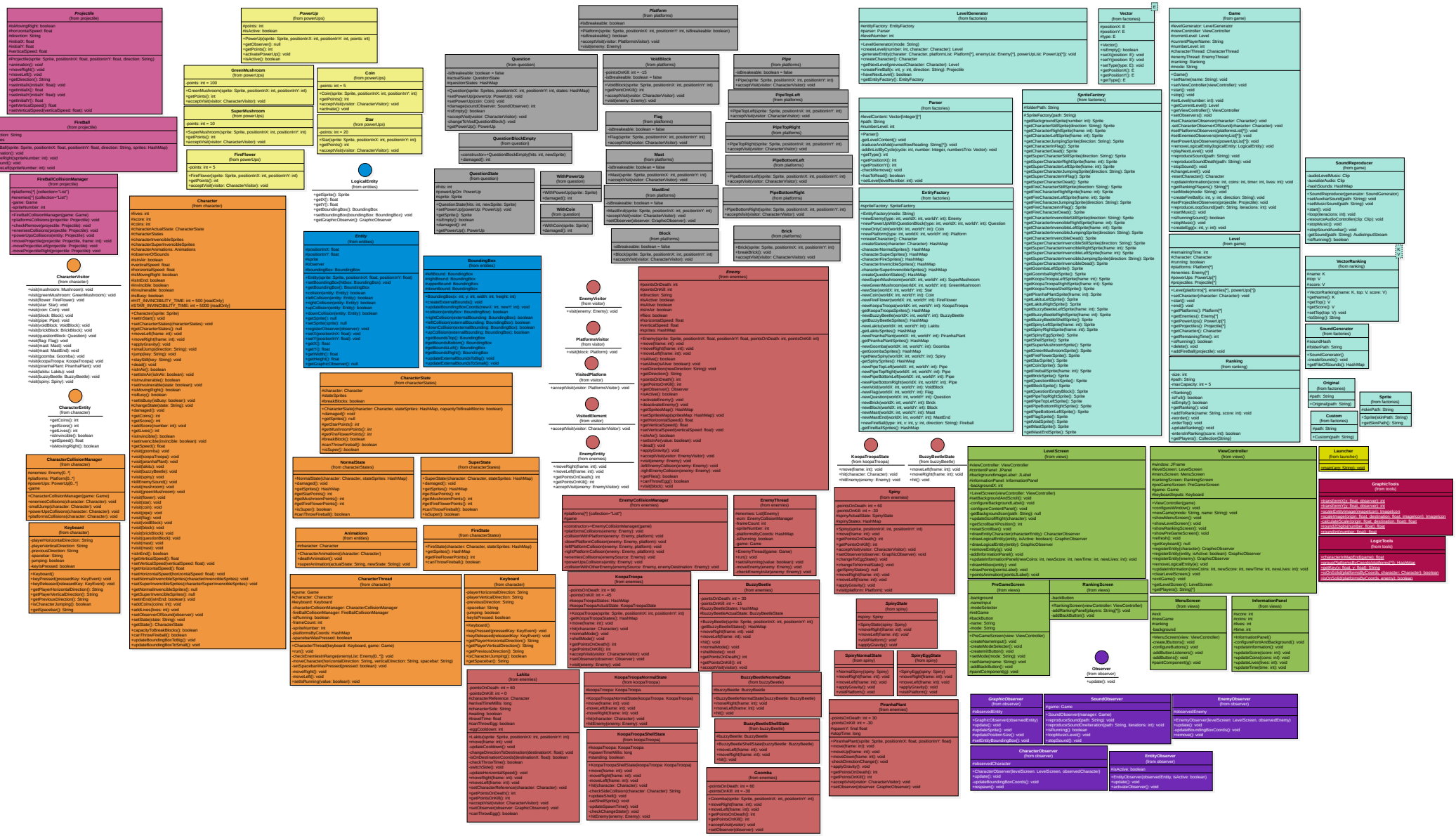
Projectile
(from projectile)
#isMovingRight: boolean
#horizontalSpeed: float
#direction: String
#initialX: float
#initialY: float
#verticalSpeed: float
+Projectile(sprite: Sprite, positionX: float, positionY: float, direction: String)
+animation(): void
+moveRight(): void
+moveLeft(): void
+getDirection(): String
+setInitialX(initialX: float): void
+getInitialX(): float
+setInitialY(initialY: float): void
+getInitialY(): float
+getVerticalSpeed(): float
+setVerticalSpeed(verticalSpeed: float): void
Fireball
(from projectile)
#direction: String
#sprites
+FireBall(sprite: Sprite, positionX: float, positionY: float, direction: String, sprites: HashMap)
+animation(): void
+moveRight(spriteNumber: int): void
+rebound(): void
+moveLeft(spriteNumber: int): void
FireBallCollisionManager
(from projectile)
#platforms[*] {collection="List"}
#enemies[*] {collection="List"}
-game: Game
-spriteNumber: int
+FireBallCollisionManager(game: Game)
+platformsCollisions(projectile: Projectile): void
+checkMotionless(projectile: Projectile): void
+enemiesCollisions(projectile: Projectile): void
+powerUpsCollisions(entity: Projectile): void
+moveProjectile(projectile: Projectile, frame: int): void
+moveProjectileLeft(projectile: Projectile): void
+moveProjectileRight(projectile: Projectile): void
CharacterVisitor
(from character)
+visit(mushroom: Mushroom): void
+visit(greenMushroom: GreenMushroom): void
+visit(flower: FireFlower): void
+visit(star: Star): void
+visit(coin: Coin): void
+visit(block: Block): void
+visit(pipe: Pipe): void
+visit(voidBlock: VoidBlock): void
+visit(brickBlock: BrickBlock): void
+visit(questionBlock: QuestionBlock): void
+visit(flag: Flag): void
+visit(mast: Mast): void
+visit(mast: MastEnd): void
+visit(goomba: Goomba): void
+visit(koopaTroopa: KoopaTroopa): void
+visit(piranhaPlant: PiranhaPlant): void
+visit(lakitu: Lakitu): void
+visit(buzzyBeetle: BuzzyBeetle): void
+visit(spiny: Spiny): void
CharacterEntity
(from character)
+getCoins(): int
+getScore(): int
+getLives(): int
+isInvincible(): boolean
+getSpeed(): float
+isMovingRight(): boolean
CharacterCollisionManager
(from character)
#enemies: Enemy[0..*]
#platforms: Platform[0..*]
#powerUps: PowerUp[0..*]
-game
+characterCollisionManager(game: Game)
+enemiesCollisions(character: Character): void
+smallJump(character: Character): void
+powerUpsCollisions(character: Character): void
+platformsCollisions(character: Character): void
Keyboard
(from character)
-playerHorizontalDirection: String
-playerVerticalDirection: String
-previousDirection: String
-spacebar: String
-jumping: boolean
-keysPressed: boolean
+Keyboard()
+keyPressed(pressedKey: KeyEvent): void
+keyReleased(releasedKey: KeyEvent): void
+getPlayerHorizontalDirection(): String
+getPlayerVerticalDirection(): String
+getPreviousDirection(): String
+isCharacterJumping(): boolean
+getSpacebar(): String
PowerUp
(from powerUps)
#points: int
#isActive: boolean
+PowerUp(sprite: Sprite, positionX: int, positionY: int, points: int)
+getObserver(): null
+getPoints(): int
+activatePowerUp(): void
+isActive(): boolean
GreenMushroom
(from powerUps)
-points: int = 100
+GreenMushroom(sprite: Sprite, positionX: int, positionY: int)
+getPoints(): int
+acceptVisit(visitor: CharacterVisitor): void
SuperMushroom
(from powerUps)
-points: int = 10
+SuperMushroom(sprite: Sprite, positionX: int, positionY: int)
+getPoints(): int
+acceptVisit(visitor: CharacterVisitor): void
FireFlower
(from powerUps)
-points: int = 5
+FireFlower(sprite: Sprite, positionX: int, positionY: int)
+getPoints(): int
+acceptVisit(visitor: CharacterVisitor): void
Coin
(from powerUps)
-points: int = 5
+Coin(sprite: Sprite, positionX: int, positionY: int)
+getPoints(): int
+acceptVisit(visitor: CharacterVisitor): void
+activate(): void
Star
(from powerUps)
-points: int = 20
+Star(sprite: Sprite, positionX: int, positionY: int)
+getPoints(): int
+acceptVisit(visitor: CharacterVisitor): void
LogicalEntity
(from entities)
+getSprite(): Sprite
+getX(): float
+getY(): float
+getBoundingBox(): BoundingBox
+setBoundingBox(boundingBox: BoundingBox): void
+getGraphicObserver(): GraphicObserver
Character
(from character)
#lives: int
#score: int
#coins: int
#characterActualState: CharacterState
#characterStates
#characterInvincibleSprites
#characterSuperInvincibleSprites
#characterAnimations: Animations
#observerOfSounds
#isInAir: boolean
#verticalSpeed: float
#horizontalSpeed: float
#isMovingRight: boolean
#isInEnd: boolean
#invincible: boolean
#invulnerable: boolean
#isBusy: boolean
#HIT_INVINCIBILITY_TIME: int = 500 {readOnly}
#STAR_INVINCIBILITY_TIME: int = 5000 {readOnly}
+Character(sprite: Sprite)
+setInStart(): void
+setCharacterStates(characterStates): void
#getCharacterStates(): null
+moveLeft(frame: int): void
+moveRight(frame: int): void
+applyGravity(): void
+smallJump(direction: String): void
+jumpNow(): void
+staySlidEkey: String): void
+dead(): void
+isInAir(): boolean
+setIsInAir(isInAir: boolean): void
+isInvulnerable(): boolean
+setInvulnerable(state: boolean): void
+isMovingRight(): boolean
+isBusy(): boolean
+setIsBusy(isBusy: boolean): void
+changeState(state: String): void
+damaged(): void
+getCoins(): int
+getScore(): int
+addScore(number: int): void
+getLives(): int
+isInvincible(): boolean
+setInvincible(invincible: boolean): void
+getSpeed(): float
+isMovingRight(): boolean
+visit(goomba): void
+visit(koopaTroopa): void
+visit(piranhaPlant): void
+visit(lakitu): void
+visit(buzzyBeetle): void
+visit(spiny): void
+visit(enemySound): void
+visit(mushroom): void
+visit(greenMushroom): void
+visit(flower): void
+visit(star): void
+visit(coin): void
+visit(pipe): void
+visit(flag): void
+visit(voidBlock): void
+visit(block): void
+visit(questionBlock): void
+visit(mast): void
+visit(mast): void
+isInEnd(): boolean
+getVerticalSpeed(): float
+setVerticalSpeed(verticalSpeed: float): void
+getHorizontalSpeed(): float
+setHorizontalSpeed(horizontalSpeed: float): void
+setNormalInvincibleSprites(characterInvincibleSprites): void
+setSuperInvincibleSprites(characterSuperInvincibleSprites): void
+getNormalInvincibleSprites(): null
+getSuperInvincibleSprites(): null
+setIsInEnd(isInEnd: boolean): void
+addCoins(coins: int): void
+addLives(lives: int): void
+setObserverOf(sound observer): void
+setState(state: String): void
+getState(): CharacterState
+capacityToBreakBlocks(): boolean
+canThrowFireball(): boolean
+updateBoundingBoxToBig(): void
+updateBoundingBoxToSmall(): void
Entity
(from entities)
#positionX: float
#positionY: float
#sprite
#observer
#boundingBox: BoundingBox
+Entity(sprite: Sprite, positionX: float, positionY: float)
+setBoundingBox(bbox: BoundingBox): void
+getBoundingBox(): BoundingBox
+collision(entity: Entity): boolean
+isRCollision(entity: Entity): boolean
+isLCollision(entity: Entity): boolean
+isUCollision(entity: Entity): boolean
+isDownCollision(entity: Entity): boolean
+setSprite(): null
+setSprite(sprite): null
+registerObserver(observer): void
+setPositionX(positionX: float): void
+setY(positionY: float): void
+getX(): float
+getY(): float
+getWidth(): float
+getHeight(): float
+getGraphicObserver(): null
BoundingBox
(from entities)
#leftBound: BoundingBox
#rightBound: BoundingBox
#upperBound: BoundingBox
#downBound: BoundingBox
+BoundingBox(x: int, y: int, width: int, height: int)
+createExternalBounds(): void
+updateBoundingBoxCoords(newX: int, newY: int): void
+collision(entityBox: BoundingBox): boolean
+rightCollision(externalBounding: BoundingBox): boolean
+leftCollision(externalBounding: BoundingBox): boolean
+downCollision(externalBounding: BoundingBox): boolean
+upCollision(externalBounding: BoundingBox): boolean
+getBoundsTop(): BoundingBox
+getBoundsBottom(): BoundingBox
+getBoundsLeft(): BoundingBox
+getBoundsRight(): BoundingBox
+updateExternalBoundsToBig(): void
+updateExternalBoundsToSmall(): void
CharacterState
(from characterStates)
#character: Character
#stateSprites
#breakBlocks: boolean
+CharacterState(character: Character, stateSprites: HashMap, capacityToBreakBlocks: boolean)
+damaged(): void
+getSprites(): null
+getStarPoints(): int
#getMushroomPoints(): int
#getFireFlowerPoints(): int
#isBreakBlock(): boolean
#canThrowFireball(): boolean
#isSuper(): boolean
NormalState
(from characterStates)
+NormalState(character: Character, stateSprites: HashMap)
+damaged(): void
+getSprites(): HashMap
#getStarPoints(): int
#getMushroomPoints(): int
#getFireFlowerPoints(): int
#isSuper(): boolean
#canThrowFireball(): boolean
SuperState
(from characterStates)
+SuperState(character: Character, stateSprites: HashMap)
+damaged(): void
+getSprites(): HashMap
#getStarPoints(): int
#getMushroomPoints(): int
#getFireFlowerPoints(): int
#canThrowFireball(): boolean
#isSuper(): boolean
Animations
(from entities)
#character: Character
+CharacterAnimations(character: Character)
+deathAnimation(): void
+superAnimation(actualState: String, newState: String): void
FireState
(from characterStates)
+FireState(character: Character, stateSprites: HashMap)
+getSprites(): HashMap
#getFireFlowerPoints(): int
+canThrowFireball(): boolean
CharacterThread
(from character)
#game: Game
#character: Character
#keyboard: Keyboard
-characterCollisionManager: CharacterCollisionManager
-fireBallCollisionManager: FireBallCollisionManager
-isRunning: boolean
-frameCount: int
-spriteNumber: int
-platformsByCoords: HashMap
-spacebarWasPressed: boolean
+CharacterThread(keyboard: Keyboard, game: Game)
+run(): void
-checkEnemiesInRange(enemyList: Enemy[0..*]): void
-moveCharacter(horizontalDirection: String, verticalDirection: String, spacebar: String)
-setSpacebarWasPressed(pressed: boolean): void
-moveRight(): void
-moveLeft(): void
-setIsRunning(value: boolean): void
Question
(from question)
-isBreakable: boolean = false
#actualState: QuestionState
#questionStates: HashMap
+Question(sprite: Sprite, positionX: int, positionY: int, states: HashMap)
+setPowerUp(powerUp: PowerUp): void
+setPowerUp(coin: Coin): void
+Manage(soundObserver: SoundObserver): int
+isEmpty(): boolean
+acceptVisit(visitor: CharacterVisitor): void
+ChangeToVoidQuestionBlock(): void
+getPower(p): PowerUp
QuestionBlockEmpty
(from question)
+constructor(+QuestionBlockEmpty(this: int, newSprite)
+damaged(): int
QuestionState
(from question)
#this: int
#powerUpOn: PowerUp
#sprite: Sprite
+QuestionState(this: int, newSprite: Sprite)
+setPower(powerUp: PowerUp): void
+getSprite(): Sprite
+isEmpty(): boolean
+damaged(): int
+getPower(p): PowerUp
WithPowerUp
(from question)
+WithPowerUp(sprite: Sprite)
+damaged(): int
WithCoin
(from question)
+WithCoin(sprite: Sprite)
+damaged(): int
Platform
(from platforms)
#isBreakable: boolean
+Platform(sprite: Sprite, positionX: int, positionY: int, isBreakable: boolean)
+isBreakable(): boolean
+acceptVisit(visitor: PlatformsVisitor): void
+visit(enemy: Enemy)
VoidBlock
(from platforms)
-pointsOnKill: int = -15
-isBreakable: boolean = false
+VoidBlock(sprite: Sprite, positionX: int, positionY: int)
+getPointOnKill(): int
+acceptVisit(visitor: CharacterVisitor): void
+visit(enemy: Enemy): void
Flag
(from platforms)
-isBreakable: boolean = false
+Flag(sprite: Sprite, positionX: int, positionY: int)
+acceptVisit(visitor: CharacterVisitor): void
Mast
(from platforms)
-isBreakable: boolean = false
+Mast(sprite: Sprite, positionX: int, positionY: int)
+acceptVisit(visitor: CharacterVisitor): void
MastEnd
(from platforms)
-isBreakable: boolean = false
+MastEnd(sprite: Sprite, positionX: int, positionY: int)
+acceptVisit(visitor: CharacterVisitor): void
+setObserver(observer: GraphicObserver): void
Block
(from platforms)
-isBreakable: boolean = false
+Block(sprite: Sprite, positionX: int, positionY: int)
+acceptVisit(visitor: CharacterVisitor): void
Pipe
(from platforms)
-isBreakable: boolean = false
+Pipe(sprite: Sprite, positionX: int, positionY: int)
+acceptVisit(visitor: CharacterVisitor): void
PipeTopLeft
(from platforms)
+PipeTopLeft(sprite: Sprite, positionX: int, positionY: int)
+acceptVisit(visitor: CharacterVisitor): void
PipeTopRight
(from platforms)
+PipeTopRight(sprite: Sprite, positionX: int, positionY: int)
+acceptVisit(visitor: CharacterVisitor): void
PipeBottomLeft
(from platforms)
+PipeBottomLeft(sprite: Sprite, positionX: int, positionY: int)
+acceptVisit(visitor: CharacterVisitor): void
PipeBottomRight
(from platforms)
+PipeBottomRight(sprite: Sprite, positionX: int, positionY: int)
+acceptVisit(visitor: CharacterVisitor): void
Brick
(from platforms)
+Brick(sprite: Sprite, positionX: int, positionY: int)
+breakBrick(): void
+acceptVisit(visitor: CharacterVisitor): void
EnemyVisitor
(from visitor)
+visit(enemy: Enemy): void
PlatformsVisitor
(from visitor)
+visit(block: Platform): void
VisitedPlatform
(from visitor)
+acceptVisit(visitor: PlatformsVisitor): void
VisitedElement
(from visitor)
+acceptVisit(visitor: CharacterVisitor): void
EnemyEntity
(from enemies)
+moveRight(frame: int): void
+moveLeft(frame: int): void
+getPointsOnDeath(): int
+getPointsOnKill(): int
+acceptVisit(visitor: EnemyVisitor): void
Enemy
(from enemies)
#pointsOnDeath: int
#pointsOnKill: int
#isBusy: boolean
#isActive: boolean
#isAlive: boolean
#isInAir: boolean
#files: boolean
#horizontalSpeed: float
#verticalSpeed: float
#sprites: HashMap
+Enemy(sprite: Sprite, positionX: float, positionY: float, pointsOnDeath: int, pointsOnKill: int)
+move(frame: int): void
+moveRight(frame: int): void
+moveLeft(frame: int): void
+isAlive(): boolean
+setAliveIsAlive: boolean): void
+setDirection(newDirection: String): void
+getDirection(): String
+pointsOnDeath(): int
+getPointsOnKill(): int
+getObserver(): Observer
+setIsBusy(): boolean
+activateEnemy(): void
+deactivateEnemy(): void
+getSprites(Map): HashMap
+setSpritesMap(spritesMap: HashMap): void
+getHorizontalSpeed(): float
+getVerticalSpeed(): float
+setVerticalSpeed(verticalSpeed: float): void
+isInAir(): boolean
+setIsInAirValue: boolean): void
+dead(): void
+applyGravity(): void
+acceptVisit(visitor: EnemyVisitor): void
+visit(enemy: Enemy): void
+leftEnemyCollision(enemy: Enemy): void
-rightEnemyCollision(enemy: Enemy): void
+getFlies(): boolean
+canThrowEgg(): boolean
+visit(block): void
KoopaTroopa
(from koopaTroopa)
+move(frame: int): void
+moveLeft(frame: int): void
+hit(character: Character): void
+moveRight(frame: int): void
+applyGravity(): void
+setSpriteNumber(spriteNumber: int): void
BuzzyBeetle
(from buzzyBeetle)
+move(frame: int): void
+moveLeft(frame: int): void
+moveRight(frame: int): void
+applyGravity(): void
+hit(character: Character): void
+setIsBusy(): boolean
+setSpriteNumber(spriteNumber: int): void
SpinyNormalState
(from spiny)
+NormalSpiny(spiny: Spiny)
+moveRight(frame: int): void
+moveLeft(frame: int): void
+applyGravity(): void
+setPlatform(): void
SpinyEggState
(from spiny)
+SpinyEgg(spiny: Spiny)
+moveRight(frame: int): void
+moveLeft(frame: int): void
+applyGravity(): void
+setPlatform(): void
PiranhaPlant
(from enemies)
-pointsOnDeath: int = 30
-pointsOnKill: int = -30
#spawnY: final float
#stopTime: long
+PiranhaPlant(sprite: Sprite, positionX: float, positionY: float)
+move(frame: int): void
+moveUp(frame: int): void
+moveDown(frame: int): void
-checkDirectionChange(): void
+applyGravity(): void
+getPointsOnDeath(): int
+getPointsOnKill(): int
+acceptVisit(visitor: CharacterVisitor): void
+setObserver(observer: GraphicObserver): void
EnemyCollisionManager
(from enemies)
#platforms[*] {collection="List"}
#game
+constructor(+EnemyCollisionManager(game)
+platformsCollisions(enemy: Enemy): void
-collisionWithPlatform(enemy: Enemy, platform): void
-downPlatformCollision(enemy: Enemy, platform): void
-leftPlatformCollision(enemy: Enemy, platform): void
-rightPlatformCollision(enemy: Enemy, platform): void
+enemiesCollisions(enemySource: Enemy): void
+powerUpsCollisions(entity: Enemy): void
-collisionWithOtherEnemy(enemySource: Enemy, enemyDestination: Enemy): void
EnemyThread
(from enemies)
#enemies: List[Enemy]
-ecm: EnemyCollisionManager
#frameCount: int
-spriteNumber: int
-platformsByCoords: HashMap
-isRunning: boolean
-game: Game
+EnemyThread(game: Game)
+run(): void
+setIsRunning(value: boolean): void
-moveEnemy(enemy: Enemy): void
-checkEnemyInAir(enemy: Enemy): void
BuzzyBeetle
(from enemies)
-pointsOnDeath: int = 30
-pointsOnKill: int = -25
#buzzyBeetleActualState: BuzzyBeetleState
+BuzzyBeetle(sprite: Sprite, positionX: int, positionY: int)
-getBuzzyBeetleStates(): HashMap
+moveRight(frame: int): void
+moveLeft(frame: int): void
+hit(): void
+isInMoveMode(): void
+isHeMode(): void
+getPointsOnDeath(): int
+getPointsOnKill(): int
+acceptVisit(visitor): void
BuzzyBeetleNormalState
(from buzzyBeetle)
+BuzzyBeetleNormalState(buzzyBeetle: BuzzyBeetle)
+moveRight(frame: int): void
+moveLeft(frame: int): void
+hit(): void
BuzzyBeetleShellState
(from buzzyBeetle)
+BuzzyBeetleShellState(buzzyBeetle: BuzzyBeetle)
+moveLeft(frame: int): void
+moveRight(frame: int): void
+hit(): void
Goomba
(from enemies)
-pointsOnDeath: int = 60
-pointsOnKill: int = -30
+Goomba(sprite: Sprite, positionX: int, positionY: int)
+moveRight(frame: int): void
+moveLeft(frame: int): void
+getPointsOnDeath(): int
+getPointsOnKill(): int
+acceptVisit(visitor: CharacterVisitor): void
+setObserver(observer: GraphicObserver): void
Lakitu
(from enemies)
-pointsOnDeath: int = 60
-pointsOnKill: int = 0
#characterReference: Character
#arrivalTimeMills: long
#characterSide: String
#waiting: boolean
#travelTime: float
#canThrowEgg: boolean
#eggCooldown: int
+Lakitu(sprite: Sprite, positionX: int, positionY: int)
+move(frame: int): void
-updateCooldown(): void
-changeDirectionToDestination(destinationX: float): void
-doDestinationCoordsDestinationX: float): boolean
-checkThrowTime(): boolean
-switchSide(): void
-updateHorizontalSpeed(): void
+moveRight(frame: int): void
+moveLeft(frame: int): void
+setCharacterReference(character: Character): void
+getPointsOnDeath(): int
+getPointsOnKill(): int
+acceptVisit(visitor: CharacterVisitor): void
+setObserver(observer: GraphicObserver): void
+canThrowEgg(): boolean
KoopaTroopaNormalState
(from koopaTroopa)
+KoopaTroopaNormalState(koopaTroopa: KoopaTroopa)
+move(frame: int): void
+moveLeft(frame: int): void
+moveRight(frame: int): void
+hit(character: Character): void
+isEnemyJeremy: Enemy): void
KoopaTroopaShellState
(from koopaTroopa)
+KoopaTroopa: KoopaTroopa
+spawnTimeMills: long
-isSending: boolean
+KoopaTroopaShellState(koopaTroopa: KoopaTroopa)
+move(frame: int): void
+moveRight(frame: int): void
+moveLeft(frame: int): void
+hit(character: Character): void
-checkSideCollision(character: Character): String
+updateShell(): void
-setShellSprite(): void
-updateSpawnTime(): void
-checkChangeState(): void
+hitEnemy(enemy: Enemy): void
Level/Generator
(from factories)
#entityFactory: EntityFactory
#parser: Parser
#levelNumber: int
+LevelGenerator(mode: String)
+CreateLevel(number: int, character: Character): Level
-generateEntity(character: Character, platformList: Platform[*], enemyList: Enemy[*], powerUpList: PowerUp[*]): void
+CreateCharacter(): Character
+getNextLevelPreviousCharacter: Character): Level
+CreateFirstLevel: int, y: int, direction: String): Projectile
+haveNextLevel(): boolean
+getEntityFactory(): EntityFactory
Vector
(from factories)
#positionX: E
#positionY: E
#type: E
+Vector()
+isEmpty(): boolean
+setXposition: E): void
+setYposition: E): void
+setType(type: E): void
+getPositionX(): E
+getPositionY(): E
+getType(): E
Parser
(from factories)
#levelContent: Vector[Integer][*]
#path: String
#numberLevel: int
+Parser()
-getLevelContent(): void
-traduceAndAddCurrentRowReading: String[*]): void
-addLuckyCycleCycle: int, number: Integer, numbersTrio: Vector): void
+getType(): int
+getPosition(Y): int
-checkRemove(): void
+hasToRead(): boolean
+getLevelLevelNumber: int): void
EntityFactory
(from factories)
#spriteFactory: SpriteFactory
+EntityFactory(mode: String)
+newEnemy(type: int, worldX: int, worldY: int): Enemy
+newPowerUpWithQuestionBlock(type: int, worldX: int, worldY: int): Question
+newOnlyCoin(worldX: int, worldY: int): Coin
+newPlatform(type: int, worldX: int, worldY: int): Platform
+CreateCharacter(): Character
-createNormalSprites(): HashMap
-characterNormalSprites(): HashMap
-characterSuperSprites(): HashMap
-characterFireSprites(): HashMap
-characterInvincibleSprites(): HashMap
-characterSuperInvincibleSprites(): HashMap
-createQuestionSprites(): HashMap
-newSuperMushroom(worldX: int, worldY: int): SuperMushroom
-newGreenMushroom(worldX: int, worldY: int): GreenMushroom
-newStar(worldX: int, worldY: int): Star
-newCoin(worldX: int, worldY: int): Coin
-newFireFlower(worldX: int, worldY: int): FireFlower
-newKoopaTroopa(worldX: int, worldY: int): KoopaTroopa
-getKoopaTroopaSprites(): HashMap
-newBuzzyBeetle(worldX: int, worldY: int): BuzzyBeetle
-getBuzzyBeetleSprites(): HashMap
-newLakitu(worldX: int, worldY: int): Lakitu
-getLakituSprites(): HashMap
-newPiranhaPlant(worldX: int, worldY: int): PiranhaPlant
-getPiranhaPlantSprites(): HashMap
-newGoomba(worldX: int, worldY: int): Goomba
-getGoombaSprites(): HashMap
-newSpiny(worldX: int, worldY: int): Spiny
-getSpinySprites(): HashMap
-newPipeTopLeft(worldX: int, worldY: int): Pipe
-newPipeTopRight(worldX: int, worldY: int): Pipe
-newPipeBottomLeft(worldX: int, worldY: int): Pipe
-newPipeBottomRight(worldX: int, worldY: int): Pipe
-newVoid(worldX: int, worldY: int): VoidBlock
-newFlag(worldX: int, worldY: int): Flag
-newMast(worldX: int, worldY: int): Question
-newBrick(worldX: int, worldY: int): Brick
-newBlock(worldX: int, worldY: int): Block
-newMast(worldX: int, worldY: int): Mast
-newMastEnd(worldX: int, worldY: int): MastEnd
+newFireBall(type: int, x: int, y: int, direction: String): Fireball
-getFireBallSprites(): HashMap
SpriteFactory
(from factories)
#folderPath: String
#SpriteFactory(path: String)
+getBackgroundSprite(number: int): Sprite
+getCharacterSitSprite(direction: String): Sprite
+getCharacterRightSprite(name: int): Sprite
+getCharacterLeftSprite(name: int): Sprite
+getCharacterJumpingSprite(direction: String): Sprite
+getCharacterFlag(): Sprite
+getCharacterDead(): Sprite
+getSuperCharacterSitSprite(direction: String): Sprite
+getSuperCharacterRightSprite(name: int): Sprite
+getSuperCharacterLeftSprite(name: int): Sprite
+getSuperCharacterJumpingSprite(direction: String): Sprite
+getSuperCharacterFlag(): Sprite
+getSuperCharacterDead(): Sprite
+getFireCharacterSitSprite(direction: String): Sprite
+getFireCharacterRightSprite(name: int): Sprite
+getFireCharacterLeftSprite(name: int): Sprite
+getFireCharacterJumpingSprite(direction: String): Sprite
+getFireCharacterFlag(): Sprite
+getFireCharacterDead(): Sprite
+getCharacterInvincibleSitSprite(direction: String): Sprite
+getCharacterInvincibleRightSprite(name: int): Sprite
+getCharacterInvincibleLeftSprite(name: int): Sprite
+getCharacterInvincibleJumpingSprite(direction: String): Sprite
+getCharacterInvincibleDead(): Sprite
+getSuperCharacterInvincibleSitSprite(direction: String): Sprite
+getSuperCharacterInvincibleRightSprite(name: int): Sprite
+getSuperCharacterInvincibleLeftSprite(name: int): Sprite
+getSuperCharacterInvincibleJumpingSprite(direction: String): Sprite
+getSuperCharacterInvincibleDead(): Sprite
+getGoombaLeftSprite(): Sprite
+getGoombaRightSprite(): Sprite
+getKoopaTroopaLeftSprite(name: int): Sprite
+getKoopaTroopaRightSprite(name: int): Sprite
+getKoopaTroopaShellSprite(): Sprite
+getPiranhaPlantSprite(name: int): Sprite
+getLakituLeftSprite(): Sprite
+getLakituRightSprite(): Sprite
+getBuzzyBeetleLeftSprite(name: int): Sprite
+getBuzzyBeetleRightSprite(name: int): Sprite
+getBuzzyBeetleShellSprite(): Sprite
+getSpinyLeftSprite(name: int): Sprite
+getSpinyRightSprite(name: int): Sprite
+getSpinyEggSprite(): Sprite
+getShellSprite(): Sprite
+getSuperMushroomSprite(): Sprite
+getGreenMushroomSprite(): Sprite
+getFireFlowerSprite(): Sprite
+getStarSprite(): Sprite
+getCoinSprite(): Sprite
+getFireballSprite(name: int): Sprite
+getBrickSprite(): Sprite
+getQuestionBlockSprite(): Sprite
+getBlockSprite(): Sprite
+getQuestionEmptyBlock(): Sprite
+getPipeTopRightSprite(): Sprite
+getPipeTopLeftSprite(): Sprite
+getPipeBottomRightSprite(): Sprite
+getPipeBottomLeftSprite(): Sprite
+getFlagSprite(): Sprite
+getVoidSprite(): Sprite
+getMastSprite(): Sprite
+getMastEndSprite(): Sprite
Game
(from game)
#levelGenerator: LevelGenerator
#viewController: ViewController
#currentLevel: Level
#currentPlayerName: String
#numberLevel: int
#characterThread: CharacterThread
#enemyThread: EnemyThread
#ranking: Ranking
#mode: String
+Game()
+setName(name: String): void
+setViewController(viewController): void
+start(): void
+stop(): void
+setLevel(number: int): void
+getCurrentLevel(): Level
+getViewController(): ViewController
+setObservers(): void
+setCharacterObserver(character: Character): void
-setCharacterObserverOfSound(character: Character): void
+setPlatformsObservers(platformsList[*]): void
+setEnemiesObservers(enemyList[*]): void
+setPowerUpsObservers(powerUpsList[*]): void
+removeLogicalEntity(logicalEntity: LogicalEntity): void
+playNextLevel(): void
+reproduceSound(path: String): void
+reproduceSoundDeath(path: String): void
+stopSound(): void
#changeLevel(): void
-resetCharacter(): Character
+updateInformation(score: int, coins: int, timer: int, lives: int): void
+getRankingPlayers(): String[*]
+setMode(mode: String): void
+createFireBall(x: int, y: int, direction: String): void
+setProjectilesObservers(projectile: Projectile): void
+reproduceLoopSound(path: String, iterations: int): void
+startMusic(): void
+isRunningSound(): boolean
+stopMusic(): void
+createEggx: int, y: int): void
Level
(from game)
#remainingTime: int
#character: Character
#running: boolean
#platforms: Platform[*]
#enemies: Enemy[*]
#powerUps: PowerUp[*]
#projectiles: Projectile[*]
+Level(platforms[*], enemies[*], powerUps[*])
+setCharacter(character: Character): void
+start(): void
+end(): void
+getPlatforms(): Platform[*]
+getEnemies(): Enemy[*]
+getPowerUps(): PowerUp[*]
+getProjectiles(): Projectile[*]
+getCharacter(): Character
+getRemainingTime(): int
+isRunning(): boolean
+delete(): void
+addFireBall(projectile): void
Ranking
(from ranking)
-size: int
#path: String
#maxCapacity: int = 5
+Ranking()
-isFull(): boolean
-isEmpty(): boolean
getRanking(): void
-addToRank(name: String, score: int): void
-reorder(): void
-getRankTop(): void
-updateRanking(): void
-enterIntoRankingScore: int): boolean
+getPlayers(): Collection(String)
VectorRanking
(from ranking)
#name: K
#top: V
#score: V
+VectorRanking(name: K, top: V, score: V)
+getName(): K
+getTop(): V
+getScore(): V
+setTop(top: V): void
+toString(): String
SoundReproducer
(from game)
-audioLevelMusic: Clip
-aloniaiaAudio: Clip
-hashSounds: HashMap
+SoundReproducer(generator: SoundGenerator)
+setAuxiliarSound(path: String): void
+setMusicSound(path: String): void
+start(): void
+loop(iterations: int): void
+resourceAudioController(clip: Clip): void
+stopMusic(): void
+stopSoundAuxiliar(): void
+getSound(path: String): AudioInputStream
+isRunning(): boolean
SoundGenerator
(from factories)
#soundHash
#folderPath: String
+SoundGenerator()
+createSounds(): void
-getFileOfSounds(): HashMap
Original
(from factories)
#path: String
+Original(path: String)
Sprite
(from factories)
#skinPath: String
+Sprite(skinPath: String)
+getSkinPath(): String
Custom
(from factories)
#path: String
+Custom(path: String)
Launcher
(from launcher)
+mainLate_Stress: void
KoopaTroopaState
(from koopaTroopa)
+move(frame: int): void
+hit(character: Character): void
+hitEnemy(enemy: Enemy): void
BuzzyBeetleState
(from buzzyBeetle)
+moveLeft(frame: int): void
+moveRight(frame: int): void
+hit(): void
LevelScreen
(from views)
#viewController: ViewController
#contentPanel: JPanel
#backgroundImageLabel: JLabel
#informationPanel: InformationPanel
-backgroundX: int
+LevelScreen(viewController: ViewController)
#setBackgroundAndScroll(): void
-configureBackgroundLabel(): void
-configureContentPanel(): void
-getBackgroundIcon(path: String): null
+updateScrollRight(character): void
+getScrollleaXPosition(): int
+resetScrollBar(): void
+drawEntityCharacter(characterEntity): CharacterObserver
+drawLogicalEntity(entity, isActive: boolean): GraphicObserver
+drawLogicalEntity(entity): GraphicObserver
+removeEntity(entity): void
-addInformationPanel(): void
+updateInformationPanel(viewCoins: int, newScore: int, newTime: int, newLives: int): void
+drawHitbox(entity): void
+drawPoints(pointsLabel): void
+pointsAnimation(points:JLabel): void
Spiny
(from enemies)
-pointsOnDeath: int = 60
-pointsOnKill: int = 30
#spinyActualState: SpinyState
#spinyStates: HashMap
+Spiny(sprite, positionX: int, positionY: int)
+move(frame: int): void
+getPointsOnDeath(): int
+getPointsOnKill(): int
+acceptVisit(visitor: CharacterVisitor): void
+setObserver(observer: GraphicObserver): void
+changeToNormal(): void
+changeToNormalState(): void
-getSpinyStates(): null
+moveRight(frame: int): void
+moveLeft(frame: int): void
+applyGravity(): void
+setPlatform: Platform): void
ViewController
(from views)
#window: JFrame
#levelScreen: LevelScreen
#menuScreen: MenuScreen
#rankingScreen: RankingScreen
#preGameScreen: PreGameScreen
#game: Game
#keyboardInputs: Keyboard
+ViewController(game)
+configureWindow(): void
+newGame(mode: String, name: String): void
+showMenuScreen(): void
+showLevelScreen(): void
+showRankingScreen(): void
+showPreGameScreen(): void
+refresh(): void
+getKeyboard(): null
+registerEntity(character): GraphicObserver
+registerEntity(entity, isActive: boolean): GraphicObserver
+registerEntity(enemy): GraphicObserver
+removeLogicalEntity(e): void
+updateInformation(newCoins: int, newScore: int, newTime: int, newLives: int): void
+clearLevelScreen(): void
+endGame(): void
+getLevelScreen(): LevelScreen
+getPlayers(): String[*]
PreGameScreen
(from views)
-background
-nameInput
-modeSelector
#initGame
#backButton
-name: String
-mode: String
+PreGameScreen(view: ViewController)
-createNameInput(): void
-createModeSelector(): void
-createButton(): void
+setMode(mode: String): void
+setName(name: String): void
-addBackButton(): void
RankingScreen
(from views)
-backButton
+RankingScreen(viewController: ViewController)
-addRankingPanel(players: String[*]): void
-addBackButton(): void
GraphicTools
(from tools)
+leanOnVoid_first_observer(): void
+leanOnSpiny_first_observer(): void
+leanOnPlatformCollisionLeftEnemy_firstEnemy(): void
+leanOnCoinCollision_first_destination: int, destination: int, direction: int, firstEntity: int, secondEntity: int, first_destination: boolean, first_secondEntity: int, first_destination: boolean, first_secondEntity: int, first_destination: boolean, first_