

EmptyLevel

+ method(type): type

boss: Boss

Character

checkCharacterCollisionSpecific(obj: GameObject): void

attackDelay: Delay

+ update(): void + getInput(): void

attack(): void

checkDeath(): void

Player

checkCharacterCollisionSpecific(obj: GameObject): void

speed: float

attackDamage: int

checkCharacterCollision(): void

loseHealth(hit: int): void

#isDead: boolean

+ getHealth(): int

attack(): void

checkDeath(): void

health: int

Game Object

finit(xPos: float, yPos: float, width: int, height: int, imgPath: String): void

Door

+ field: type

+ update(): void

xPos: float

yPos: float

width: int

height: int

sprite: Sprite

+ update(): void

+ render(): void

remove(): void

getHeight(): int

getWidth(): int

- getX(): float

- getY(): float

isRemoved: boolean

+ setTexture(imgPath String): void

Items

ConsumableItem

addedHealth: int

+ getAddedHealth(): int

despawnDelay: Delay

+ update(): void

toRemove: boolean

- gameLoop(): void

+ getHeight(): int + getWidth(): int

Interface

+ init(filePath: String): void

MainMenu

- startButton: GameButton

- quitButton: GameButton

+ update(): void

+render(): vc₁d

init(fileName: String): void

- background: Sprite

+ render(): void

- playerHealthBar: Rectangle

- playerHealthFactor: float

- bossHealthBar: Rectangle

- PLAYER_HEALTH_BAR_WIDTH: int

- BOSS_HEALTH_BAR_WIDTH: int

- bossHealthFactor: float

showPlayerHealth(): void

- showBossHealth(): void

+ update(): void

+ render(): void

+ start(): void

+ quit(): void

Sprite

+ render(xPos: float, yPos: float): void

renderTex(textureID: int): void

+ setAlpha(a: float): boolean

+ setTexture(imgPath: String): void

PauseMenu

startButton: GameButton

quitButton: GameButton

+ init(fileName: String): void

+ update(): void

+render(): void

- a: float

- width: int

- height: int

textureID: int

renderRGB(): void

+ getWidth(): int

+ getHeight(): int

GameButton

btnTexture: String

- oldBtnTexture: String

onPressed: Runnable

- btnReleased: String

- btnPressed: String

+ update(): voic



checkCharacterCollisionSpecific(obj: GameObject): void

Boss

+ shootAbility(proWidth: int, proHeight: int, numberOfPros: int): void

+ targetPlayerAbility(proWidth: int, proHeight: int, numberOfPros: int): void + wallOfFireAbility(proWidth: int, proHeight: int, numberOfPros: int) void

Boss Level

+ getHealth(isPlayer: boolean): int

- boss: Boss

Projectile

shootAngle: Vector2f

damage: int

speed: float

update(): void

none

none

targetAttackDelay: DelayburstAttackDelay: Delay

+ update(): void

checkDeath(): void

+ attack(): void

- projectiles: ArrayList<StandardProjectile

getDamage(): int

StandardProjectile

TextureLoader

+ loadTexture(fileName: String): int

Physics

+ checkCollision(obj1: GameObject, obj2: GameObject): boolean

+ checkCollision(field: Rectangle, obj: GameObject): boolean

+ checkCollision(obj: GameObject, x: int, y: int): boolean

none

- endTime: long

- startTime: long

- hasStarted: boolean

+ isOver(): boolean

+ start(): void

+ stop(): void + restart(): void

+ x: float

+ y: float

+ length(): float

+ isStarted(): boolean

+ getCurrTime(): long

+ restart(time_ms: int): void

+ dot(vector: Vector2f): float

+ normalize(): Vector2f

+ negate(): Vector2f

+ angle(vector: Vector2f): float

+ rotate(angle: float): Vector2f

+ translate(x: float, y: float): Vector2f

+ add(vector: Vector2f): Vector2f

+ sub(vector: Vector2f): Vector2f

+ mul(vector: Vector2f): Vector2f

+ div(vector: Vector2f): Vector2f

+ equals(vector: Vector2f): boolean

+ add(amt: float): Vector2f

+ sub(amt: float): Vector2f

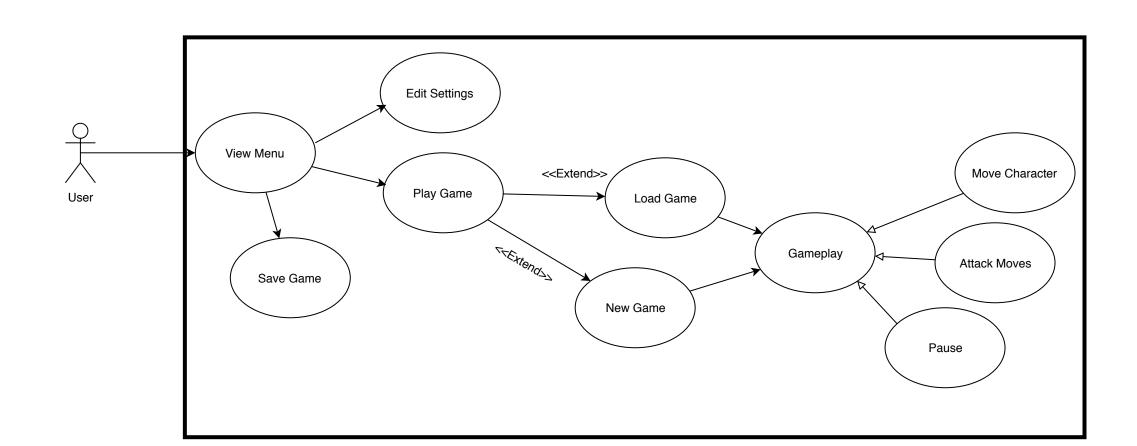
+ mul(amt: float): Vector2f

+ div(amt: float): Vector2f

+ abs(): Vector2f

+ toString(): String

+ getX(): float + getY(): float Vector2f



State Diagram

