# **Technical Review & Design Doc**

### *Luke Courtney, Ishka Yao, Adam Sheedy, Cillian Murphy*

**Design Document**

**Assets**

An asset in game development is essentially any resources that are used to develop a video game. There are many different types of assets such as graphics, sound , textures and many more. Assets are important since they make the game visually and audibly appealing, this will attract more players to want to engage in your game. For our project we will mainly be using the sound and graphics assets.

**Graphics**

Graphics are very important in games design as it can completely alter the appearance of your game from something basic to something phenomenal. In our game “Submarine Mayhem” our graphics will be hand drawn and altered to fit in the game perfectly. The graphics will mainly be going on sprites, background, menu and obstacles. This asset will be stored in a graphics folder.

A drawing of a submarine

Description automatically generated

Figure 1:Graphics

**Sound**

The sound asset proves vital to making your game more realistic, for example if a game character goes in the water, you would want a splash sound as it is more realistic and can make the game more engaging to play. In our game the sound would be generated using BFXR. This is a helpful tool in creating sound effects which can then be imported into our game. We will have a sound folder to store all our generated audio.

A screenshot of a computer

Description automatically generated

Figure 2: Bfxr

**Technical Review**

**SFML**

**A green and white logo

Description automatically generated**

Figure 1 - SFML Logo

As part of the project, we will be working with the SFML C++ Library. SFML, which stands for “Simple and Fast Multimedia Library”, is a library that can be used in the development of games and other multimedia applications. SFML is a multi-platform library. This means that it can be compiled and run on most major operating systems without much if any additional work.

We have chosen to use the SFML library in our project as it is relatively simple to use, and we have past experience in using it, with such projects as Zombie, and Timber in previous semesters. As we are already familiar with SFML, it should save us from the challenge of learning a new library and using it to create a game. SFML is also multi-language, but we will be using it in C++ as out previous experience with SFML has been in C++.

SFML covers all of the needs we could have for a library. The five modules of SFML are system, window, graphics, audio and network. Of these, we will primarily be using the graphics and audio components. As our project will be created using the frame of a previous project, we will not have to spend as much time setting up the window component of the project, as it will already have been created.

SFML is well documented, and a number of books and other resources have been written on the topic. This will be a significant help with regards to development using SFML. A number of these resources are linked on the [SFML website](https://www.sfml-dev.org/learn.php), and many more can be found online on various forums and websites.

**Class Diagram**

A class diagram shows the classes of a project their attributes, methods and the relationships of each classes. The diagram provides a basic notation for other structure diagrams prescribed by UML.

A black rectangles with black text

Description automatically generated

Figure 1:Submarine-Mayhem class diagram

This diagram is the class diagram of the team project Submarine-Mayhem.

* The PlayableCharacter holds the characters positions, spawns them in and sprite.
* The Enemy and Player are connected to PlayableCharacter so they can use those methods, they also have their own inputs to handle movement and jumping.
* The Engine holds the method to spawn in the shaders, background, textures it runs the game. The Hud is use to load text in the main menu or in the paused menu.
* The LevelManager loads a new level from a text file, count the number of rows in the file and store the integers in an array to find the appropriate sprite for that number.
* The Particle class moves the particles and the ParticleSystem creates the particles into the level.
* The SoundManager associates the sounds to a buffer, the sound can be heard from a certain distance from the Player.
* The TextureHolder holds the textures for the game and uses a key-value-pair to search for the file that has the textures.

**References**

*Simple and fast multimedia library* (no date) *SFML*. Available at: https://www.sfml-dev.org/index.php (Accessed: 12 October 2023).

*Learn (SFML)*. Available at: https://www.sfml-dev.org/learn.php (Accessed: 12 October 2023).