Requirements

Group Name: Team 6
Group Number: 6

Ryan Bulman
Frederick Clarke
Jack Ellis
Yuhao Hu
Thomas Nicholson
James Pursglove

a) Introduction to requirements

Single Statement of Need: Design a game for usage in university display situations based around pirate ships and York university colleges, as well as a customer looking to eventually market and sell the game.

We held a meeting within our group to discuss the brief and the initial ideas, and what questions we had. The questions were separated into different sections such as general ones about which platform the game should run on, or specific ones about the details of combat. This list was refined to:

- Remove duplicates
- Remove questions that were already answered in the brief
- Give certain questions priority over others
    - If, for example, we asked if combat should be real time or turn based, questions about specific implementations of an unnecessary system could be ignored

With this list of questions we booked a meeting with our client and presented them to him. We took notes of the answers, and followed up with any additional queries that were raised from the meeting.

Research and implementation of methodology

The primary source for our method in eliciting the requirements was the lecture and information provided by the university. The system presented gave us an effective way of distinguishing between the types and levels of requirements, and also an effective way of displaying the requirements as a set of tables. This gives us three different forms of requirements:

- User requirements
    - The broadest and highest level requirements. These define general requirements that the client has about the game without necessarily specifying functionality
- System Requirements
    - Contain more detail about how those user requirements are to be implemented.
- Non-functional requirements
    - Features that need to be added for non-functional reasons – the product will work without them, but they might improve the games accessibility for example.

This method also calls for a priority rating for the user requirements which was divided into:

- Shall
    - Deemed necessary by the client for the game to be considered complete
- Should
    - Not strictly necessary for the game to be considered complete, but still desirable to be included
- May
    - Ideas that are not necessary, introduce the potential for additional workload, and are likely not going to be included. But they have not been explicitly ruled out.

b)
The following requirements are a selection that we elicited. The full list is available on the website at the following link:

https://tomnicho.github.io/eng1-team-6/requirements.html

This selection was based on requirements that needed additional explanation to either discuss their meaning, or what would be needed in order to deem them complete and passed. The requirements omitted from this document are self-evident if they are completed in the finished game.

User Requirements

ID	Description	Priority
UR_PLATFORMS	The game shall be available on desktop for MacOS, Linux and Windows systems	Shall
UR_SCALABLE	The game shall be playable on most PC's and laptops.	Shall
UR_FIRST_TIME_PLAYER	The game shall be easily understood by a player who has not seen the game before	Shall
UR_CHILD_APPROPRIATE	The game shall be appropriate for children to play.	Shall
UR_GAME_LENGTH	The game shall be playable in a single session	Shall
UR_COLLEGE_FIGHT_BACK	Colleges should be able to fight back in some seemingly intelligent way	Should

Non-Functional

ID	Description	User Requirements	Fit Criteria
NFR_SCREEN_SIZE	The game shall make good use of screen real estate on most screen sizes.	UR_SCALABLE	Adjust for screen sizes between 13 and 27 inches.
NFR_HARDWARE_REQUIREMENTS	The game shall not have high hardware requirements.	UR_SCALABLE	Runs well without issues on a department laptop.
NFR_CRASHES	The game should not crash.	UR_RELIABILITY	Played 10 times while engaging with all systems without it crashing.
NFR_EPILEPSY	The game shall not use any forms of flashing images or repeated patterns that are likely to cause seizures.	UR_EPILEPSY	Avoids common epilepsy triggers[1]

- UR_PLATFORMS
    - This requirement has an assumption that each of these platforms specified are up to date and work on the version of the engine we look to use. There is the risk that, for example, some versions of MacOS are incompatible while others are.
- UR_SCALABLE
    - This is a general requirement which follows the same logic - the game may work on some while not running on others. In this case it should be deemed complete in the case of running on the variety of machines in our group, and as per NFR_HARDWARE_REQUIREMENTS, a department computer
- UR_FIRST_TIME_PLAYER
    - This requires some assumptions about what would be unclear to a first time player, and also some assumptions about what a first time player would intuitively understand. Some might not even get the premise while others might not necessarily need to be told anything. From our perspective, if we have added enough tutorial and information to explain the concepts on each launch this requirement will be met.
- UR_CHILD_APPRORIATE
    - This has some environmental context, as what is considered child friendly might not be shared by everyone. As per NFR_CHILD_APPROPRIATE, we have chosen to design our game around the PEGI standard, and this requirement can be deemed complete if the game conforms to PEGI 7[2]
- UR_GAME_LENGTH
    - For the purpose of this use case, we have defined a single session as being a maximum of about 5 minutes, meaning this requirement is met if the game does not take that long to achieve the objectives.
- UR_COLLEGE_FIGHT_BACK
    - This requirement uses the terminology “seems intelligent”, our group understood this to mean that the colleges have some potential to aim for the player, and are able to hit them enough to pose a threat
- NFR_SCREEN_SIZE
    - This requirement also features a deal of ambiguity with an almost infinite amount of combinations of aspect ratios and resolutions available for screens. This will be deemed complete if the game automatically adjusts for the different sizes without compromising the image itself.
- NFR_HARDWARE_REQUIREMENTS
    - This relies on the assumption that we can source a department laptop to test our game on. If this is not the case, we can make an assumption that within our group the range of computers at our disposal would be enough of a representation that if the game does not take too much resources, it will at least run on a department laptop.
- NFR_CRASHES
    - The fit criteria is relatively arbitrary here, and there is always the possibility that a crash is in the code but it remains uncaught by our group. Therefore we have to set a point that means we test the game enough to eliminate these issues, but also means we are not pointlessly testing.
- NFR_EPILEPSY
    - The clear risk with this requirement is that we fail, and there is an epilitic trigger in the game. We therefore have to be especially sure that this is met by avoiding rapid sprite changes for example.

References:

1. http://gameaccessibilityguidelines.com/avoid-flickering-images-and-repetitive-patterns/

2. https://pegi.info/index.php/what-do-the-labels-mean