

Requirements Elicitation and Negotiation

The project brief provided a high-level overview of some of the requirements that would need to be included in our research. In the first session as a group we read through this brief together to give everyone a solid understanding of what was being asked of us, while making notes of the high-level requirements that were mentioned - such as it being possible to capture another college via combat. During this process we developed a series of questions which needed to be clarified to ensure we were better informed about what was desired by the stakeholders. The primary stakeholders are the customer, University of York Communications Office and the players of the game (our SEPR cohort as well as prospective students). Building on this, we decided on methods which would allow us to further elicit the requirements for the game. Discussion as a group led to the decision of 3 more methods for elicitation:

- Meetings with the customer - This method of requirements elicitation would give the customer a chance to feedback on our ideas as a group meaning that if they had any disagreements with our proposal we could discuss and come to a conclusive idea. As well as this, we used the opportunity of meeting our customer to ask our questions about the brief in order to elicit more information of what was actually wanted by him. We are designing a game for the customer so it needs to be what they desire - rather than what we want in a game. Despite this we can suggest ideas to him and explain to him in meetings why we think certain gameplay choices should be made. In reflection, this method of requirements elicitation was an effective method - although we found more efficient results through emailing the customer due to his fast and detailed replies.
- User survey - One of the other main stakeholders of the game would be the players of the game. Therefore in order to gain an overview of what they would like in a game, we designed a survey which provided us with useful information regarding the requirements of the users. Using a survey is a fast way to acquire input from the players to see what they want in a game. Using this method allowed us to make better decisions as we had input from players - rather than just our group of 6 making a decision. Decisions for the game design were then easier to make due to the players influence in the survey meaning that we could design requirements based on what they wanted. For example, 82.4% of the responses agreed that the idea of resources in order to sail between places was a good idea. This is advantageous as realising what the players wanted in a later stage of the project, as explained by Dr Gary Morgan of ETAS Ltd, would increase the cost of fixing the problem 10 fold per stage.
- Prototyping - Prototyping is a good method to give a visual representation of what our vision is in order to show our ideas to the customer as well as to give us better understanding of how it would work and what extra things would be needed. By creating prototypes [1] of the combat system as well as the map and mechanics, we were able to

more easily express our ideas to the customer during our meeting. Doing this meant that he could fully understand how our ideas would come together in the game so that he could discuss with us any disparities between his and our ideas. As well as this, the prototypes made the elicitation of requirements easier for us as we could see how the game would come together and what would be need for it to run smoothly.

Presentation of Requirements

In order to fully develop our list of requirements we took inspiration from the IEEE format for gathering requirements [2]. The functional requirements table provided, on page 18, was helpful in crafting our own table. We decided not to include the 'Priority' column as it is implied in the wording of the document - words such as 'must' and 'could' for example. The 'Description' section from the IEEE format was renamed 'Requirements'. We added a 'Notes' column which included environmental assumptions, for clarity and detail; risks, as it is important to acknowledge times where our system may require more action and areas with a potential to stall our production; alternatives, where we attempted to offer solutions in case our original plan became unattainable, and fit criteria, which we felt was needed for some requirements to justify why they have been added. Perhaps our most important section was the 'Extends' column in system requirements which links the system requirement to the relevant user requirement, there is usually a many-to-one mapping with this.