SEPR 2019/20 Assessment 1

Team CheatCodez

Jonathan Grout, George Lesbirel, Cheuk Wang Wu, Lauren Quarshie, Muaz Atif, Lillian Coultas

Requirements

Introduction

In order to gain domain knowledge about the game and its requirements, the two main methods of requirements elicitation used were brainstorming sessions and interviews [1] with the stakeholders.

A whole group brainstorming session took place before the interview with the stakeholders, in which our primary objective was to prepare an agenda of relatively open-ended questions. At the end of the session, a document consisting of questions about requirements given in the product brief, as well as potential new requirements, was produced. An interview was then conducted with Dimitris Kolovos, our stakeholder and liaison for communication with the York Communications Office, in which we recorded the responses given to the questions and assumptions we had produced.

After the interview a second brainstorming session was held in which the requirements that were left up to the development team were discussed. The main objective of this session was to decide which specifications would be possible to implement within the timeframe and on budget, while also producing a game which best represented the specifications given by product brief and our stakeholders (i.e. the most impressive finished product, and the most engaging game.)

In accordance with IEEE standard 729 (IEEE standard glossary of software engineering terminology), a requirement is defined as :

(1) "A condition or capability needed by a user to solve a problem or achieve an objective

(2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification or other formally imposed documents

(3) *A documented representation of a condition or capability as in 1 and 2"* [2]

Using these definitions as a guide, we have chosen to present the requirements in the form of three linked tables which are sorted into user requirements, functional requirements, and non- functional requirements. This method of presenting the requirements allows relationships between requirements to be recognised easily by both the development team and the stakeholders. Each requirement can be identified with a unique meaningful ID to ensure that the requirements can be referred to easily in different contexts. Each user requirement has a priority attached to it:

- Shall: the finished product will conform to this requirement.
- May: the product might conform to the product, as time and other factors allow.

The non-functional requirements have fit criteria attached to them, which can be used to quantify the intentions of the stakeholder, and the success of the product.
Requirements are presented in three linked tables, with each identified by a unique ID (the format of which is shown for each table).

List of Requirements

User Requirements Table - UR_NameOfUserRequirement

ID	Description	Priority
UR_USER_EXPERIENCE	The system shall offer a pleasant user experience	Shall
UR_ENGAGING	The system shall be engaging for the user, for the entirety of the gameplay	Shall
UR_PRODUCT_BRIEF	The system shall conform to all stakeholder requirements in the product brief as discussed.	Shall
UR_SUITABLE	The system shall be suitable for use at open days for prospective students	Shall
UR_EASY	The system shall be easy for players to pick up with no instructions necessary	Shall
UR_ACCESSIBLE	The system shall be usable for all users, and shall comply with the Disability Discrimination Act 1995@j	May

---

Functional Requirements Table - FR_NameOfFunctionalRequirement

ID	Descriptions & User Requirements ID
FR_ENGINE_SPECS	The system shall have unique specs for each fire engine - **UR_PRODUCT_BRIEF**
FR_REPAIR_REFILL	The system shall allow fire engines to repair and refill at the fire station. - **UR_PRODUCT_BRIEF**
FR_FORTRESS_IMPROVEMENT	The system shall allow the ET fortresses to improve over time, and become more difficult to flood. - **UR_PRODUCT_BRIEF**
FR_FORTRESS_SPECS	The system shall have unique specs for each fortress - **UR_PRODUCT_BRIEF**
FR_ENGINE_NO	The system shall have at least 4 fire engines in play - **UR_PRODUCT_BRIEF**
FR_FORTRESS_NO	The system shall have at least 6 ET fortresses based on locations in York - **UR_PRODUCT_BRIEF**
FR_FORTRESSES_FLOODED	The player shall win once all fortresses have been flooded, and the game shall end - **UR_PRODUCT_BRIEF**
FR_ENGINES_DESTROYED	The player shall lose once all the fire engines have been destroyed and the game shall end - **UR_PRODUCT_BRIEF**

FR_MINI_GAME	The system shall provide an embedded mini game for the user, which shall be in a different style to the main game but conforms to the same theme. - **UR_PRODUCT_BRIEF**
FR_DESTROY_STATION	The system should have the ETs destroy the fire station a fixed amount of time after the game begins - **UR_PRODUCT_BRIEF**
FR_NO_REPAIRS	The system shall not allow fire engines to be repaired or refilled after the fire station has been destroyed - **UR_PRODUCT_BRIEF**
FR_APPROPRIATE	The system shall have no inappropriate violence, and should be relaxing and enjoyable to play - **UR_SUITABLE**
FR_GAME_DIFFICULTIES	The system shall provide the user with a range of different game difficulties. - **UR_ENGAGING**
FR_PLAYER_NAME	The system shall not require a player to log in or set up an account, just to enter a name for a leaderboard - **UR_PRODUCT_BRIEF**

Non-Functional Requirements Table - NFR_NameOfNonFunctionalRequirement

ID	Description	User Requirement ID(s)	Fit Criteria
NFR_PLAYTIME	The playtime of the game will be between 10-15 minutes	UR_ENGAGING	90% of the games will have a playtime of 10-15 minutes
NFR_CONTROLS	Game controls should be explained to the player before the game begins	UR_EASY	Game controls are easily understandable by first time player after explanation.
NFR_FUNCTIONS	Functions of game (e.g. showing the map) require clear instructions to inform the user of their purpose, i.e. in a user manual users can access	UR_EASY	The game functions will be available to the user within 5 seconds from when the user requests them.
NFR_SIMPLE	All user instructions or written statements should be in plain english and easily understood i.e. no abbreviations or colloquial language	UR_EASY UR_ACCESSIBLE	Eight out of ten new users should be able to understand what the written statements are trying to convey upon first read through.
NFR_READING_TIME	All user instructions or written statements	UR_ACCESSIBLE	All instructions will be shown until the user

	should allow sufficient time for users to read through, i.e. the user dismisses the text and there's no time limit on reading	UR_EASY	dismisses them; the instructions should be able to be dismissed within 3 seconds of the user action.
NFR_COLOURS	The game should not rely on colour coding alone to distinguish things, or should have a colourblind assist mode.	UR_ACCESSIBLE	95% of users who are colour blind should be able to play the game as well as a regular user.

Assumptions were made about future development - more on Appendix 1.4

Assumption	Requirement ID
We assume that our users will have an age range of 16-19.	• UR_SUITABLE
We assume that data we fetch from online will remain the same	• *All~*
We assume users will be given sufficient time to experience the whole game.	• UR_ENGAGING • NFR_PLAYTIME
We assume that the code will be maintained	
We assume that the costs of development is negligible	
We assume that the input data from the computer (e.g. keyboard and mouse) will not change standard (e.g. ASCII)	• *All~* • NFR_CONTROLS • UR_ACCESSIBLE

Risk Association - please refer to Risk1.pdf for corresponding risks

Requirement ID + exp.	Risk ID	Requirement ID + exp.	Risk ID
All If function names do not match, this could hinder maintenance and future development.	2	**FR_MINI_GAME/NFR_PLAYTIME** As a small part of the game, time consumed could easily be unbalanced.	11
All Productivity issues (personal/team) could affect any and all parts of the project.	10	**NFR_COLOURS / ACCESSIBLE** Lack of understanding towards colour schemes could lead to poor system design and implementation.	14 -> 5 -> 2
NFR_SIMPLE If we rush the development then the user manual could get delayed and then be rushed.	11	**UR_ACCESSIBLE** We are likely to underestimate the complexity for accessibility, hence causing unexpected delays.	4