Object

Object

+ objects: vector<Object*>
+ *drawTo(sf::RenderWindow&): void*
+ *fall(): void*
+ *getRect(): sf::FloatRect*
+ Object()
+ ~Object()

Listener

Listener

+ listeners: vector<Listener*>
+ Listener()
+ ~Listener()
+ *onUpdate(): void*

AnimManager

AnimManag...

- animBack: bool
- animes: vector<sf::Te...
- index: int
- rect: sf::IntRect
- scale: sf::Vector2f
- size: sf::Vector2i
- sprite: sf::Sprite*
+ addAnim(string): void
+ anim(): void
+ AnimManager(sf::Spr...
+ ~AnimManager()
+ setScale(sf::Vector2f)...
+ setSize(sf::Vector2i): ...

-stand

Build

Build

- color: sf::Color
- position: sf::Vector2f
- rect: sf::RectangleShape
- texture: sf::Texture*
+ Build(sf::Vector2f, sf::Vector2f, sf::Color)
+ Build(sf::Vector2f, sf::Vector2f, sf::Texture...
+ ~Build()
+ *drawTo(sf::RenderWindow&): void*
+ *fall(): void*
+ *getRect(): sf::FloatRect*

Character

Character

attackSpeed: float
attackTimer: sf::Clock
+ characters: vector<Character*>
# damage: float
# finalJumpHeight: float
# health: int
# isJumping: bool
# jumpHeight: float
# maxSlideX: float
# maxSlideY: float
# moveSpeed: float
# score: int
# velocity: sf::Vector2f
+ Character()
+ ~Character()
+ *drawTo(sf::RenderWindow&): void*
+ *getPos(): sf::Vector2f*
+ *getRect(): sf::FloatRect*
+ *move(sf::Vector2f): void*
+ *takeDamage(float): void*

Plataform

Plataform

- color: sf::Color
- move: sf::Vector2f
- moveActual: sf::Vector2f
- moving: bool
- mRight: bool
- mUp: bool
- originalPos: sf::Vector2f {readOnly}
- rect: sf::RectangleShape
- texture: sf::Texture*
- vel: sf::Vector2f
+ *drawTo(sf::RenderWindow&): void*
- fmove(): void
+ *getRect(): sf::FloatRect*
+ *onUpdate(): void*
+ Plataform(sf::Vector2f, sf::Vector2f, sf::Color)
+ Plataform(sf::Vector2f, sf::Vector2f, sf::Texture...
+ ~Plataform()
+ setMove(sf::Vector2f): void
+ setMoveSpeed(sf::Vector2f): void
+ setMoving(bool): void

Enemy

Enemy

enemy: sf::RectangleShape
mLeft: bool
moving: sf::Vector2f
name: string
originalPos: sf::Vector2f
RespawnPos: sf::Vector2f
+ *drawTo(sf::RenderWindow&): void*
+ Enemy(sf::Vector2f, string)
+ ~Enemy()
+ *fall(): void*
+ *getPos(): sf::Vector2f*
+ *getRect(): sf::FloatRect*
+ *move(sf::Vector2f): void*
+ moveLeft(): void
+ moveRight(): void
+ *onUpdate(): void*
+ setPos(sf::Vector2f): void
+ *takeDamage(float): void*

Menu

- font: sf::Font
- menu_text: sf::Text ([NUM_ITEMS])
- selectedItem: int

+ draw(sf::RenderWindow&): void
+ getEnter(): int
+ Menu(float, float)
+ ~Menu()
+ moveDown(): void
+ moveUp(): void

World

- background: sf::Sprite*

+ getBackground(): sf::Sprite*
+ setBackground(sf::Sprite*): void
+ World()
+ ~World()

ColisionManager

+ ColisionManager()
+ ~ColisionManager()
+ intersectsDown(sf::FloatRect): bool
+ intersectsLeft(sf::FloatRect): bool
+ intersectsRight(sf::FloatRect): bool
+ intersectsUp(sf::FloatRect): bool
+ moveChars(Object*, sf::Vector2f): void

-menu

-world

-run

Engine

- menu: Menu
- menuEnabled: bool
- view: sf::View&
- window: sf::RenderWindow&
- world: World*

+ addObject(Object*): void
+ draw(sf::RenderWindow&): void
+ Engine(sf::RenderWindow&, sf::View&)
+ ~Engine()
+ getView(): sf::View& {query}
+ getWindow(): sf::RenderWindow& {query}
+ gravity(): void
+ setWorld(World*): void
+ update(): void

Player

isMoving: bool
isMovingRight: bool
name: string
pSprite: sf::Sprite
RespawnPos: sf::Vector2f
run: AnimManager
spriteClock: sf::Clock
stand: AnimManager
view: sf::View&

---

attack(): void
drawTo(sf::RenderWindow&): void
fall(): void
getPos(): sf::Vector2f
getRect(): sf::FloatRect
jump(): void
move(sf::Vector2f): void
moveLeft(): void
moveRight(): void
onUpdate(): void
Player(sf::View&)
~Player()
setPos(sf::Vector2f): void
takeDamage(float): void

TextureManager

+	grass: sf::Texture*
+	playerAnim: sf::Texture*

+	getTexture(string): sf::Texture
-	TextureManager()
-	~TextureManager()