Topic description and sketch of UI

Dawid Bogocz

Project Topic:

Mario like platformer

Topic description:

I chose to create a platformer game in style of Super Mario Bros. The goal of this game is to get to the other side of the map without dying collecting as much gold coins as possible. It's going to have simple controls that let the player go left and right, up and down. There will be few kinds of enemies:

1. Ones that just walk from one side to another and kill the player if they run into him. The player can kill them by jumping on their head.
2. Ones that the player can't kill them.
3. Ones that do not move but they shot projectiles at the player.

Another trap is a pit willed with water that is going to kill the player if he falls into it.

User interface:

Main menu will contain:

1. Start game button
2. Settings
3. Game description
4. Exit

After starting the game player will be allowed to change his setting by using Tabulator on his keyboard. Points will be counted for every coin collected and level passed and stored into logs file that can be read after finishing the game. There could be ester eggs hidden throughout the game.

Classes that my program will contain are as stated bellow:

World Class – A class that renders and updates the game window and all the components

Player Class – A class that will be responsible for player movement and render
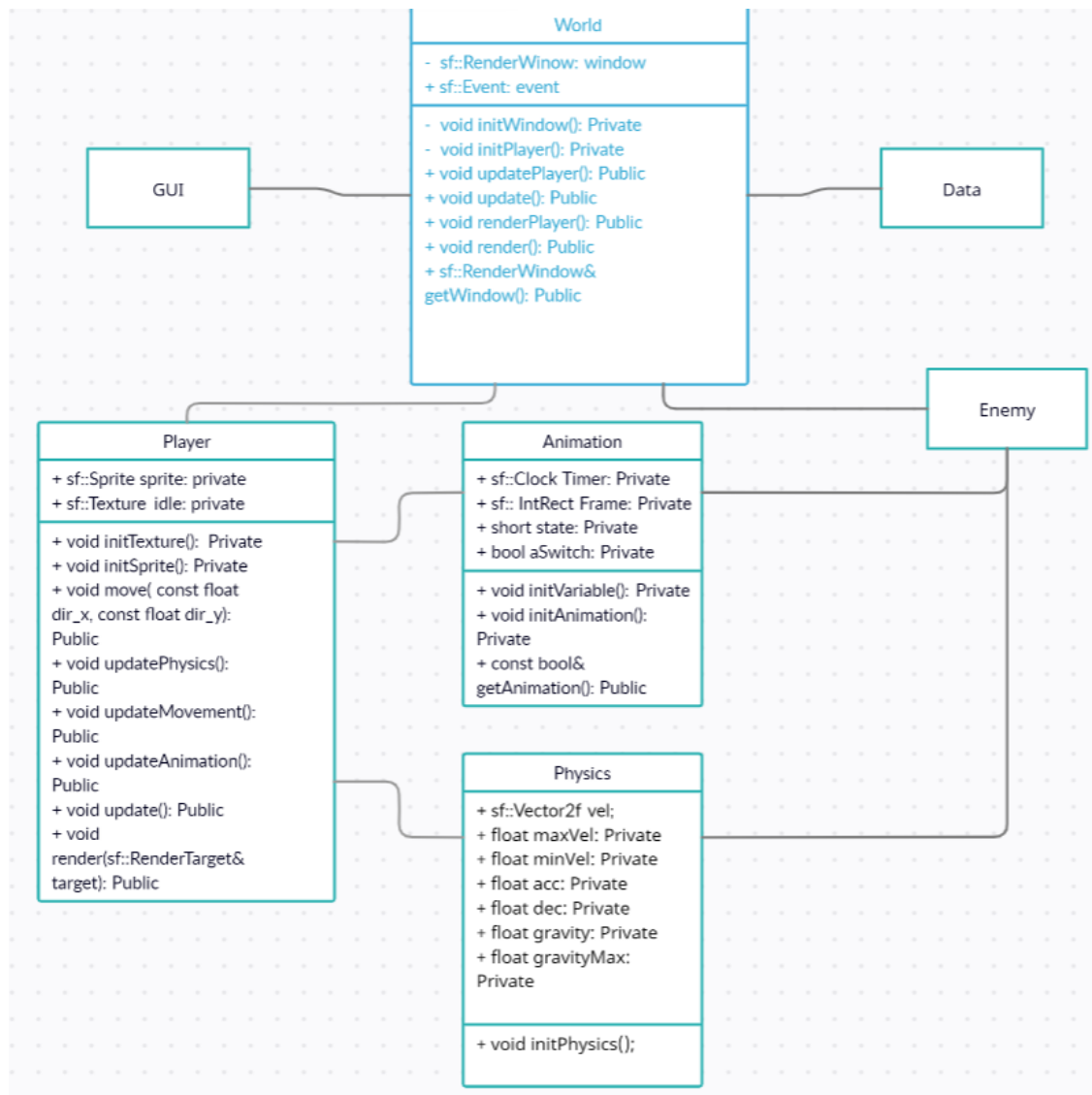
Animation Class – A class that will be responsible for animations present in the game

Physics Class – A class that will be responsible for player and enemy physics

Enemy Class – A class that will be responsible for enemies in the game

GUI Class – A class that will be responsible for player interface

Data Class – A class that will be responsible for player date such as hit score etc.

World

- sf::RenderWinow: window
+ sf::Event: event

- void initWindow(): Private
- void initPlayer(): Private
+ void updatePlayer(): Public
+ void update(): Public
+ void renderPlayer(): Public
+ void render(): Public
+ sf::RenderWindow&
getWindow(): Public

GUI

Data

Enemy

Player

+ sf::Sprite sprite: private
+ sf::Texture idle: private

+ void initTexture(): Private
+ void initSprite(): Private
+ void move(const float
dir_x, const float dir_y):
Public
+ void updatePhysics():
Public
+ void updateMovement():
Public
+ void updateAnimation():
Public
+ void update(): Public
+ void
render(sf::RenderTarget&
target): Public

Animation

+ sf::Clock Timer: Private
+ sf:: IntRect Frame: Private
+ short state: Private
+ bool aSwitch: Private

+ void initVariable(): Private
+ void initAnimation():
Private
+ const bool&
getAnimation(): Public

Physics

+ sf::Vector2f vel;
+ float maxVel: Private
+ float minVel: Private
+ float acc: Private
+ float dec: Private
+ float gravity: Private
+ float gravityMax:
Private

+ void initPhysics();

The general scheme of the program operation:

After running the game User has to choose from menu on the screen with following items: Start a game, Score and Quit.
After starting the game player will have to set his name and will be met with a text which informs that the first level has been started. Player starts with 3 lives, which can be lost in case of collision with an enemy or enemy bullet. After eliminating an enemy or collecting coins players score will increase. After completing the last stage, the game is finished. Players score is saved and he can see it by selecting Score in Main screen.

Techniques from thematic classes and external libraries:

My Project fill make use of:

- Modules
- File systems
- Ranges
- Threads
- SFML Library for graphical implementation