Module	SEPR
Year	2019/20
Assessment	2
Team	Salt N Sepr
Members	Alberto Lee, Archie Godfrey, Jacqueline Eilertsen, Jake Schoonbrood, Joshua Levett, Matteo Barberis
Deliverable	Implementation (Impl2)

Implementation

We chose to use open-source assets to style the majority of the game in order to reduce the development time. This meant more time could be spent on the code and less on creating custom assets. However, we did create our own fire truck assets so that we could implement a fake 3D effect to make our game more unique. The assets that were used can be found in the references for this page. This document outlines each requirement from the product brief for assessment 2 in bold, followed with how it is met.

Each Fire Engine must have a unique spec in terms of the volume of water it can carry, its speed, the range and delivery rate of its water cannon, and the amount of damage it can take before it is completely destroyed.

Each fire truck has its own array containing all of the properties of the truck, including the amount of water it can hold, its speed, its range, its delivery rate and the amount of damage it can take. Using our test cases as proof, we have the tests CONTROL_SPRAY, VARIED_TRUCKS, CONTROL_TRUCK, each of these test cases passed the tests. Each fire truck has a status bar for its health and water volume. The truck is destroyed when the health bar is empty, and the truck is no longer able to spray water once the water volume bar is depleted.

Fire Engines need to return to the Fire Station to repair and refill.

The fire station has a range and when the firetruck is inside, it will be repaired and refilled. The functional requirement RETURN_HOME_FUNC is representative of this feature, and in the test cases spreadsheet it has passed.

Each ET fortress must have a unique spec in terms of the range of its defensive weapons, the amount of damage these weapons can deal to Fire Engines over a period of time, and the volume of water it takes to flood.

ET fortresses have a specific attack range which is based on the size of the fortress. The amount of water required to flood a fortress is based on its size as well. Each ETFortress produces projectiles which damage the firetruck if they collide. This has been tested and is shown in the testcase

There should be at least two Fire Engines and three different ET fortresses based (possibly loosely) on real locations in York.

The screenshots from running the game can be seen in the appendix below, (1) and (2) show a fire truck being modelled in the software Magicavoxel and the two different playable fire trucks respectively. Each firetruck is shown to be different by a specific colour and number on the rear of the truck for each. Screenshots (3), (4) and (5) are representative of Clifford’s tower, York Minster and York Railway station respectively.

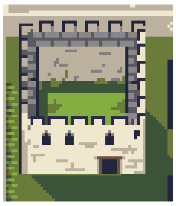
The game is won when all ET fortresses have been flooded and is lost when all Fire Engines have been destroyed.

The test cases, WIN_GAME_FUNC, DESTROY_ENTITIES_FUNC, that are derived from our requirements passed our tests, thus the game can be won by “destroying” each ET fortress.

Appendix:

(1)

(2)



(3)

(4)

(5)

References:

[1] G. City, "Gallet City by Adam Saltsman", *itch.io*, 2020. [Online]. Available: https://adamatomic.itch.io/gallet-city

[2] "SEPR Documentation", *Sepr-documentation.firebaseapp.com*, 2020. [Online]. Available: https://sepr-documentation.firebaseapp.com/testing