MŰEGYETEM 1782

FINAL PROJECT DOCUMENTATION

Project: “Nostalgia” 2D Game

Course: Basic of programming 3

Student: Mourad Gaaloul

Neptun: INZYHQ

Classes Description:

GamePanel class:

The GamePanel class represents the drawing canvas, it contains the game loop which keeps the game moving forward. It contains the main functions of the game : render and draw (Double buffering) which are responsible for drawing the graphics on the screen and update function which responsible for updating the game through time.

StateManager class:

The StateManager class contains the list of game states (menu, play, game over), it decides which state to update and draw and switches between the different states.

GameState:

Represents the abstract class for the other 3 game state classes, it contains the main game methods that are overridden by the other child classes.

MenuState class:

The MenuState class represents the main menu for the game, you can select between 4 choices: play, help, score and quit.

PlayState class:

The PlayState class represents the main playing class of the game, it contains all you need for the gameplay: the player, the map, the books, items. It draws all the objects of the game and updates them.

GameOverState class:

This class is responsible for the ending of the game, it just draws the ending gif image and draws the finishing text.

Score class:

The score class is a serializable class, it stores a score every time you finish the game, the score is the time spent in a round. A score object is loaded every time you run the program so you can check the previous scores.

Content class:

This class is responsible for loading the image tiles and splitting them into parts if needed (the player tileset for example) , also It contains the file names of the audio files.

Object class:

Represents the abstract class for all game objects (player, book, item), it contains the logic to move around the game map.

Player class:

Child class of the Object super class and the only one that fully utilizes it because the player is the only object that moves in the game map. It contains the gameplay logic for associated with the player.

Book class:

This class represents the book object that should be collected by the player. When the player collects all the books, he wins

Item class:

The item class represents two types of items: key and axe, every item is responsible for opening a specific type of door. When the item is collected, the player gets informed.

Animation class:

This class takes a series of tile images and animate it in the appropriate order, it is used to animate the player tiles.

Game_Map class:

This class takes the tile set that contains the different elements of the map and the map text file which contains a matrix of different indexes, each index corresponds to a specific tile in the tile set. We only draw the part of the map that is shown on the game panel and it gets updated every time the player moves.

PS : if you wonder whether I wrote all the different 1600 (40*40) entries in the map text file manually. Yes I did and it was a stupid idea LOL.

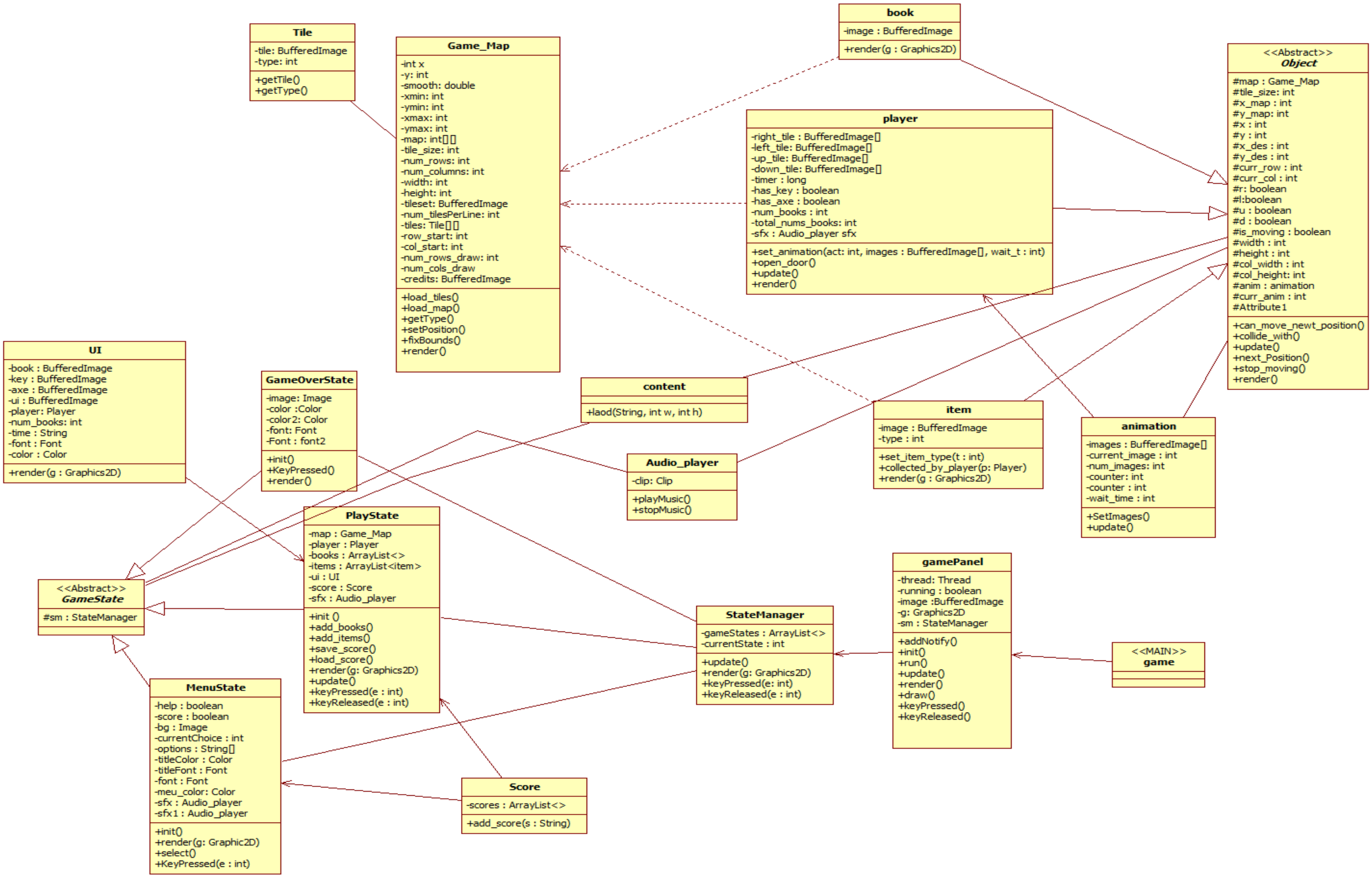
Audio_player class:

Responsible for reading ,playing and stopping the audio files

UI class:

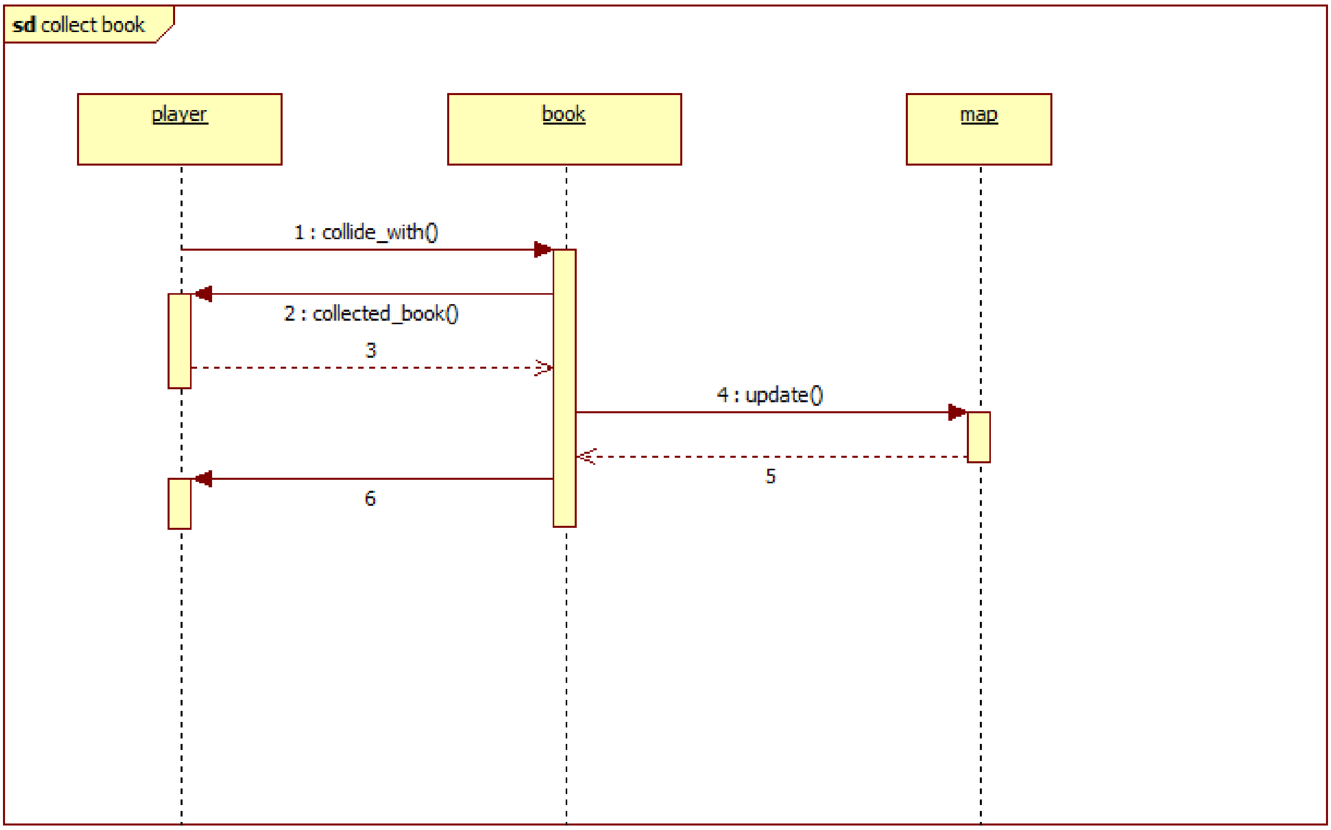
Represents the user interface of the game, it draws the book counter, The HUD and the game timer.

Class Diagram

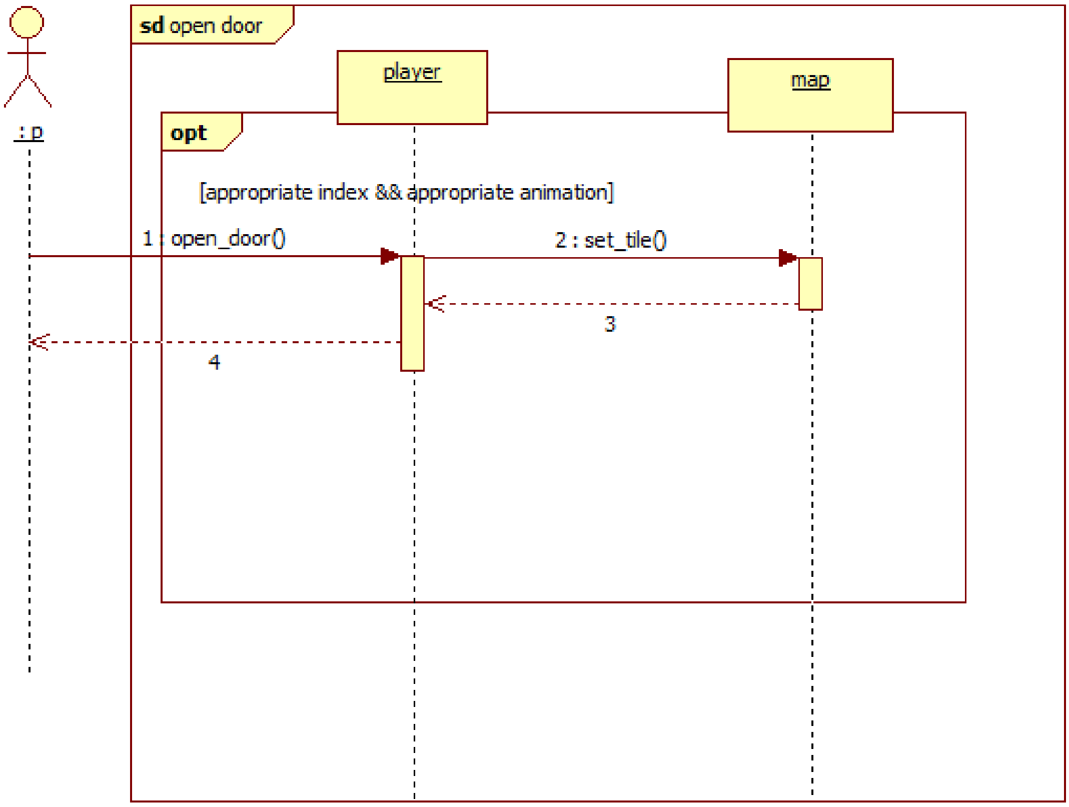


Sequence Diagrams

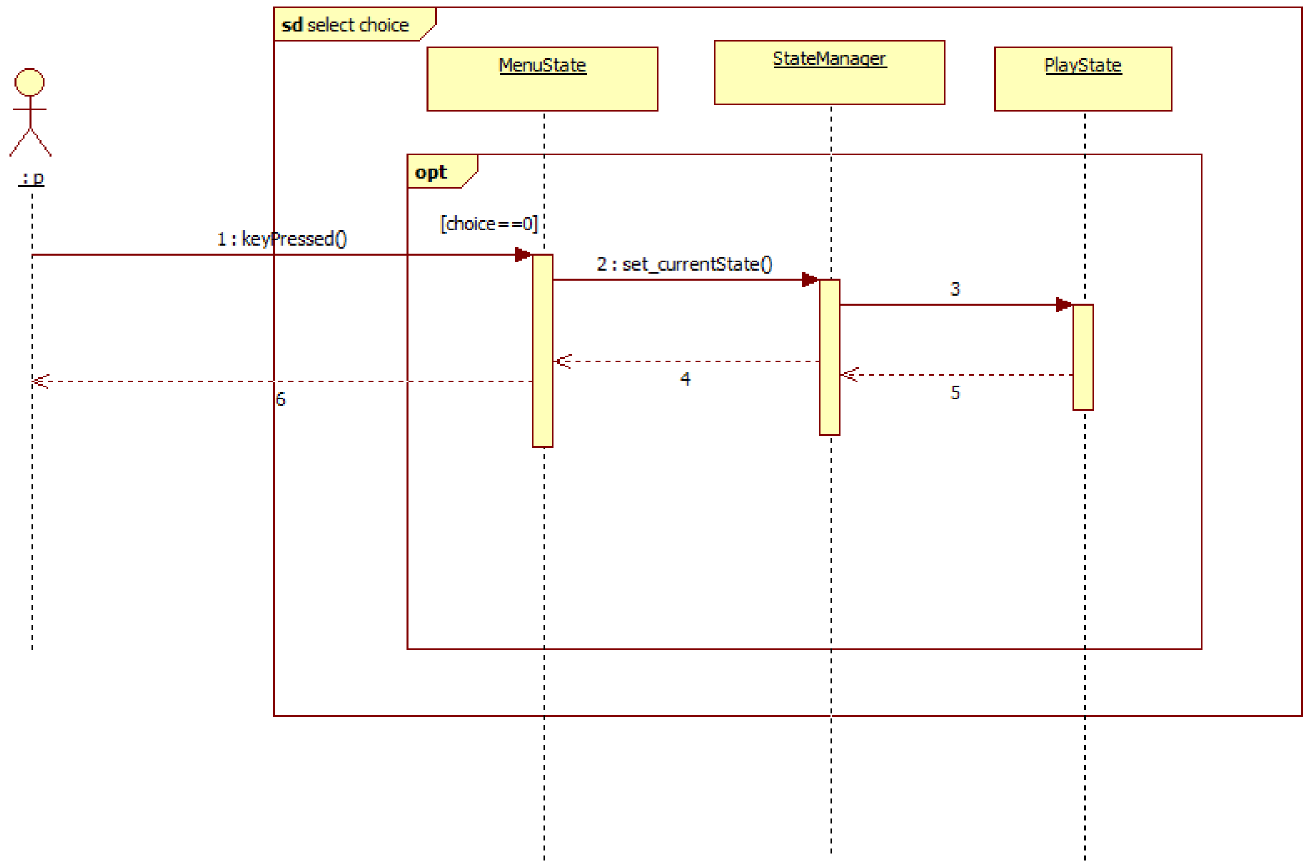
Player collects book:



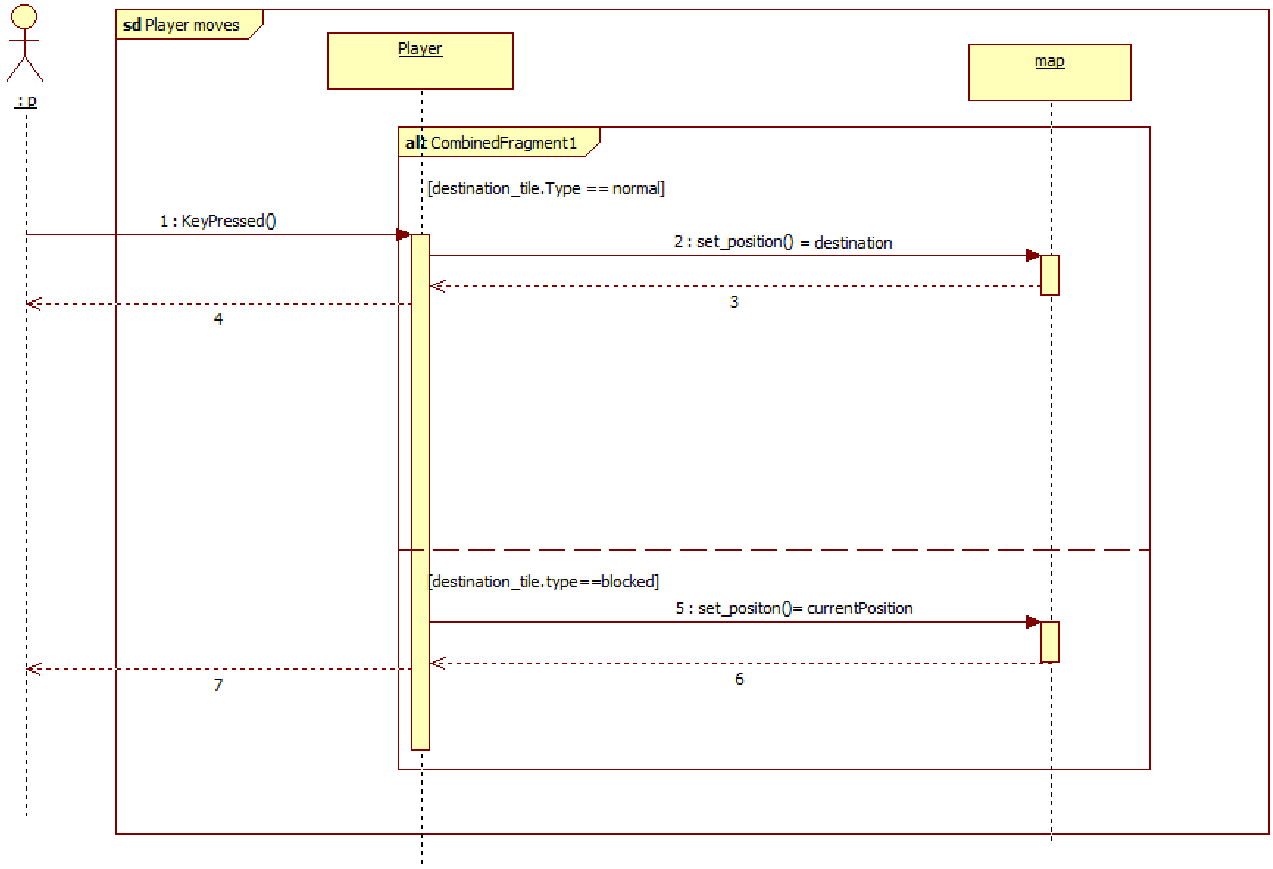
Player opens door:



Select Play in Menu:



Player moves:


sd Player moves
: p
Player
map
alt CombinedFragment1
[destination_tile.Type == normal]
1 : KeyPressed()
2 : set_position() = destination
3
4
[destination_tile.type==blocked]
5 : set_positon()= currentPosition
6
7

Description of unit tests:

Game map class test:

This test checks if the tiles are loaded correctly by reading different values from the map_tile text file and check its type if it is blocked or normal tile.

Content class test:

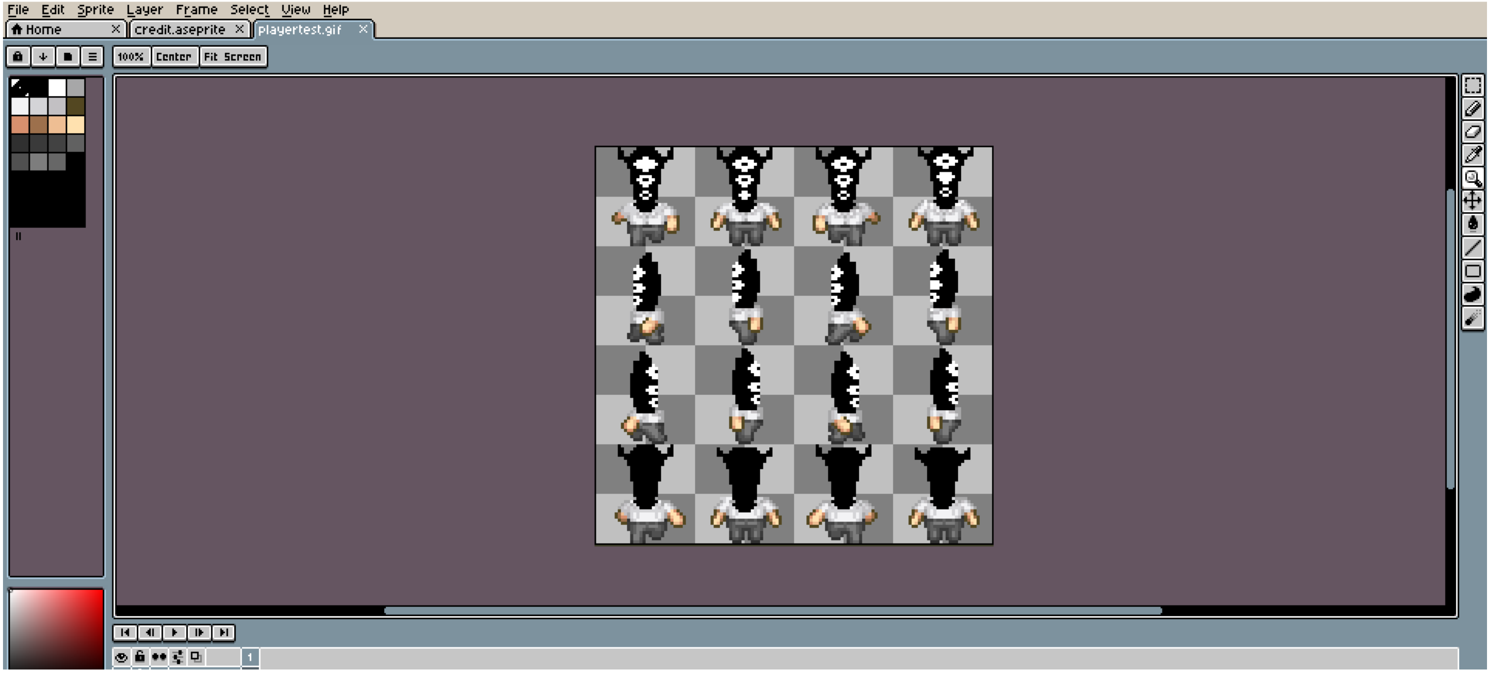
This test checks if the images are loaded and divided correctly into the Image matrix with the appropriate dimensions.

Player class test:

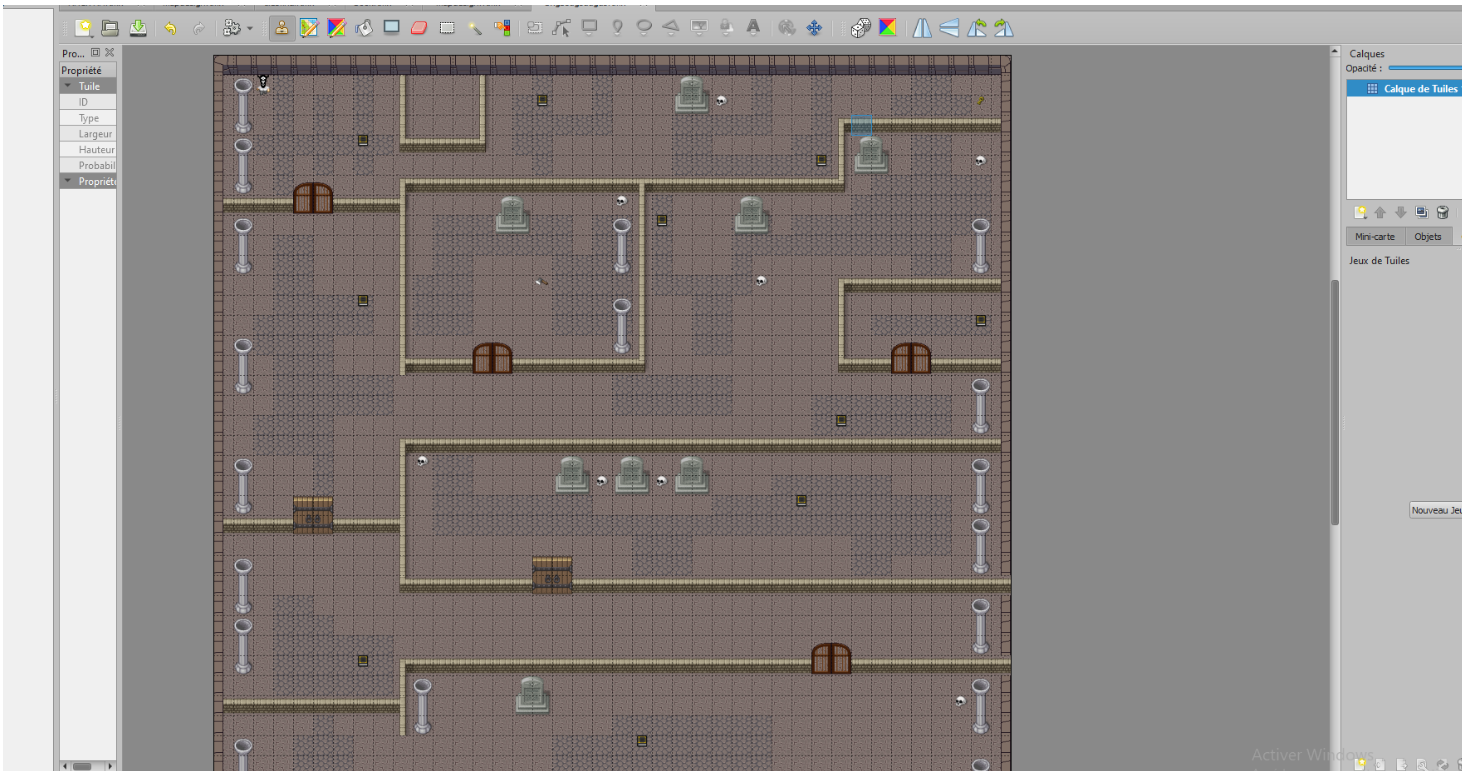
This test checks if the player character collides with a book or item.

Other tools used in the project:

Aseprite software : (for drawing the character)



Tiled software: (for drawing the map)



Note: the ending images in the gameover state are some of my old artworks done after discovering Budapest for the first time.

External resources:

- Basics of 2D game developments youtube tutorials.
- Stackoverflow/other forums

Mourad Gaaloul
INZYHQ