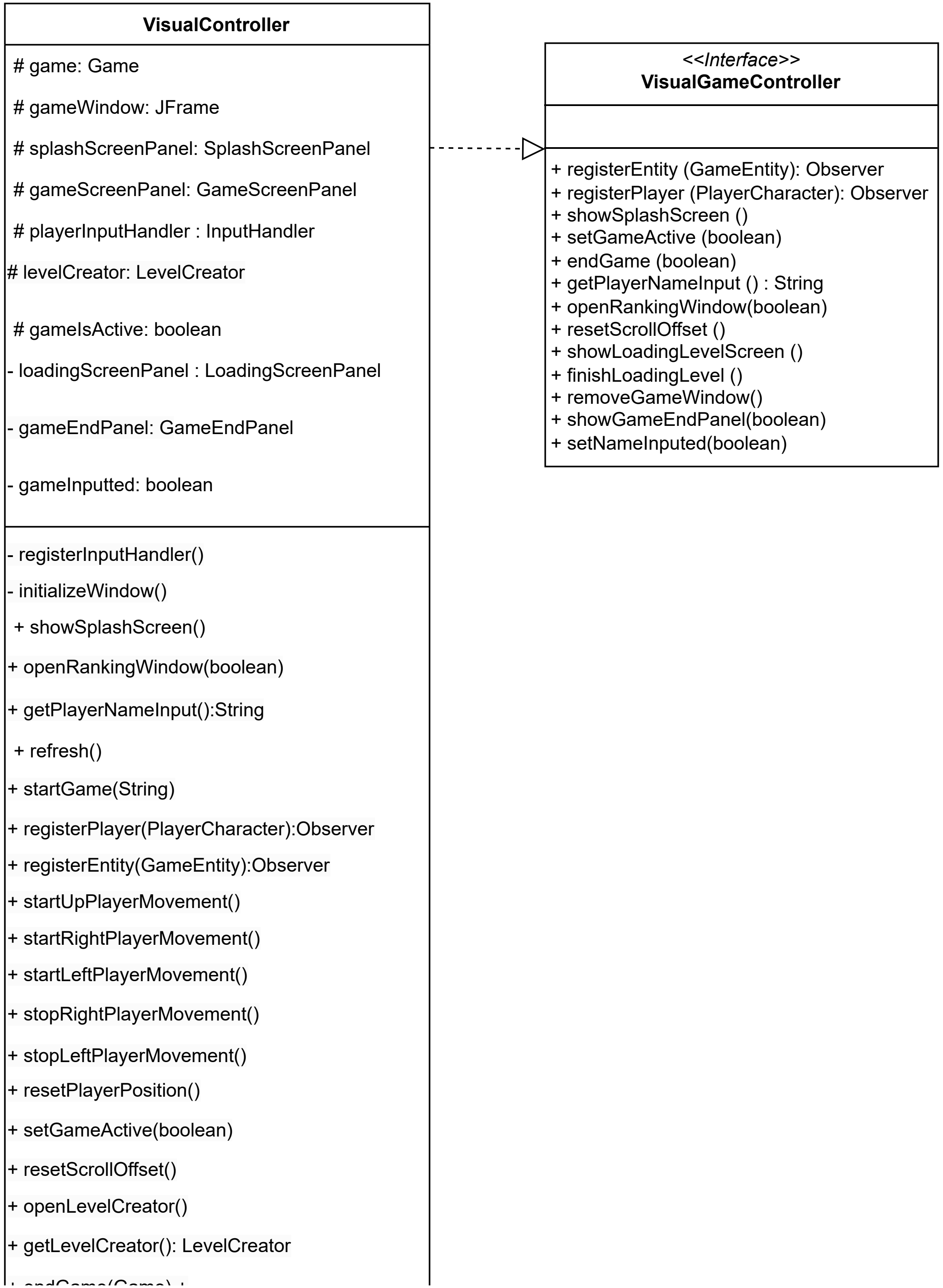
+ setVisualController(VisualController)
+ startGame()
- initializeFactories()
- initializeThreads()
- initializeLevel()
+ startGameLoop()
- registerObservers(boolean)
+ registerPlatformObservers()
+ registerEnemyObservers()
+ registerPowerUpObservers()
+ registerProjectileObservers()
+ registerPlayerObserver(PlayerCharacter)
+ registerEntityObservers(GameEntity)



PlayerCharacter

# currentPowerUpState: PlayerPowerUpState
# level: Level
# entityFactory: EntityFactory
- currentScore: int
- lives: int
- collidesWithEnemies: boolean
- dead: boolean
- inputEnabled: boolean
- playingAnimation: boolean

- initializePlayerCharacter()
+ getLevel() : Level
+ getEntityFactory() : EntityFactory
+ isPlayingAnimation() : boolean
+ setPlayingAnimation(boolean)
+ hitPlatform(Platform) +
- resetSprite()
+ hitPowerUp(PowerUp)
+ hitProjectile(Projectile)
+ hitEnemy(Enemy)
+ startUpMovement()
+ startRightMovement()
+ startLeftMovement()
+ stopRightMovement()
+ stortLeftMovement()
+ accept(Visitor)
+ getCurrentPowerUpState(): PlayerPowerUpState
+ getLives(): int
+ getScore(): int
+ addScore(int)
+ setScore(int)
+ updateXPosition()
+ updateYPosition()



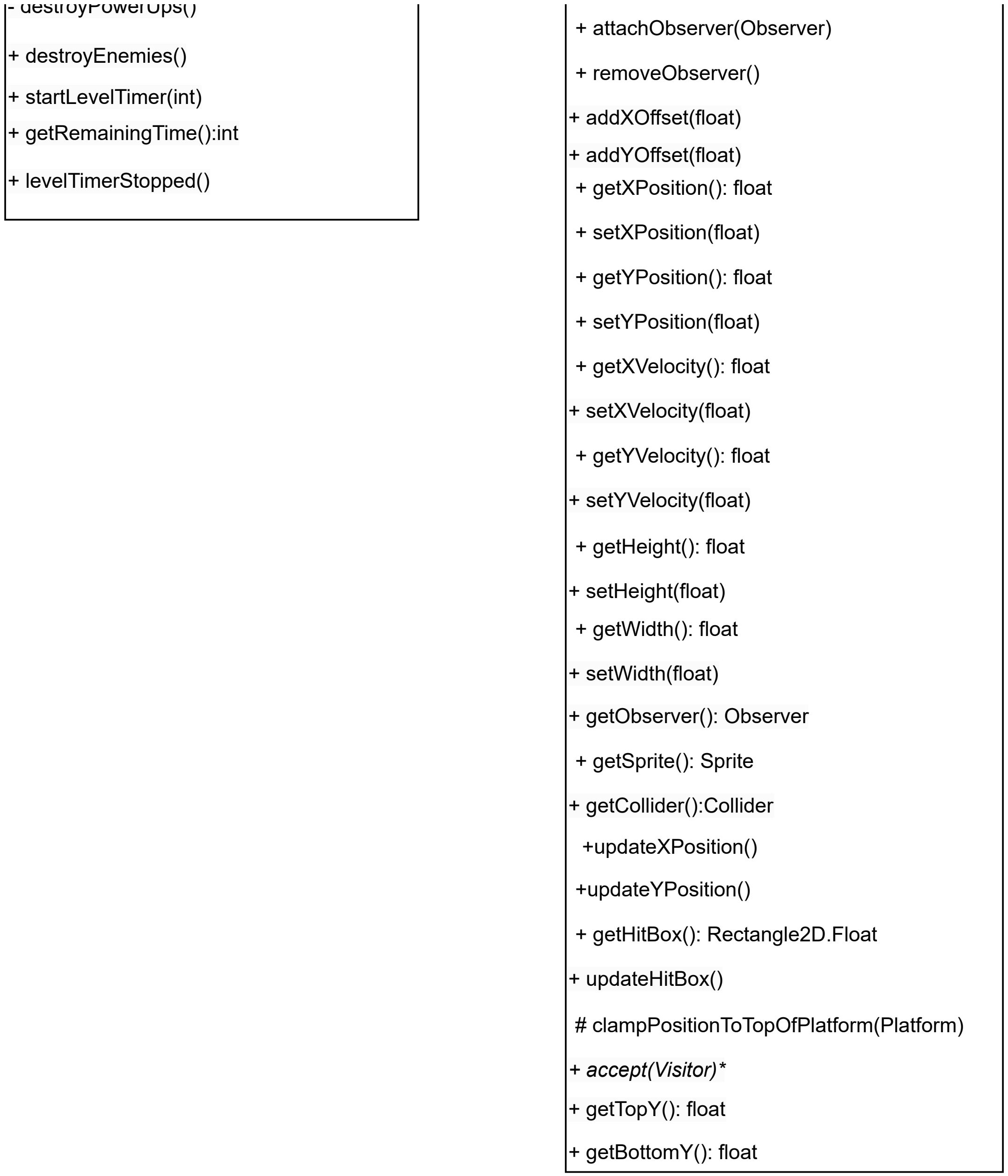
VisualController

# game: Game
# gameWindow: JFrame
# splashScreenPanel: SplashScreenPanel
# gameScreenPanel: GameScreenPanel
# playerInputHandler : InputHandler
# levelCreator: LevelCreator
# gameIsActive: boolean
- loadingScreenPanel : LoadingScreenPanel
- gameEndPanel: GameEndPanel
- gameInputted: boolean

- registerInputHandler()
- initializeWindow()
+ showSplashScreen()
+ openRankingWindow(boolean)
+ getPlayerNameInput():String
+ refresh()
+ startGame(String)
+ registerPlayer(PlayerCharacter):Observer
+ registerEntity(GameEntity):Observer
+ startUpPlayerMovement()
+ startRightPlayerMovement()
+ startLeftPlayerMovement()
+ stopRightPlayerMovement()
+ stopLeftPlayerMovement()
+ resetPlayerPosition()
+ setGameActive(boolean)
+ resetScrollOffset()
+ openLevelCreator()
+ getLevelCreator(): LevelCreator
+ endGame(Game)

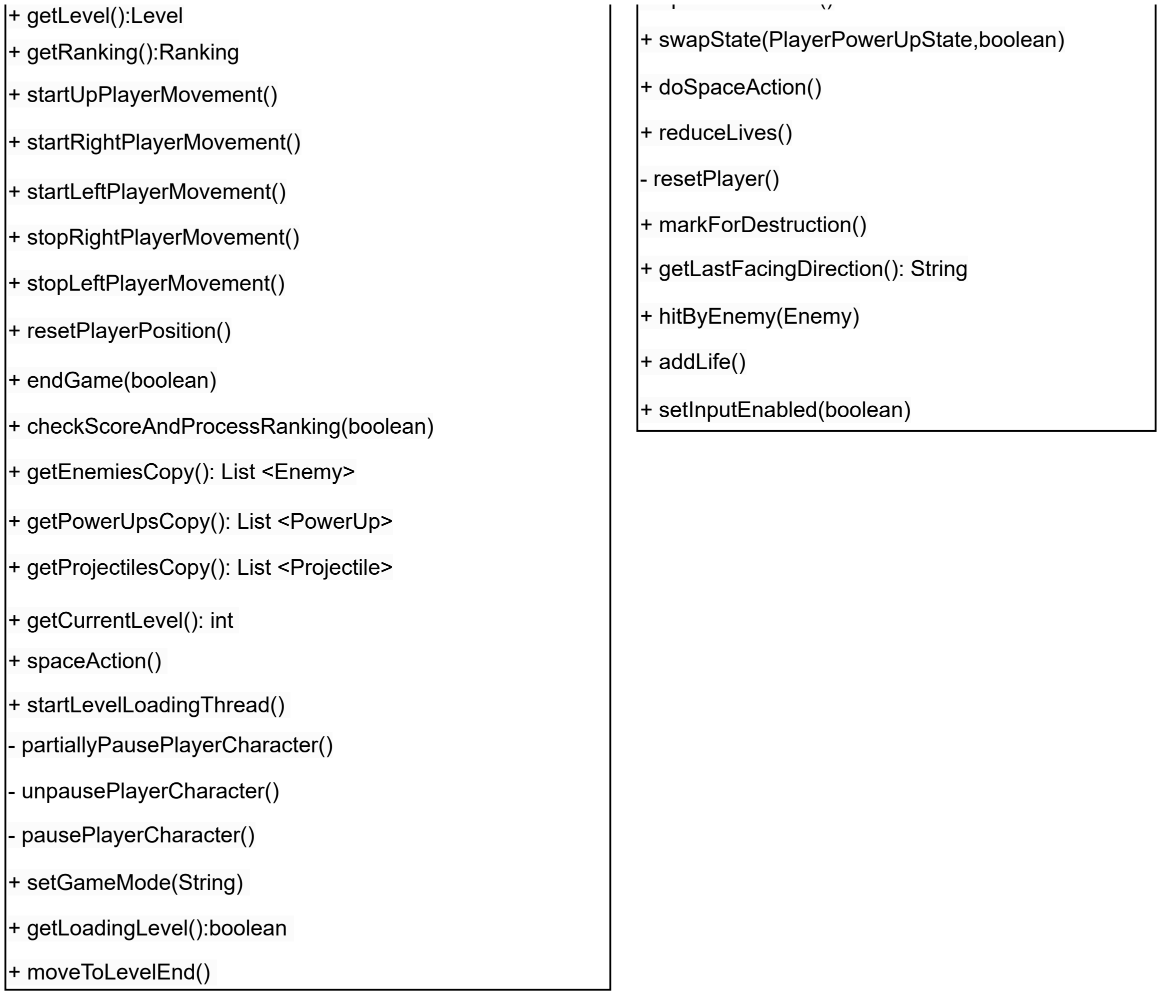
<<Interface>>
VisualGameController

+ registerEntity (GameEntity): Observer
+ registerPlayer (PlayerCharacter): Observer
+ showSplashScreen ()
+ setGameActive (boolean)
+ endGame (boolean)
+ getPlayerNameInput () : String
+ openRankingWindow(boolean)
+ resetScrollOffset ()
+ showLoadingLevelScreen ()
+ finishLoadingLevel ()
+ removeGameWindow()
+ showGameEndPanel(boolean)
+ setNameInputed(boolean)



- destroyPowerUps()
+ destroyEnemies()
+ startLevelTimer(int)
+ getRemainingTime():int
+ levelTimerStopped()

+ attachObserver(Observer)
+ removeObserver()
+ addXOffset(float)
+ addYOffset(float)
+ getXPosition(): float
+ setXPosition(float)
+ getYPosition(): float
+ setYPosition(float)
+ getXVelocity(): float
+ setXVelocity(float)
+ getYVelocity(): float
+ setYVelocity(float)
+ getHeight(): float
+ setHeight(float)
+ getWidth(): float
+ setWidth(float)
+ getObserver(): Observer
+ getSprite(): Sprite
+ getCollider():Collider
+updateXPosition()
+updateYPosition()
+ getHitBox(): Rectangle2D.Float
+ updateHitBox()
# clampPositionToTopOfPlatform(Platform)
+ accept(Visitor)*
+ getTopY(): float
+ getBottomY(): float



+ getLevel():Level
+ getRanking():Ranking
+ startUpPlayerMovement()
+ startRightPlayerMovement()
+ startLeftPlayerMovement()
+ stopRightPlayerMovement()
+ stopLeftPlayerMovement()
+ resetPlayerPosition()
+ endGame(boolean)
+ checkScoreAndProcessRanking(boolean)
+ getEnemiesCopy(): List <Enemy>
+ getPowerUpsCopy(): List <PowerUp>
+ getProjectilesCopy(): List <Projectile>
+ getCurrentLevel(): int
+ spaceAction()
+ startLevelLoadingThread()
- partiallyPausePlayerCharacter()
- unpausePlayerCharacter()
- pausePlayerCharacter()
+ setGameMode(String)
+ getLoadingLevel():boolean
+ moveToLevelEnd()

+ swapState(PlayerPowerUpState,boolean)
+ doSpaceAction()
+ reduceLives()
- resetPlayer()
+ markForDestruction()
+ getLastFacingDirection(): String
+ hitByEnemy(Enemy)
+ addLife()
+ setInputEnabled(boolean)

+ endGame(Game) +

+ setScroollOffset(int)

+ spaceAction()

+ showLoadingLevelScreen() +

+ finishLoadingLevel() +

+ showGameEndPanel(boolean)

+ moveToLevelEnd()

+ removeGameWindow()

+ setNameInputed(boolean)