transform *

actions *

IAComponent
- type : int
- lastShot : int
+ IAComponent()
+ IAComponent(target : Entity *, type: int)
+ init()
+ update()
+ draw()
+ simpleFollow()
+ simpleFollowFar()
+ simpleShoot()

target *

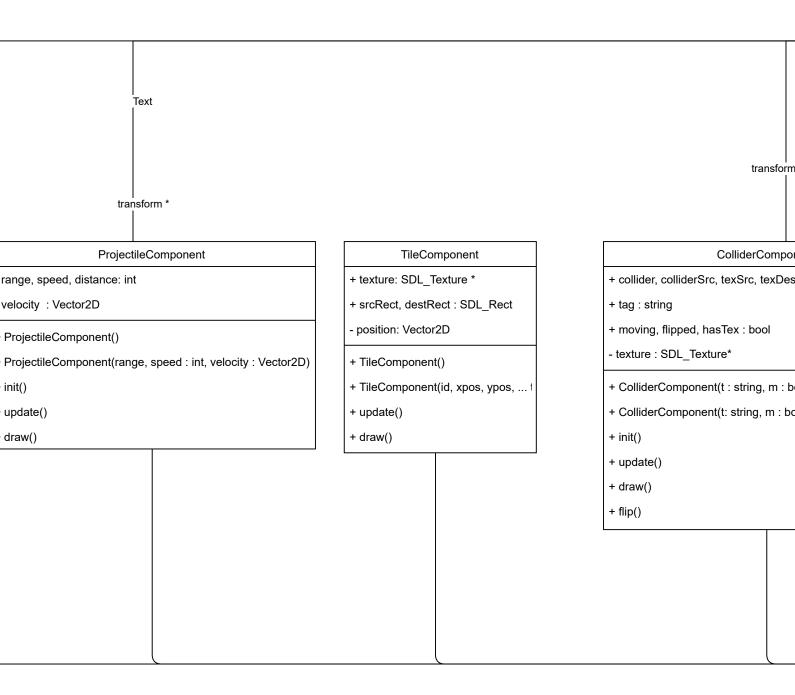| UILabel |
|---|
| + labelTexture : SDL_Texture * |
| - position: SDL_Rect |
| - textColor : SDL_Color |
| - labelText, labelFont : string |
| + UILabel(position : Vector2D, text, font : string, color : SDL_col |
| + setLabelText(text, font : string) |
| + draw() |

Text

transform *

transform

ProjectileComponent

- range, speed, distance: int

- velocity : Vector2D

+ ProjectileComponent()

+ ProjectileComponent(range, speed : int, velocity : Vector2D)

+ init()

+ update()

+ draw()

TileComponent

+ texture: SDL_Texture *

+ srcRect, destRect : SDL_Rect

- position: Vector2D

+ TileComponent()

+ TileComponent(id, xpos, ypos, ... t

+ update()

+ draw()

ColliderCompon

+ collider, colliderSrc, texSrc, texDes

+ tag : string

+ moving, flipped, hasTex : bool

- texture : SDL_Texture*

+ ColliderComponent(t : string, m : bo

+ ColliderComponent(t: string, m : bc

+ init()

+ update()

+ draw()

+ flip()

TransformComponent

+ position, velocity, previousPos : Vector2D

+ rotation : double

+ rotationCenter : SDL_Point

+ speed, height, width, scale : int

+ TransformComponent()

+ TransformComponent(sc : int)

+ TransformComponent(x,y : int)

+ TransformComponent(x, y, w, h, sc : int)

+ init()

+ update()

transform *

WeaponComponent

- colliderRect : SDL_Rect*

- tag : string

- ownerPos, ownerHand : Vector2D*

- attacking: bool

- ownerFacing : char

- attackStart: int

+ WeaponComponent(t : string, m : bool)

+ init()

+ update()

+ draw()

+ attack()

owner *

targets : vector

...ment

...t : SDL_Rect

...ool)

...ool, rect : SDL_Rect)

collider *

- previous...

- gravity, j...

- jumpSta...

+ Actions(...

+ init()

+ update(...

+ draw()

+ walk(dir...

+ jumpPro...

+ jumpSta...

+ shootPr...

+ setWea...

InputController

- eventHandler: EventHandler

+ InputController()

+ init()

+ update()

<struct> Animation

+ Animation()

+ Animation(i, f, s : int)

+ index, frames, speed : int

transform *

actions *

animations : map<string, Animat

ActionsComponent

Pos: Vector2D

umping, falling: bool

rtPos: int

Component()

)

ection : int)

ocess()

art()

ojectile([...])

pon(weapon : WeaponComponent*)

transform *

sprite *

actions *

SpriteComponent

- texture : SDL_Texture*

- src, dest : SDL_Rect

- animated : bool

- frames, speed : int

+ animIndex : int

+ SpriteComponent()

+ SpriteComponent(idTex : string)

+ SpriteComponent(idTex : string, isAnimated

+ setTex(idTex : int)

+ init()

+ update()

+ draw()

+ play(animName : string)

ion>

Component

+ *init()*

+ *update()*

+ *draw()*

entity *

components : vector

vector

Entity

- active : bool

+ Entity(mManager : Manager*)

+ update()

draw()

destroy()

+ hasGroum(mGroup : groupLabels) : bool

+ delGroup(mGroup : groupLabels)

+ <T>addComponent(...): Type T

+ <T>getComponent(...) : Type T

groupedEntities

Manager

+ update()

+ draw()

+ refresh()

+ addToGroup(mEntity : Entity*, Group mGroup)

+ getGroup(Group mGroup) : vector<Entity*>

+ addEntity() : Entity*

entities: vector

manager*

manager &

AssetManager

- textures: map<string, SDL_Texture*>

- animations: map<string, map<string,Animation>>

- fonts: map<string, TTF_Font*>

createProjectile() : void

createWeapon() : void

createEnemies() : void

TextureManager

LoadTexture() : SDL_Texture*

Draw()

DrawRotate()

Collison

AABB(...) : bool

AABB(...) : bool

resolveCollisions(players : Entity, colliders: vector<Entity *>)

<ENUM> groupLabels

groupMap,
groupPlayers,
groupEnemies,
groupTerrainColliders,
groupProjectiles,
groupWeapon

Game

- windows : SDL_Window*

+ renderer : SDL_Renderer*

+ event: SDL_Event*

+ isRunning: bool

+ camera: SDL_Rect

+ windowSize: Vector2D

+ currentMapSize: Vector2D

+ gravityStrength: int

+ FPS: int

Game()

init(title : const char*, xpos,ypos,width,height : int, fullscreen

update()

render()

clean()

clean()

clean()

running() : bool

terrainColliders

enemies

tiles

players

projectiles

weapons

playerPointer *

assets *

area1 *

Map
- area, areaPath : string
- mapSize : Vector2D
- texPerLine, mapScale, tilescale, scaledSize: int
- hasColliders: bool
+ Map(area : string)
+ LoadMap()
+ addTile(idSrc, x, y : int)
+ addTile(idSrc, x, y : int, collider : SDL_Rect)

: bool