J1App

J1Module

Entity

Animation* :animation

ENTITY_TYPE: type

SDL_Texture* :texture

SDL_Renderer:Flip

fPoint: pos

bool: to_destroy

Entity(int x, int y, ENTITY_TYPE type)

virtual ~Entity()

virtual bool :Update(float dt)

virtual bool : Load(pugi::xml_node& data)

virtual bool : Save(pugi::xml_node& data) const

virtual void :LoadTexture()

virtual void : FindEntitySpawn(ENTITY_TYPE type)

virtual void :SpawnEntity()

int: GetEntityTile(fPoint pos) const

virtual void: Draw()

virtual COLLISION_TYPE: CheckCollision(int x) const

ModuleEntities

p2List<Entity*>:entities

Player:player

ModuleEntities()

~ModuleEntities()

bool:Awake(pugi::xml_node&)

bool: Start()

bool: PreUpdate()

bool: Update(float dt)

bool: CleanUp

Player*: GetPlayer const

bool: SpawnEntity(int x, int y, ENTITY_TYPE type)

1

0..*

ModulePlayer

Animation* animation

Animation idle: Animation()
Animation running: Animation()
Animation jumping: Animation()
Animation die: Animation()
Animation slide: Animation()
Animation fall: Animation()
Animation wall_slide: Animation()
Animation punch1: Animation()
Animation punch2: Animation()
Animation punch3: Animation()
Animation punch_barrage: Animation();
Animation kick1: Animation()
Animation kick2: Animation()
Animation double_kick: Animation()
Animation unsheathe: Animation()
Animation sheathe: Animation()
Animation idle_sword: Animation()
Animation running_sword: Animation()
Animation slash1: Animation()
Animation triple_slash: Animation()

PlayerData: playerData

int: cont
int: timer_second_punch

bool: sword
bool: unsheathing
bool: sheathing
bool: is_punching
bool: is_kicking
bool: is_slashing
bool: is_jumping
bool: is_falling
bool: can_jump
bool: on_the_floor
SDL_RendererFlip: flip
bool: ability_boost
bool: looking_right
bool: looking_left
bool: god_mode
bool: offset_x_added
bool: offset_y_added
p2Point<int>: spawn
SDL_Texture*: texture

ModulePlayer()
virtual ~ModulePlayer()

bool: Start()
bool: Update(float dt)
bool: CleanUp()

int: GetPlayerTile(fPoint pos) const

COLLISION_TYPE: CheckCollision(int x) const

bool: Load(pugi::xml_node& data)
bool: Save(pugi::xml_node& data) const
void: LoadTexture()
void: LoadAnimation(pugi::xml_node animation_node, Animation* animation)
void: FindPlayerSpawn()
void: SpawnPLayer()

Skeleton

Animation idle: Animation()

Animation notice: Animation()

Animation hit: Animation()

Animation walk: Animation()

Animation death: Animation()

p2Point<int>: spawn

EntityData: skeletonData

Skeleton(int x, int y, ENTITY_TYPE type)
~Skeleton()

bool: Start()
bool: Update(float dt)
bool: CleanUp()

bool: Load(pugi::xml_node& data)

bool : Save(pugi::xml_node& data) const

void: LoadTexture()

void: LoadAnimation(pugi::xml_node animation_node, Animation* animation)

BlackBandit

Animation idle: Animation()

Animation idleSwordUp: Animation()

Animation running: Animation()

Animation attack: Animation()

Animation death: Animation()

Animation falling : Animation()

p2Point<int>: spawn

EntityData:banditData;

BlackBandit(int x, int y, ENTITY_TYPE type)

~BlackBandit()

bool: Start()

bool: Update(float dt)

bool: CleanUp()

bool: Load(pugi::xml_node& data)

bool: Save(pugi::xml_node& data) const

void: LoadTexture()

void: LoadAnimation(pugi::xml_node animation_node, Animation* animation)

BigBat

Animation fly: Animation()

p2Point<int>: spawn

EntityData: batData

BigBat(int x, int y, ENTITY_TYPE type);

~BigBat();

bool: Start()
bool: Update(float dt)
bool: CleanUp()

bool: Load(pugi::xml_node& data)

bool: Save(pugi::xml_node& data) const

void: LoadTexture()

void: LoadAnimation(pugi::xml_node animation_node, Animation* animation);