## Game -window: sf::RenderWindow StateInteger -title\_font: sf::Font +StateInteger: enum -game\_font: sf::Font state1 -num\_font: sf::Font state2 state3 **Animations** -title: sf::Text GameState state4 -title colour: sf::Color +playerState: PlayerState state5 +stateInteger: StateInteger +gameState: enum -play\_option: sf::Text MAINMENU -credits option: sf::Text STORY -spritesheet: sf::Texture -quit\_option: sf::Text **TUTORIAL** -menu\_integer: int -animated\_sprite: sf::Sprite LEVEL1 -pixels\_height: int -credits\_title: sf::Text -pixels\_width: int LEVEL2 -credits text: sf::Text -current frame: sf::IntRect LEVEL3 -animation\_timer: sf::Clock VICTORY1 -credits\_exit: sf::Text VICTORY2 PlayerState VICTORY3 -story: sf::Text +animationsInitialise(): void **GAMEOVER** -greyscale: sf::Color +animationUpdate: void +PlayerState: enum **CREDITS** -story\_continue: sf::Text +stateChange: void IDLE -story\_done: bool RUNNING -conclusion: sf::Text **JUMPING** -thank\_you: sf::Text CROUCHING -conclusion\_done: bool ATTACKING -tutorial move: sf::Text -tutorial\_sword: sf::Text -tutorial\_counters: sf::Text **Player** -title\_background: GameObject -title\_background\_texture: sf::Texture +isgrounded: bool -level1\_background: GameObject +is\_running: bool -level1\_background\_texture: sf::Texture +is\_crouching: bool -level1\_text: sf::Text TextSetup +sword\_equipped: bool -level2\_background: GameObject +attacking: bool -level2\_background\_texture: sf::Texture +TextSetup(): void +jump timer: sf::Clock -level2 text: sf::Text -level3\_background: GameObject +attack\_range: GameObject -level3 background texture: sf::Texture +range test: sf::Texture -level3\_text: sf::Text +facing right: bool -level no: int player\_spritesheet: sf::Texture -player: Player -sound: Sound -running\_timer: sf::Clock -sword: GameObject -sword texture: sf::Texture +playerInitialise(): bool -sword\_light: GameObject +keyReleased(): void -light\_texture: sf::Texture +mouseReleased(): void -invincible timer: sf::Clock +update(): void -invincible anim: sf::Clock +setGrounded(): void -isInvincible: bool +playerRender(): void +attackingRange(): void -\*tiles\_array: GameObject +playerAudioInit(): void -tiles\_texture: sf::Texture -\*coins\_array: GameObject -coins texture: sf::Texture -\*enemy\_array: GameObject -enemy\_texture: sf::Texture -enemy\_counter: int -coin\_counter: int Sounds -defeat\_title: sf::Text +coin\_sound: sf::Sound -defeat text: sf::Text +coin\_sound\_buffer: sf::SoundBuffer -defeat\_quit: sf::Text +get\_sword\_sound: sf::Sound -victory\_title: sf::Text +get\_sword\_sound buffer: sf::SoundBuffer -final\_score\_text: sf::Text +attack\_sound: sf::Sound -final\_score\_calc: sf::Text +attack sound buffer: sf::SoundBuffer -continue\_text: sf::Text +death\_sound: sf::Sound -cont\_yes: sf::Text +death sound buffer: sf::SoundBuffer -cont\_no: sf::Text +jump\_sound: sf::Sound -continue selected: bool +jump sound buffer: sf::SoundBuffer +menu\_sound: sf::Sound -hud: GameObject +menu\_sound\_buffer: sf::SoundBuffer -hud\_texture: sf::Texture -score: int +title\_music: sf::Sound -score\_display: sf::Text +title music buffer: sf::SoundBuffer -score\_symbol: GameObject +game music: sf::Sound -score texture: sf::Texture +game music buffer: sf::SoundBuffer -lives: int -lives display: sf::Text -lives\_symbol: GameObject -lives texture: sf::Texture -sound: Sound GameData +init(): bool +ground\_height: const int +gravity\_speed: float +update(): void +dt: float +render(): void +renderArrays(): void +keyPressed(): void +kevReleased(): void +mouseReleased(): void +collision(): void +blocksCollision(): void +coinsCollision(): void +enemyCollision(): void +textInitialise(): bool +titleScreenInitialise(): bool +spritesInitialise(): bool +audioInitialise(): bool +blocksInitialise(): bool +coinsInitialise(): bool +enemylnitialise(): bool

+set\_to\_levelstart(): bool +set\_to\_default(): bool

GameObject

+visibility: bool

+direction: Vector

#sprite: sf::Sprite\*

+update(): virtual void

+initialiseSprite(): bool

+getSprite(): sf::Sprite\*

Vector

+enemyVectorL(): void +enemyVectorR(): void

+invertX(): void

+x: float

+y: float

+getVector(): Vector\*

+collision(): bool

+speed: int