

Evaluation and Testing Report

Evaluation approach

First and foremost, we had to make sure our final product met all the criteria within the initial brief. This was achieved by going through all the original requirements, checking whether they align with and fully cover the assessment brief. No change was needed to this set of requirements as we believed they had properly included everything within the original brief.

For Assessment 4, the client asked for a change in the existing requirements to be implemented. On receiving the announcement, we got together as a team and went through it together, noting down any uncertainties for further clarification with the client. Addressing these ambiguities prior to the development phase consolidated our confidence with the direction we were heading in. After all internal and external discussions had finished, two new functional requirements **F19** and **F20** were added to the original set of requirements to reflect our understanding of what were asked for along with their associated risks.

For this assessment, we also carried out rigorous bug clearing and code refactoring on the inherited code.

The newly revised list of requirements later on played a crucial role in our evaluation process. As the game got closer to being finished, we started checking if requirements had been fulfilled by cross checking them with the game's functionalities. All requirements had been achieved apart from one (**F16**), which we agreed as a group to not implement and was noted down to be referred to in the Requirement Report. This process of cross referencing our list of requirements to the final product ensure that we had fulfilled everything asked for in the brief, as well as satisfied with the final product as the requirement list had been put together as team after thorough discussion and input from every members.

Last but not least, once we finished the game the final executable was exhaustively checked to make sure that it was running as expected. This was achieved through multiple playthrough test runs which allowed us to validate various different aspects of the game.

Testing approach

We consider appropriate quality code to be clear, concise and reasonably efficient whilst adhering to the requirements specification.

To test how our code adhered to our chosen definition of appropriate quality we primarily used black-box testing. More specifically, testing of the final product was done by one external member (with no knowledge of the game) and two of the developers. The external member had no cheats or methods to speed up the progression of the game, whereas one of the developers used a hotkey to add gold - removing the need for lots of grinding of battles allowing the testing of features such as game completion and boss battles to be more effectively tested.

We felt that black-box testing was the most effective way to test the game at this point in its production as the majority of the inner workings of the code had been created and tested in previous assessments. We therefore felt that the method of black-box testing would be most effective at testing the game's value as a final product as it would mimic a real playthrough of the game and we would be able to test all the features and make sure all the requirements had been met. With the use of the hotkey to add gold, we could also focus on manually testing the aspects of the game that were being developed to meet the brief for assessment 4, for example: crew members were earned through the

defeat of a boss so to speed up the testing we could generate gold and upgrade our ship quickly so as to test this functionality.

Some of our findings from this testing are as follows:

- Detection for completion of the minigame is slightly buggy; it doesn't always detect that you have won the game and can sometimes cause the game to crash.
- Dragons can sometimes breach the edge of the map

Requirement Evaluation

With the conclusion of the project's production cycle in Assessment 4, it was essential that we completed all feasible requirements given to us, with an additional emphasis on the updated requirements given to us at the beginning of this quarter - alongside ensuring that the brief was met. Below is a list of all requirements **[1]** along with quick descriptions of how we met them or why we feel they were unnecessary.

Functional Requirements

Requirement **F1** was carried out by using the islands on the map to represent different colleges and departments of the University. **F2**, **F3** and **F4** were all implemented throughout the process, while **F5** was fulfilled by have multiple layers of difficulty; sea monsters, ships, and bosses, with the bosses being difficult to beat without having previously battled and upgraded. Requirement **F6** was completed in production, as was **F7**, which was developed by varying the amount of points received depending on whether it was a ship or boss that had been defeated. Similarly, **F8** was fulfilled, and **F9** was carried out by having colleges be captured once their respective boss had been defeated. The team implemented **F10** by having departments used purely for buying upgrades and health. This was done with requirement **F11** in mind, as these are bought with the gold acquired from enemies. We fulfilled **F12** by creating an optional maze minigame that rewards the player with points if completed successfully. Requirement **F13** was completed when the team worked on Assessment 3, whilst **F14** has been fulfilled as the player ship is displayed in the sailing screen whilst enemy ships are only shown on the combat screen. We have implemented the ability for players to save their game and come back to that save in a different session, satisfying requirement **F15**. As a group, we decided to miss out **F16**, as we felt the game was complex enough with sea monsters, ships of varying difficulty and a minigame, and a weather system would only confuse this balance. Multiple points of damage (hull and sails) were implemented into the combat system, meaning different outcomes depending on the area of attack, and the completion of **F17**. Further, requirement **F18** was completed as once colleges are defeated, they can be used in a similar way to departments or the player's home college. With Assessment 4 came two new requirements, **F19** and **F20**, which we fulfilled by implementing the ability to obtain crew members with differing perquisites for the player and the possibility of challenging sea monsters whilst sailing.

Non-Functional Requirements

We believe that we have sufficiently fulfilled requirement **NF1** by providing relatively intuitive controls and gradual increase in difficulty as the player discovers more of the game. **NF2** has also been tested, whilst the game was created with **NF3** constantly in mind. Further, **NF4** was strictly adhered to in the conception of the project.

Constraint Requirements

Requirement **C1** has been completed by submitting Assessment 4 before our given deadline, whilst throughout we have adhered to **C2**. Further, we have attempted to keep as close to **C3** as possible by constantly testing and adjusting the game where suitable. Finally, we believe we have also fulfilled **C4** by relating the game to the University of York.

References

- [1] Rear Admirals Assessment 4 Requirements, 2019. [Online]. Available:
<https://therandomnessguy.github.io/SEPR/Assessment/4/Updates/Req4.pdf>
[Accessed 30 - April - 2019]