#### MoveComponent

+MoveComponent(const char\*) +getVelocity(): MathLibrary::Vector2 +setVelocity(MathLibrary::Vector2):

+getMovedRightLast() {return m\_movedRightLast3: bool

+aetMaxSpeed(): float +setMaxSpeed(float): void update(float): void override

-m maxSpeed: float -m\_velocity: MathLibrary::Vector2

> -m\_maxX : int -m maxY: int

-m mixX: int

-m\_mixY: int

#### InputComponet

+InputComponet(Actor\*). +getMoveAxis(): MathLibrary::Vector2 +update(float): void override +aetAttack[put(): bool +getGorwardAxis(): MathLibrary::Vector2

-m\_killPlayer: bool -m\_Up: RAYLIB\_H::KeyboardKey -m\_Down: RAYLIB\_H::KeyboardKey m\_Right: RAYLIB\_H::KeyboardKey m Left: RAYLIB H::KevboardKev -m\_meleeSwing:RAYLIB\_H::KeyboardKey

# FollowComponent

+FollowComponent(Actor\*, const char\*) +getMoveAxis(): MathLibrary::Vector2 +update(float deltaTIme) override

-m\_targetActor: Actor\* -m\_moveComponent: MoveComponnt\*

# Component

## AttackComponet

+AttackComponet(float, int) +update(flaot): void: override +getCharcaterOwner(): Character\* +getisAttacking() {return m\_isAttacking} : bool +setIsAttacking(bool): void -m\_attackActor: Actor\*

> -m\_attackStarted: bool -m attackFrames: int

-m\_isAttacking: bool -m\_timeTracher: float

-m\_fps: flaot

#### HealthComponet

+HealthComponet(int) +takeDamage(): void +getHealth() {return m\_currentHealth}

+setHealth(int): void +getTiemSinceDamage() {return +m\_timeSinceDamage3:float +getDamageCoolDown() { return m\_damageCooldown3: float +start(): override void +update(float deltaTIme): override void

+onColllision(Actor\* actor): override void -m\_maxHealth: int -m\_currentHealth: int

> -m damageCooldown: float -m\_timeSinceDamage: float

#### SpriteComponent

+SprteComponet(const car\*, const char\*) +SprteComponet(Texture2D\*, const char\*) +~SprteComponet(): override

+setTextur(Textur2D\*): void +getWidth() {return m\_width}: int +getHeight() {return m\_height} : int +update(float): virtual void: override +draw(): virtual void: override

> -m texture: Texture2D\* -m\_width: int -m\_height: int

### AnimatedSpriteComponet

+AnimatedSpriteComponet(const char\*, const char\*, int, int, float)

+getFrameRec() {return &m\_frameRec} : Rectangle +getCharacter() {3return Character\*} : Character\* +getTimeTracher() {returns m\_timeTracher} : float +getFps() {returns m\_fps} : float +getCurrentXFrame() {m\_currentXFrame} : int +getCurrentYFrame() {m\_currentYFrame} : int +getXFrameCount() &m\_xFrameCount3: int +getYFrameCount() &m\_yFrameCount3: int +getRightTexture() &m\_rightTexture3: Texture2D\* +aetLeftTexture() {m leftTexture} : Texture2D\* +getIsTextureLeft() {m\_textureIsLeft} : bool +setTimeTracher(float) {m\_timeTracher} : void +setFps(float) &m\_fps3: void +setCurrentXFrame(int) {m currentXFrame} : void +setCurrentYFrame(int) {m\_currentYFrame} : void +setIsTextureLeft(bool) {m\_textureIsLeft}: void +setStartAttack(bool) & m startAttack 3: void +setCharacterIsDead(bool) { m\_characterIsDead} : void +getCharacterIsDead() { return m\_characterIsDead} : bool

> +update(float): void override +draw(): virtual void override +aetCurrentFrames(): void +updteFrames(float): virtual void

-m\_rightTexture: Texture2D\* -m\_leftTexture: Texture2D\* -m texture[s] eft: bool -m\_characterIsDead: bool -m startAttack: bool -m\_timeTracher: float -m\_fps:float -m\_currentXFrame: int -m currentYFrame: int -m\_xFrameCount: int -m\_yFrameCount: int

# EnemyAnimationsComponent: AnimatedSpriteComponent

+EnemyAnimationsComponent(const char\*, const char\*, int, int, float) +updateFrames(float): override void +draw(): override void

#### PLayerAnimationsComponent: AnimatedSpriteComponent

+PI averAnimationsComponent(const char\*, const char\*, int, int, float) +updateFrames(float): override void +draw(): override void