HitboxComponent

- sprite: sf::Sprite&

- offsetX: float

- offsetY: float

- hitbox: sf::RectangleShape

+ HitboxComponent(sprite: sf::Sprite&, offset\_x: float, offset\_y: float, width: float, height: float)

+ ~HitboxComponent()

+ checkIntersect(frect: const sf::FloatRect&): bool

+ update(): void

+ render(target: sf::RenderTarget&): void

GameState

- window: sf::RenderWindow\*

- supportedKeys: std::map<std::string, int>\*

- states: std::stack<State\*>\*

- keybinds: std::map<std::string, int>

- textures: std::map<std::string, sf::Texture>

- player: Player\*

+ GameState(sf::RenderWindow\*, std::map<std::string, int>\*, std::stack<State\*>\* )

+ ~GameState()

+ updateInput(const float&)

+ update(const float&)

+ render(sf::RenderTarget\*)

- initializeKeybinds()

- initializeTextures()

- initializePlayers()

class Game

\* window: sf::RenderWindow\*

\* windowSettings: sf::ContextSettings

\* supportedKeys: std::map<std::string, int>

\* states: std::stack<State\*>

\* fullscreen: bool

\* videoModes: std::vector<sf::VideoMode>

\* dt: float

\* dtClock: sf::Clock

\* sfEvent: sf::Event

- initializeVariables()

- initializeWindow()

- initializeKeys()

- initializeStates()

+ Game()

+ ~Game()

+ updateDt()

+ updateSFMLEvents()

+ endApplication()

+ update()

+ render()

+ run()

EditorState

- font: sf::Font

- buttons: std::map<std::string, Button\*>

- initializeVariables(): void

- initializeBackground(): void

- initializeFonts(): void

- initializeKeybinds(): void

- initializeButtons(): void

- updateInput(dt: const float&): void

- updateButtons(): void

- renderButtons(target: sf::RenderTarget&): void

- update(dt: const float&): void

- render(target: sf::RenderTarget\*): void

Entity

sprite: sf::Sprite

hitboxComponent: HitboxComponent\*

movementComponent: MovementComponent\*

animationComponent: AnimationComponent\*

initializeVariables(): void

setTexture(texture: sf::Texture&): void

createHitboxComponent(sprite: sf::Sprite&, offset\_x: float, offset\_y: float, width: float, height: float): void

createMovementComponent(maxVelocity: const float, acceleration: const float, deceleration: const float): void

createAnimationComponent(texture\_sheet: sf::Texture&): void

setPosition(x: const float, y: const float): void

move(dir\_x: const float, dir\_y: const float, dt: const float&): void

update(dt: const float&): void

render(target: sf::RenderTarget&): void

Button

shape: sf::RectangleShape |

font: sf::Font\*

text: sf::Text

textIdleColor: sf::Color

textHoverColor: sf::Color

textActiveColor: sf::Color

idleColor: sf::Color

hoverColor: sf::Color

activeColor: sf::Color

buttonState: int

Button(x: float, y: float, width: float, height: float, font: sf::Font\*, text: std::string, character\_size: unsigned, text\_idle\_color: sf::Color, text\_hover\_color: sf::Color, text\_active\_color: sf::Color, idle\_color: sf::Color, hover\_color: sf::Color, active\_color: sf::Color)

~Button()

isPressed(): const bool

update(mousePos: const sf::Vector2f): void

render(target: sf::RenderTarget&): void

AnimationComponent

sprite: sf::Sprite&

textureSheet: sf::Texture&

lastAnimation: Animation\*

priorityAnimation: Animation\*

animations: std::map<std::string, Animation\*

addAnimation(key: std::string, animation\_timer: float, start\_frame\_x: int, start\_frame\_y: int, frames\_x: int, frames\_y: int, width: int, height: int): void

play(key: std::string, dt: const float&, priority: const bool): void

play(key: std::string, dt: const float&, modifier: const float&, modifier\_max: const float&, priority: const bool): void

| Player |

| - sprite: sf::Sprite|

| - hitboxComponent: HitboxComponent\* |

| - movementComponent: MovementComponent\* |

| - animationComponent: AnimationComponent\* |

| + Player(x: float, y: float, texture\_sheet: sf::Texture&) |

| + ~Player() |

| + update(dt: const float&) : void |

| - initializeVariables() : void |

| - initializeComponents() : void |

MovementComponent

- sprite: sf::Sprite&

- velocity: sf::Vector2f

- maxVelocity: float

- acceleration: float

- deceleration: float

+ MovementComponent(sprite: sf::Sprite&, maxVelocity: float, acceleration: float, deceleration: float)

+ ~MovementComponent()

+ getMaxVelocity(): const float&

+ getVelocity(): const sf::Vector2f&

+ getState(state: short unsigned): const bool

+ move(dir\_x: float, dir\_y: float, dt: const float&)

+ update(dt: const float&)

MainMenuState

-window: sf::RenderWindow\*

-supportedKeys: std::map<std::string, int>\*

-states: std::stack<State\*>\*

-background: sf::RectangleShape

-backgroundTexture: sf::Texture

-font: sf::Font

-mousePosView: sf::Vector2f

-buttons: std::map<std::string, Button\*>

-keybinds: std::map<std::string, int>

+MainMenuState(window: sf::RenderWindow\*, supportedKeys: std::map<std::string, int>\*, states: std::stack<State\*>\*)

+~MainMenuState()

+updateInput(dt: const float&)

+updateButtons()

+update(dt: const float&)

+renderButtons(target: sf::RenderTarget&)

+render(target: sf::RenderTarget\* = nullptr)

-initializeVariables()

-initializeBackground()

-initializeFonts()

-initializeKeybinds()

-initializeButtons()

State

| - window: sf::RenderWindow\* |

| - supportedKeys: std::map<std::string, int>\* |

| - states: std::stack<State\*>\* |

| - quit: bool |

| - mousePosScreen: sf::Vector2i |

| - mousePosWindow: sf::Vector2i |

| - mousePosView: sf::Vector2f |

| + State(window: sf::RenderWindow\*, supportedKeys: std::map<std::string, int>\*, states: std::stack<State\*>\*) |

| + virtual ~State() |

| + const bool& getQuit() const |

| + void endState() |

| + void updateMousePositions() |

| ~ virtual void update(const float& dt) = 0 |

| ~ virtual void render(sf::RenderTarget\* target = nullptr) = 0 |

| ~ virtual void handleInput(const float& dt, sf::Event event) = 0 |