

SFML & Box2D classes

These are classes which would fit to the scope of the project and could be used in our own implementations. This is only an initial list. Thus, the classes used in the final implementation may vary from classes listed here.

SFML

- Background music
 - [sf::Music](#)
 - Not a high priority and should be easy to implement
- Main drawing window
 - [sf::RenderWindow](#)
 - Collects graphical objects (sprites) and displays those
- Off-screen graphics rendering
 - [sf::RenderTexture](#)
 - Could be implemented to reduce on-screen rendering
- Textures
 - [sf::Texture](#)
 - Use these and entity objects to create sprites
- Sprites
 - [sf::Sprite](#)
 - Can be used in drawing
- Clock
 - [sf::Clock](#)
 - Can be used in the points system of the game
- Input streams
 - [sf::InputStream](#)
 - Possibly used with level class

UI

Note: Buttons don't have their own SFML class so they need to be implemented and then used in UI window.

- [sf::Event](#)
 - `sf::Event::KeyEvent::code`

- Get which key was pressed
 - Used with [sf::Keyboard::Key](#)
 - Player movement and menu movement
- Main Menu: [sf::MouseButton](#),
 - Use with menu buttons
- Window [sf::Window](#)
 - Main graphical window for the app
- Text objects
 - [sf::Text](#)
 - These need to be at least used with menu buttons

Box2D

[Box2D Manual](#)

- b2World
 - Our own World class implementation uses Box2D World
 - Needs a gravity vector: gravity(0.0f, -10.0f)
 - Horizontal and vertical gravity
- Bodies
 - b2Body
 - Created and destroyed via the World
 - Needs a groundbox fixture
 - Either static or dynamic
 - All entity class object should use dynamic bodies and likewise barrier class objects use static bodies
- Simulating the World
 - world.Step
 - This is periodically called from GameEngine