# Game Engine Development - Assignment 1

The game that is going to be developed using this game engine, Will be based around a 'Brick breaking' type genre of game, Where the user will use keyboard inputs to control a character platform to bounce a ball back and forth to destroy target objects. There may be an aspect of AI as a final moving block that will fire back or dodge the user's attacks.

In my opinion the game engine will use SFML as the external library and API, SFML has the wider selection of libraries and functionality compared to SDL, while also being automatically hardware accelerated to run processes faster rather than SDL libraries. This is because of the use and access of the GPU rather than SDL's sole use of the CPU. Taking them factors aside, SFML is a more recent API so in turn the updates and bugs are being pushed out much faster than SDL libraries. There is good documentation and online tutorials for SFML whereas due to the lack of activity from SDL development team soon parts of the libraries will be depreciated and not function correctly.

2D rendering will be easily managed by SFML. "sf::RenderWindow adds high-level functions to help you draw things easily."(Sfml-dev.org, 2017). This alongside the 'draw' functions allows a multitude of 2D rendering options, Ie. Adding sprites, Text, shapes and vertex's. (Milchev, 2015)

SFML uses an easy to access functionality for text, sprites shapes and vertexes with the ability to render with different properties, using easy to understand and simplified API language. This helps with keeping the amount of code to compile down and in turn keeping processing speeds high while also leaving it easy to read for future developers and editors increasing the longevity of the engine. (Sfml-dev.org, 2017)

Personal Notes

\*\*\*\*Re read through slides and check that I’m explaining the right things.

• User Input – Discuss the options available and see if its worth using the library or coding in my own.

• Collision Detection / Physics – talk about the options available and explain why its important that these are done well, to stop serious memory leaks.

• AI - I don’t plan to use much AI, But if I do, Talk about my use of quad trees and the procedural use of them to be used as a component rather than a hard coded thing.

• Resource management – SMFL provides good resource management and this is important to keep speeds and reliability up. Also include something about limiting error files and read outs and limiting what users to developers can actually use and see from the game engine.

• Scene management – Not sure on this yet.

## **References**

8/11/17

(Gregory, 2014)

Gregory, J. (2014) *Game engine architecture*. 2nd edn. Boca Raton, Florida: CRC Press.

(Moreira, 2013)

Moreira, A., Haller, J. & Hansson, H.V. (2013) *SFML Game Development,* 1st edn, Packt Publishing, Olton.

(Milchev, 2015)

Milchev, M. (2015). *SFML essentials*. Birmingham, UK: Packt Publishing, pp.24-28.

(Sfml-dev.org, 2017)

Sfml-dev.org. (2017). *Tutorials for SFML 2.0 (SFML / Learn / 2.4 Tutorials)*. [online] Available at: https://www.sfml-dev.org/tutorials/2.0/ [Accessed 8 Nov. 2017].

## **Bibliography**

<https://www.sfml-dev.org/faq.php#grl-whatis>

<https://www.reddit.com/r/gamedev/comments/44npzz/sdl20_vs_sfml2_in_2016/>

<https://www.reddit.com/r/gamedev/comments/42og4b/what_is_a_good_library_to_2d_games_in_c/>

<https://www.sfml-dev.org/tutorials/2.0/graphics-draw.php>

<https://en.wikipedia.org/wiki/Application_programming_interface>

<https://stackoverflow.com/questions/22886500/how-to-render-text-in-sdl2>