**Game Engine Construction Report**

**User Guide**

**How to play the game**

The game demo is a space survival game where the player would need to survive 20 waves of enemies increasing in difficulty as enemies spawn is as more advanced enemy types, if the player either dies or beats all waves they can restart the game to play again by pressing esc.

**Control guide**

Invincibility debug Move Around

Restart Game



Shoots where the player is pointing to with the mouse

move aim



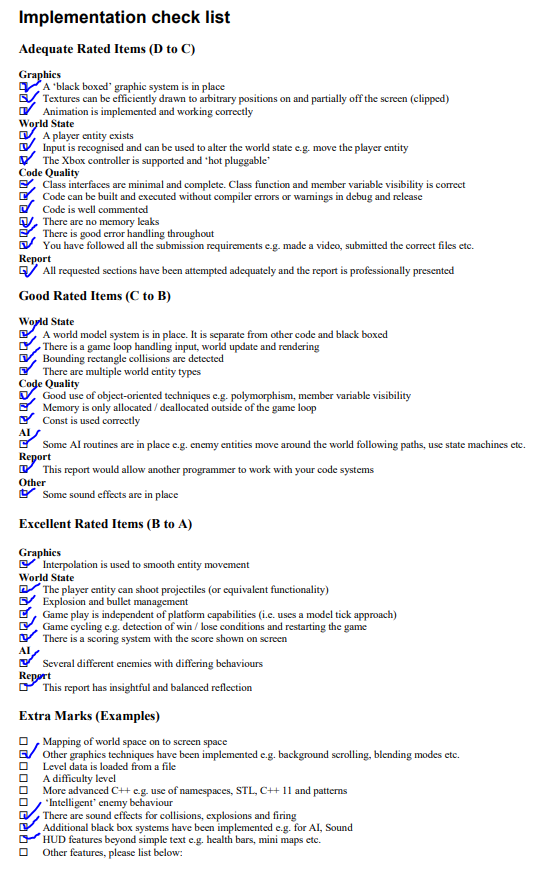
Shoot



**Known issues**

When restarting the game the first enemies spawn with already damaged health.

**Implementation checklist**

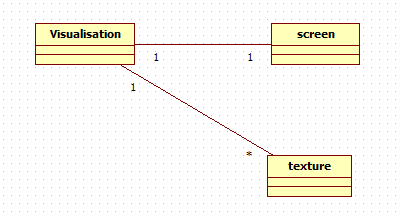


**Maintenance guide**

Visualisation System (Window)

The purpose of this system is to handle the graphics such as shapes and textures as well as managing the rendering on the screen.

This system is linked to the demo as the game tells the visualisation what needs to be drawn and where. This can be reused in other games as it isn’t completely specific to anything that’s happening in the game world. Therefore, no matter what kind of game we make with this system we can always use this system for the graphics without changing anything unless we want to add more functionality or optimisation.



HAPISPACE::BYTE

This class will be used to hold the data for the screen. This is important for all the screen and graphics functionality. Do not change the value after it has been assigned unless changing a copy.

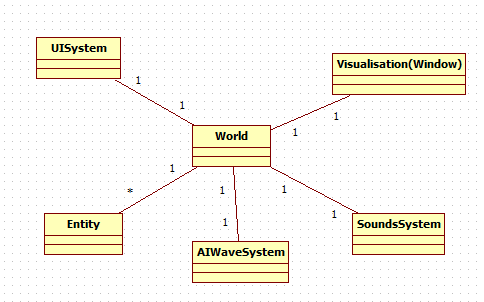
Texture

This class is used for all the sprite that will be drawn onto the screen. It includes both the data of the sprite and the functionality to draw it.

World system

This system holds all the data for the game world and holds all the other systems needed for the game. This system also runs the entire game updating the game logic and rendering.

This system is very linked to the game and how the game works therefor it would be best to reuse this system for a similar style game. However if we were to deviate to a different type of game we would have to either drastically change this system or make a new world system.



Window

This handles the visualization and the graphics side of the engine. All the draw and clear functions are handled within this class.

AIWaveSystem

This will handle waves of enemies for the player to defeat. It randomly generates wave information with specific modifiers to allow later waves to have higher tier enemies. Also has a struct for the wave data itself.

SoundSystem

This handles all the sound functionality for the game. It has functions to load sound into memory and then play them sounds. Also includes error checking to say if a file doesn’t exist.

UISystem

This handles the user interface. It has functions to draw the text that shows when the player wins or loses the game. It also has functions to draw the amount of life or a healthbar.

Entity

This class is the base for every game object that is present within the engine. It will handle all the functionality and data for everything that is present within the game world.

It has multiple child classes for things like the player, enemy, bullet and scenery objects.

Conclusion

Overall I am satisfied with my final implementation. I would have wanted to do further improvements however I do feel like I’ve learned a lot that I very much would like to incorporate into future projects. I have thought about a structure like this a few times before but never really had the opportunity to implement it. I feel glad that I was able to add features like the system that handles enemy waves but would have loved to add in a file reading function for different settings like level data, difficulty and various other settings however I feel given the time I could easily incorporate it.

I feel time management worked out pretty well considering my other big assignments. However I do feel I could further improve my time management skills to be even more efficient since I would have liked a bit more time especially to make the engine less hard coded.

I feel I could further improve my coding habits by trying to add more comments. I was able to have a list of things to do with comments and made things easier for myself by leaving a few comments along the way every now and then. But I still need to get into the habit of keeping it up since I have never really been one to comment while coding.

The thing I am most satisfied with is my understanding of both the engine structure and coding in general as with the knowledge I have gained I would love to move on to bigger projects. One such project I feel I would like to do at some point in the coming years would be to make a game engine in opengl. Which I feel so much more confident in the idea now that im able to structure code better.