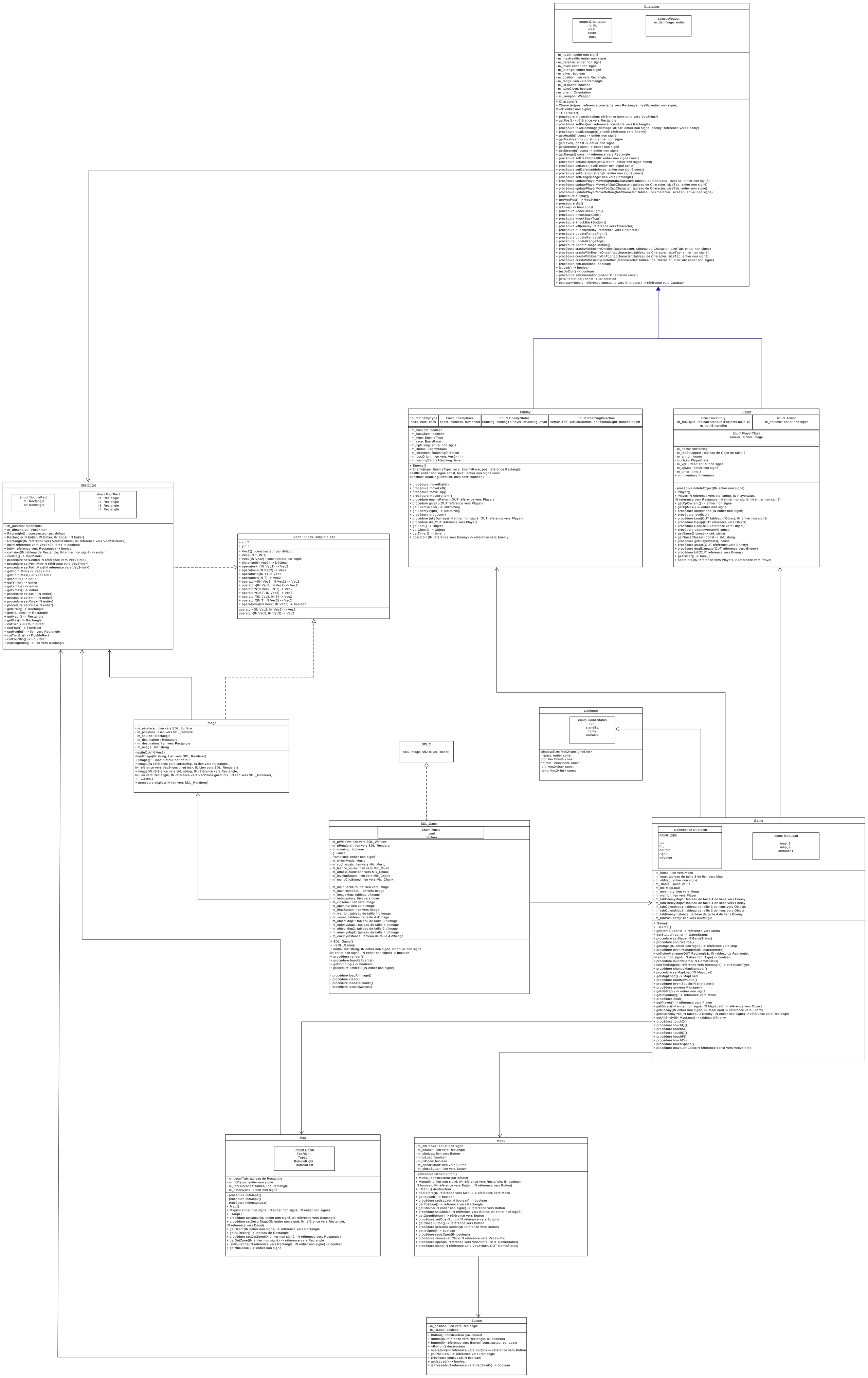
Character
enum Orientation
north,
west,
south,
east
struct Weapon
m_dammage: entier
- m_heath: entier non signé
- m_maxHealth: entier non signé
- m_defense: entier non signé
- m_level: entier non signé
- m_strengh: entier non signé
- m_alive : boolean
- m_position: lien vers Rectangle
- m_range: lien vers Rectangle
- m_isLoaded: boolean
- m_isXpGiven: boolean
- m_orient: Orientation
+ m_weapon: Weapon
+ Character()
+ Character(pos: référence constante vers Rectangle, health; entier non signé,
level: entier non signé)
+ ~Character()
+ procédure move(direction: référence constante vers Vec2<int>)
+ getPos() -> référence vers Rectangle
+ procédure setPos(rec: référence constante vers Rectangle)
+ procédure takeDammage(damageToDeal: entier non signé, enemy: référence vers Enemy)
+ procédure dealDamage(c_enemi: référence vers Enemy)
+ getHealth() const -> entier non signé
+ getMaxHealth() const -> entier non signé
+ getLevel() const -> entier non signé
+ getDefense() const -> entier non signé
+ getStrengh() const -> entier non signé
+ getRange() const -> référence vers Rectangle
+ procédure setHealth(health: entier non signé const)
+ procédure setMaxHealth(maxHealth: entier non signé const)
+ procédure setLevel(level: entier non signé const)
+ procédure setDefense(defense: entier non signé const)
+ procédure setStrengh(strengh: entier non signé const)
+ procédure setRange(range: lien vers Rectangle)
+ procédure updatePlayerMoveRight(tabCharacter: tableau de Character, sizeTab: entier non signé)
+ procédure updatePlayerMoveLeft(tabCharacter: tableau de Character, sizeTab: entier non signé)
+ procédure updatePlayerMoveTop(tabCharacter: tableau de Character, sizeTab: entier non signé)
+ procédure updatePlayerMoveBottom(tabCharacter: tableau de Character, sizeTab: entier non signé)
+ procédure display()
+ getVecPos() -> Vec2<int>
+ procédure die()
+ isAlive() -> bool const
+ procédure knockBackRight()
+ procédure knockBackLeft()
+ procédure knockBackTop()
+ procédure knockBackBottom()
+ procédure kill(enemy: référence vers Character)
+ procédure attack(enemy: référence vers Character)
+ procédure updateRangeRight()
+ procédure updateRangeLeft()
+ procédure updateRangeTop()
+ procédure updateRangeBottom()
+ procédure crashWithEnemyOnRight(tabcharacter: tableau de Character, sizeTab: entier non signé)
+ procédure crashWithEnemyOnLeft(tabcharacter: tableau de Character, sizeTab: entier non signé)
+ procédure crashWithEnemyOnTop(tabcharacter: tableau de Character, sizeTab: entier non signé)
+ procédure crashWithEnemyOnBottom(tabcharacter: tableau de Character, sizeTab: entier non signé)
+ procédure setLoad(load: boolean)
+ isLoad() -> boolean
+ testInOut() -> boolean
+ procédure setOrientation(orient: Orientation const)
+ getOrientation() const -> Orientation
+ operator=(copie: référence constante vers Character) -> référence vers Caracter
Rectangle
struct DoubleRect
r1: Rectangle
r2: Rectangle
struct FourRect
r1: Rectangle
r2: Rectangle
r3: Rectangle
r4: Rectangle
+ m_position: Vec2<int>
+ m_dimension: Vec2<int>
+ Rectangle() : constructeur par défaut
+ Rectangle(IN Entier, IN Entier, IN Entier, IN Entier)
+ Rectangle(IN référence vers Vec2<Entier>, IN référence vers Vec2<Entier>)
+ in(IN référence vers Vec2<Entier>) -> boolean
+ in(IN référence vers Rectangle) -> boolean
+ collision(IN tableau de Rectangle, IN entier non signé) -> entier
+ centre() -> Vec2<int>
+ procédure setCentre(IN référence vers Vec2<int>)
+ procédure setPointMin(IN référence vers Vec2<int>)
+ procédure setPointMax(IN référence vers Vec2<int>)
+ getPointMin() -> Vec2<int>
+ getPointMax() -> Vec2<int>
+ getXmin() -> entier
+ getYmin() -> entier
+ getXmax() -> entier
+ getYmax() -> entier
+ procédure setXmin(IN entier)
+ procédure setYmin(IN entier)
+ procédure setXmax(IN entier)
+ procédure setYmax(IN entier)
+ getDroit() -> Rectangle
+ getGauche() -> Rectangle
+ getHaut() -> Rectangle
+ getBas() -> Rectangle
+ cutTwo() -> DoubleRect
+ cutFour() -> FourRect
+ cutHeight() -> lien vers Rectangle
+ cutTwoBis() -> DoubleRect
+ cutFourBis() -> FourRect
+ cutHeightBis() -> lien vers Rectangle
Vec2 : Class Template <T>
+ x : T
+ y : T
+ Vec2() : constructeur par défaut
+ Vec2(IN T, IN T)
+ Vec2(IN Vec2) : constructeur par copie
+ distance(IN Vec2) -> Décimal
+ operator+=(IN Vec2) -> Vec2
+ operator-=(IN Vec2) -> Vec2
+ operator*=(IN T) -> Vec2
+ operator/=(IN T) -> Vec2
+ operator+(IN Vec2, IN Vec2) -> Vec2
+ operator-(IN Vec2, IN Vec2) -> Vec2
+ operator*(IN Vec2, IN T) -> Vec2
+ operator*(IN T, IN Vec2) -> Vec2
+ operator/(IN Vec2, IN T) -> Vec2
+ operator/(IN T, IN Vec2) -> Vec2
+ operator==(IN Vec2, IN Vec2) -> boolean
operator+(IN Vec2, IN Vec2) -> Vec2
operator-(IN Vec2, IN Vec2) -> Vec2
Enemy
Enum EnemyType
sbire, elite, boss
Enum EnemyRace
beast, element, humanoid
Enum EnemyStatus
roaming, comingToPlayer, attacking, dead
Enum RoamingDirection
verticalTop, verticalBottom, horizontalRight, horizontalLeft
- m_hasLoot: booléen
- m_hasChest: booléen
- m_type: EnemyTYpe
- m_race: EnelyRace
- m_xpGiving: entier non signé
- m_status: EnemyStaus
- m_direction: RoamingDirection
- m_posOrigin: lien vers Vec2<int>
- m_waitingBeforeAttacking: time_t
+ Enemy()
+ Enemy(type: EnemyType, race: EnemyRace, pos: reference Rectangle,
health: entier non signé const, level: entier non signé const,
direction: RoamingDirection, hasLooot: booleen)
+ procédure moveRight()
+ procédure moveLeft()
+ procédure moveTop()
+ procédure moveBottom()
+ procédure enemyPattern(OUT référence vers Player)
+ procédure giveXp(OUT référence vers Player)
+ getEnemyRace() -> std::string
+ getEnemyType() -> std::string
+ procédure dropLoot()
+ procédure takeDamage(IN entier non signé, OUT reference vers Player)
+ procédure die(OUT reference vers Player)
+ getLoot() -> Object
+ getChest() -> Object
+ getTimer() -> time_t
+ operator=(IN reference vers Enemy) -> reference vers Enemy
Player
struct Inventory
m_tabEquip: tableau statique d'objects taille 16
m_numEmptySlot
struct Armor
m_defense: entier non signé
Enum PlayerClass
warrior, archer, mage
- m_name: std::string
- m_tabEquipped : tableau de Objet de taille 2
- m_armor: Armor
- m_class: PlayerClass
- m_xpCurrent: entier non signé
- m_xpMax: entier non signé
- m_timer: time_t
+ m_inventory: Inventory
- procédure deleteObject(IN entier non signé)
+ Player()
+ Player(IN reference vers std::string, IN PlayerClass,
IN reference vers Rectangle, IN entier non signé, IN entier non signé)
+ getXpCurrent() -> entier non signé
+ getXpMax() -> entier non signé
+ procédure increaseXp(IN entier non signé)
+ procédure levelUp()
+ procédure Loot(OUT tableau d'Object, IN entier non signé)
+ procédure Equip(OUT référence vers Object)
+ procédure Use(OUT: référence vers Object)
+ procédure openInventory() const;
+ getName() const -> std::string
+ getNameClasse() const -> std::string
+ procédure getPlayerStats() const
+ procédure attack(OUT référence vers Enemy)
+ procédure dealDamage(OUT référence vers Enemy)
+ procédure kill(OUT référence vers Enemy)
+ getTimer() -> time_t
+ operator=(IN référence vers Player) -> reference vers Player
Common
enum GameStatus
run,
standBy,
home,
noValue
windowSize: Vec2<unsigned int>
impact: entier const
top: Vec2<int> const
bottom: Vec2<int> const
left: Vec2<int> const
right: Vec2<int> const
Image
- m_psurface : Lien vers SDL_Surface
- m_pTexture : Lien vers SDL_Texture
- m_source : Rectangle
- m_destination : Rectangle
- m_destination: lien vers Rectangle
- m_image: std::string
-testInOut(IN Vec2)
-loadImage(IN string, Lien vers SDL_Renderer)
+ Image() : Constructeur par défaut
+ Image(IN référence vers std::string, IN lien vers Rectangle,
IN référence vers Vec2<unsigned int>, IN Lien vers SDL_Renderer)
+ Image(IN référence vers std::string, IN référence vers Rectangle,
IN lien vers Rectangle, IN référence vers Vec2<unsigned int>, IN lien vers SDL_Renderer)
+ ~Game()
+ procédure display(IN lien vers SDL_Renderer)
SDL 2
sdl2-image, sfl2-mixer, sfl2-ttf
SDL_Game
Enum Music
cool
techno
- m_pWindow: lien vers SDL_Window
- m_pRenderer: lien vers SDL_Renderer
- m_running : boolean
- g: Game
- framelimit: entier non signé
- m_whichMusic: Music
- m_cool_music: lien vers Mix_Music
- m_techno_music: lien vers Mix_Music
- m_attackSound: lien vers Mix_Chunk
- m_levelupSound: lien vers Mix_Chunk
- m_menuClicSound: lien vers Mix_Chunk
- m_mainBackGround: lien vers Image
- m_mainHomeBtn: lien vers Image
- m_imageMap: tableau d'Image
- m_imInventory: lien vers Imae
- m_closeInv: lien vers Image
- m_openInv: lien vers Image
- m_healButton: lien vers Image
- m_warrior: tableau de taille 4 d'Image
- m_sword; tableau de taille 4 d'Image
- m_objectMap1: tableau de taille 3 d'Image
- m_enemyMap1: tableau de taille 4 d'Image
- m_objectMap2: tableau de taille 2 d'Image
- m_enemyMap2: tableau de taille 4 d'Image
- m_enemyInstance: tableau de taille 4 d'Image
+ SDL_Game()
+ ~SDL_Game()
+ init(IN std::string, IN entier non signé, IN entier non signé,
IN entier non signé, IN entier non signé) -> boolean
+ procédure render()
+ procédure handleEvents()
+ getRunning() -> boolean
+ procédure limitFPS(IN entier non signé)
- procédure loadAllImage()
- procédure clean()
- procédure loadAllSounds()
- procédure loadAllMusics()
Game
Namespace Diretcion
enum Type
top,
lft,
bottom,
right,
noValue
enum MapLoad
map_1,
map_2,
instance1
- m_home: lien vers Menu
- m_map: tableau de taille 3 de lien vers Map
- m_nbMap: entier non signé
- m_status: GameStatus
- m_ml: MapLoad
- m_inventory: lien vers Menu
- m_warrior: lien vers Player
- m_tabEnemyMap1: tableau de taille 4 de liens vers Enemy
- m_tabEnemyMap2: tableau de taille 4 de liens vers Enemy
- m_tabObjectMap1: tableau de taille 3 de liens vers Objects
- m_tabObjectMap2: tableau de taille 2 de liens vers Object
- m_tabEnemyInstance: tableau de taille 4 de liens Enemy
- m_tabPosEnemy: lien vers Rectangle
+ Game()
+ ~Game()
+ getHome() const -> référence vers Menu
+ getStatus() const -> GameStatus
+ procédure setStaus(IN GameStatus)
+ procédure controlePos()
+ getMap1(IN entier non signé) -> référence vers Map
+ procédure eventManagers(IN characterére)
+ collisionManager(OUT Rectangle&, IN tableau de Rectangle,
IN entier non signé, IN direction::Type) -> boolean
+ procédure launchGame(IN GameStatus)
+ isAtTheEdge(IN référence vers Rectangle) -> direction::Type
+ procédure changeMapManager()
+ procédure setMapLoad(IN MapLoad)
+ getMapLoad() -> MapLoad
+ procédure loadNewZone()
+ procédure eventTouch(IN charactère)
+ procédure ennemyManager()
+ getNbMap() -> entier non signé
+ getInventory() -> référence vers Menu
+ procédure heal()
+ getPlayer() -> référence vers Player
+ getObject(IN entier non signé, IN MapLoad) -> référence vers Object
+ getEnemy(IN entier non signé, IN MapLoad) -> référence vers Enemy
+ getAllEnemyPos(IN tableau d'Enemy, IN entier non signé) -> référence vers Rectangle
+ getAllEnely(IN MapLoad) -> tableau d'Enemy
+ procédure touchZ()
+ procédure touchQ()
+ procédure touchS()
+ procédure touchD()
+ procédure touchF()
+ procédure touchC()
+ procédure touchSpace()
+ procédure moveLeftClick(IN référence const vers Vec2<int>)
Map
enum Decal
TopRight,
TopLeft,
BottomRight,
BottomLeft
- m_decorTab: tableau de Rectangle
- m_nbDecor: entier non signé
- m_tabOutZones: tableau de Rectangle
- m_nbOutZone: entier non signé
- procédure initMap1()
- procédure initMap2()
- procédure initInstance1()
+ Map()
+ Map(IN entier non signé, IN entier non signé, IN entier non signé)
+ ~Map()
+ procédure setDecor(IN entier non signé, IN référence vers Rectangle)
+ procédure setDecorDiago(IN entier non signé, IN référence vers Rectangle,
IN référence vers Decal)
+ getDecor(IN entier non signé) -> référence vers Rectangle
+ getAllDecor() -> tableau de Rectangle
+ procédure setOutZone(IN entier non signé, IN référence vers Rectangle)
+ getOutZone(IN entier non signé) -> référence vers Rectangle
+ isInOutZone(IN référence vers Rectangle, IN entier non signé) -> boolean
+ getNbDecor() -> entier non signé
Menu
- m_nbChoice: entier non signé
- m_positon: lien vers Rectangle
- m_choices: lien vers Button
- m_isLoad: boolean
- m_isOpen: boolean
- m_openButton: lien vers Button
- m_closeButton: lien vers Button
- procédure isLoadButton()
+ Menu() constructeur par défault
+ Menu(IN entier non signé, IN référence vers Rectangle, IN boolean,
IN boolean, IN référence vers Button, IN référence vers Button)
+ ~Menu() destructeur
+ operator=(IN référence vers Menu) -> référence vers Menu
+ getIsLoad() -> boolean
+ procédure setIsLoad(IN boolean) -> boolean
+ getPosition() -> référence vers Rectangle
+ getChoice(IN entier non signé) -> référence vers Button
+ procédure setChoice(IN référence vers Button, IN entier non signé)
+ getOpenButton() -> référence vers Button
+ procédure setOpenButton(IN référence vers Button)
+ getCloseButton() -> référence vers Button
+ procédure setCloseButton(IN référence vers Button)
+ getIsOpen() -> boolean
+ procédure setIsOpen(IN boolean)
+ procédure mouseLeftClick(IN référence vers Vec2<int>)
+ procédure open(IN référence vers Vec2<int>, OUT GameStatus)
+ procédure close(IN référence vers Vec2<int>, OUT GameStatus)
Button
- m_position: lien vers Rectangle
- m_isLoad: boolean
+ Button() constructeur par défault
+ Button(IN référence vers Rectangle, IN boolean)
+ Button(IN référence vers Button) constructeur par copie
+ ~Button() destructeur
+ operator=(IN référence vers Button) -> référence vers Button
+ getPosition() -> référence vers Rectangle
+ procédure setIsLoad(IN boolean)
+ getIsLoad() -> boolean
+ isPressed(IN référence vers Vec2<int>) -> boolean