

# A DANCE OF SUN AND MOON

MIALI MATTEO  
HENRIKSEN  
MARIUS

Inspiré de « A Dance of Fire and Ice », « A dance of sun and moon » est un jeu de rythme où le but est de finir un parcours de cases en suivant le rythme d'une musique. Le joueur dispose de 2 cercles (le soleil et la lune), le second orbitant le premier.

# CLASS MAP

	Map
	<ul style="list-style-type: none"><li>-map_music_ : lien vers Conductor</li><li>-map_queue_ : lien vers std deque de string</li></ul>
	<ul style="list-style-type: none"><li>+Map(IN level_folder_path : string, IN music_bpm : int, IN first_beat_offset : float )</li><li>+~Map()</li><li>+FileToMap(IN map_path : string)</li><li>+MapToFile(IN map_path : string) const</li><li>+Display() const</li><li>+GetNextTileToDisplay() const : string</li><li>+PopNewTile()</li><li>+IsEmpty() const : bool</li><li>+UpdateMusic()</li><li>+PlayMusic()</li><li>+PlaySoundEffect()</li><li>+GetMusicBpm() const : float</li><li>+GetMusicSecondsPerBeat() const : float</li><li>+PauseMusic(IN is_paused : bool)</li></ul>



# CLASS TILE

Tile	
<pre>-Types_ : Enum des types -tile_type_ : Types_ -direction_ : string  +Tile(IN file_path : string, IN x_pos : float, IN y_pos : float, IN width : int, IN height : int, IN tile_type : string, IN direction : string) = appel a Image() +Tile() +GetType(IN type : string) : Types_ +DetermineNextDirection(IN next_tile_type : string) +DetermineNextTileCoordinates(IN next_tile_type : string, OUT next_tile_x_pos : float, OUT next_tile_y_pos : float) const +IsVertical() const : bool +IsHorizontal() const : bool +IsCorner() const : bool +IsUpCorner() const : bool +IsDownCorner() const : bool +IsEnd() const : bool +IsSwirl() const : bool +IsSpeed() const : bool +IsSlow() const : bool +GetCenter(OUT x_pos : float, OUT y_pos : float) const +GetDirection() const : string</pre>	



# CLASS LEVEL

```
Level

- displaymode_ : lien vers SDL_DisplayMode
- map_ : lien vers Map
- player_ : lien vers Player
- displayed_tiles_ : lien vers deque de lien vers Tile
- level_position_ : lien vers deque de lien vers Tile
- tile_image_path_ : string
- image_extension_ : string
- scroll_x_diff_ : float
- scroll_y_diff_ : float
- scroll_speed_ : float
- score_ : int
- lives_ : int
- finished_ : bool
- has_moved_ : bool
- input_delay_ : high_resolution_clock

+Level(IN img_tile_path : string, IN img_extension : string, IN level_folder_path : string, IN displaymode : lien vers SDL_DisplayMode, IN song_bpm : float)
+~Level()
+Load(IN level_folder_path : string, IN song_bpm : float)
+LoadNewTiles()
+UnloadOutOfScreenTiles()
+Render(player : lien vers SDL_Renderer)
+Start()
+Pause(IN is_paused : bool)
+Update()
+AddTileToDisplay(tile_idx : lien vers Tile)
+UpdateTilePosition()
+IsNextTileRenderable(IN current_tile : Tile, IN next_tile_type : string)
+IsTileOutOfScreen(IN tile : Tile) const
+GetOldestDisplayedTile() const : lien vers Tile
+GetCurrentTile() const : lien vers Tile
+PopOldestDisplayedTile()
+MoveToNextTile()
+ApplyTileEffect()
+UpdateScrolling(current_tile : lien vers Tile, next_tile : lien vers Tile)
+IsPlayerOnNextTile() const : bool
+GetDistanceFromNextTile() const : float
+CheckPlayerAccuracy(OUT rythm_text : string, OUT tile_x : float, OUT tile_y : float)
+GetScore() const : int
+GetLife() const : int
+GetEnemyHealth() const : float
+PlaySoundEffect()
+GetTimeSecondsPerBeat() const : float
```

# CLASS GAME RENDERER

Game Renderer
-renderer_ : lien vers SDL_Renderer -displaymode_ : lien vers SDL_DisplayMode -window_ : lien vers SDL_Window -level_ : lien vers Level -menu_background_ : lien vers Image -quit_button_ : lien vers Button -img_tile_path_ : string -img_extension_ : string -level_background_ : lien vers Image -pause_image_ : lien vers Image -restart_button_ : lien vers Button -go_back_button_ : lien vers Button -play_button_ : lien vers Button -rhythm_ : lien vers Text -score_ : lien vers Text -life_image_ : lien vers Image -font_ : Lien vers SDL TTF Font
+GameRenderer() +-GameRenderer() +IfKeyDown() +IfMouseButtonDown( IN button_name : string, event : SDL_event ) : bool +InitMenu() +RenderMenu() +InitLevel() +UpdateLevel() +RenderLevel(IN is_paused : bool) +StartLevel() +ClearLevel() +PauseLevel(IN is_paused : bool) +DerenderMenu() +CheckIfLevelFinished( OUT finished : bool) +RestartLevel() +Countdown()



# CONCLUSION

- L'utilisation d'environnements de travail different (WSL / Mac), et les problèmes lies
- Le projet nous a montres les difficultes de maintenir une grande architecture de code (debugger un bug a la racine du code), et donc l'importance de constamment teste ses modules et chaque ajout.
- Intéressant d'analyser un jeu pour comprendre comment il a été codé dans le but de le recréer.
- Difficultés rencontrées au niveau des assets des tiles (distance entre les centres)
- Problèmes au niveau du rythme, de la sensation de jeu et du scrolling.

Possibles ajouts : création de map, mouvement de camera prédéfini, plus de type de tiles, load des images une fois.

