MiKroysoft

Module	SEPR
Year	2019/20
Assessment	1
Team	MiKroysoft
Members	Daniel Crooks, James Rand, Irene Sarigu, Alfie Jennings, Charlotte Clark, Jasper Law
Deliverable	2) Requirements

2a

The first stage of the process was requirements elicitation,where information was gathered and used to derive the system requirements **[1][2]**. The elicitation activities came in the form of a product brief, a SSON (Single Statement of Need) and an interview (with our key contact in the client company, professor Dimitris Kolovos). The interview was useful for developing our user requirements, as clients are directly impacted by the system, making them direct stakeholders **[1]**.

The next stage of this process was requirements analysis, which is where the team made sure that no requirements overlapped or were missing **[2].** To do this we revised the requirements register by comparing them to the product brief and interview. We referred to lecture slides regularly to help us decide what was a useful requirement and what was not **[3]**.

The next stage of this process was documentation, where requirements and their descriptions are presented **[1][2]** for stakeholder approval **[1]**. We referenced lecture slides when formatting our requirements register **[3, Slides 21 – 24]**, to ensure readability. Every user requirement in the documentation includes a unique ID, a description (including assumptions, risks, and requirement sources) and a priority. Requirement priorities were decided by a modal vote during team meetings.

The functional requirements were decided on during a brainstorming session, studying the different types of functional requirements (transformation, invariant and failures **[3, Slide 14])**.

The non-functional requirements were also based off lecture slides **[3, Slide 13] [3, Slide 15 – 20]** and made sure that every non – functional Requirement was measurable.

During the validation and negotiation stage **[1]**, the client was shown the initial table of user requirements, and feedback was obtained. In response to client feedback, some requirements were re-worded for clarity, duplicate requirements were removed, and certain requirements were assigned new fit criteria.

Lastly, the team kept regular communication with the client to verify requirements, as recommended by the Web Systems Engineering team **[1].** Any changes made by the MiKroySoft team to the requirements are known by the client, who decides whether the requirement is appropriate for the system or not, and if it will be implemented.

(2) Requirements – Part B

PB = INFORMATION FROM PRODUCT BRIEF, IN = INFORMATION FROM INTERVIEW WITH CLIENT
Shall = Will be added to the system, should = might be added to the system, may = optional task

ID	Description (Source)	Priority
UR_SIN	This will be a single player game, playable offline. [**PB]** **Assumption: The user has already downloaded the game files.**	Shall
UR_THE	The game follows the theme given by the client. **[PB]** **Assumption: Availability of adequate quality assets in alignment with the theme.**	Shall
UR_ACC	The system has the appropriate measures in place to be accessible to a wide range of users. **[IN]** **Assumption: Hardware is available to accommodate accessibility features**	May
UR_LRN	The game can be understood quickly, users can build strategies. **Assumption: Users with new copies of the game have never played the game or any similar game before.**	Should
UR_ENGI	Game will have at least fire engines with unique specifications. **[PB] [RA_DIFF]**	Shall
UR_REF	In the game, fire engines are able to refill at the fire station. **[PB] [RA_GOAL-MET]**	Shall
UR_FORT	The game will provide at least 6 ET fortresses with unique specifications. **[PB] [RA_DIFF]**	Shall
UR_FLD	Overtime, the ET Fortresses will upgrade, becoming harder to flood. **[PB] [RA_DIFF]**	Shall
UR_END	Game ends correctly according to the client's specification. **[PB]**	Shall
UR_ABLE	Game will make sure that, at some point, the fire engines stop being able to repair or refill themselves. **[PB] [RA_DIFF]**	Shall
UR_MINI	Game has a playable mini game integrated, in line with requirement UR_THE. **[PB] [RA_DIFF]**	Shall
UR_DIFF	The game shall become progressively more difficult over time. **[IN] [RA_DIFF]**	Shall
UR_ENGA	The system ensures that the gameplay engages users. **[IN] [PB]**	Shall
UR_COMP	The system will ensure a sense of competitiveness between different players. **[IN]**	May
UR_TIME	The game should be completed on average in around five-to-ten minutes, as specified by the client. **[IN]**	Should

UR_DEST	The game ensures that once a fortress or station is destroyed, it cannot be rebuilt, stopping all activity or interactions related to that building. **[IN]**	Shall
UR_POI	The game will allow players to collect and spend in-game currency on items that will upgrade their fire engine. **[IN] [RA_DIFF]**	May
UR_MOB	The game will be designed for use on a computer but can be ported to mobile and be used with ease. **[IN]**	Shall
UR_STRY	The game will ensure that the user is aware of the storyline in place. **[PB] [RA_STORY]**	Shall

Functional Requirements [Table]

ID	**Description**	**User Requirement**
FR_EXTRA_TERRESTRIAL	The system will ensure that the game contains an appropriate number of ETs as enemies.	UR_THE
FR_FIRE_ENGINE	The system will ensure that the game contains Fire Engines as the entity that users control. **[RA_DIFF]**	UR_THE
FR_CONTROLS	The system shall provide intuitive controls, increasing learnability and memorability.	UR_LRN
FR_TUTORIAL	The system will provide an easy to understand tutorial at the start of the game. **Assumption: The user is aware that the game is in tutorial mode, and is able to skip it**	UR_LRN
FR_ENGINE_VOLUME	The system will ensure that each fire engine has a unique level for the maximum volume of water it can store. **Assumption: Each fire engine's maximum water volume is clear to the user**	UR_ENGI
FR_ENGINE_SPEED	The system will ensure that each fire engine has a unique maximum speed. **[RA_DIFF]**	UR_ENGI
FR_ENGINE_ACCELERATION	The system will ensure that each fire engine has a unique maximum acceleration. **[RA_DIFF]**	UR_ENGI
FR_ENGINE_RANGE	The system will ensure that each fire engine has a unique shooting range. **[RA_DIFF]**	UR_ENGI
FR_ENGINE_DELIVERY_RATE	The system will ensure that each fire engine has a unique delivery rate of water. **[RA_DIFF]**	UR_ENGI
FR_ENGINE_DAMAGE	The system ensures that each fire engine can take a unique amount of damage (before destroyed). **[RA_DIFF]**	UR_ENGI
FR_FORTRESS_RANGE	The system will ensure that each ET Fortress has a unique range for its defensive weapons. **[RA_DIFF]**	UR_FORT
FR_FORTRESS_DAMAGE	The system will ensure that each ET Fortress has a unique damage per second dealt to fire engines from its defensive weapons. **[RA_DIFF]**	UR_FORT

FR_FORTRESS_VOLUME	The system will ensure that each ET fortress has a unique maximum volume of water that it can handle, before flooding and ceasing activity. **[RA_DIFF]**	UR_FORT
FR_UPGRADE	System will upgrade the remaining ET Fortresses when one is destroyed. **[RA_DIFF]**	UR_FLD
FR_WIN	The system will ensure that the game is won only when all ET Fortresses are destroyed. **[RA_GOAL-MET]**	UR_END
FR_LOSE	The system will ensure that the game is lost when all the fire engines have been destroyed. **Assumption: User is made aware that the game is lost, before resetting.**	UR_END
FR_INCORRECT	The system will never allow the game to end without either FR_WIN or FR_LOSE being satisfied.	UR_END
FR_LOCATION	The system will ensure that after a fixed amount of time the ETs find out the location of the fire station and destroy it.	UR_ABLE
FR_LEADERBOARD	The system will ensure that there is a leaderboard (based on finish time), so that players can compete against each other.. **Assumption: The leaderboard is easily accessible. After a game, the user is asked to provide a name for the leaderboard.**	UR_COMP
FR_EARNED_POINTS	The system shall never allow users to have more points than they have rightfully earned.	UR_POI

Non – Functional Requirements [Table]

ID	Description	User Require-ments	Fit Criteria
NFR_CON	The system will cater for visually impaired users by providing a high contrast version of the game. **Assumption: Toggling high contrast mode is easily accessible from the options menu**	UR_ACC	Contrast enhanced to the level specified by a group of at least ten visually impaired product testers.
NFR_LANG	The system will ensure that all text presented on screen are in plain and correct English, allowing the user to engage and learn the game more easily.	UR_LRN UR_ENGA	All presented text verified by Microsoft Word spelling and grammar checks, and QC checked by at least 2 team members.
NFR_ MANUAL	The system will provide a manual which will be used to explain to the user how the game works. **[NFR_LANGUAGE_USED]**	UR_LRN	The user manual shall describe all system components in adequate detail; as defined by stakeholders.
NFR_HW	The system will run on low – end hardware.	UR_ACC	Maximum hardware requirements of: 512MB RAM 1GB Storage 1.5GHz Processor

Bibliography

[1] Web Systems Engineering (17th December 2015). *Requirement Engineering Process. YouTube*. [online] Available at: https://www.youtube.com/watch?v=_llqRnlrzWw [Accessed 7 Nov. 2019].

[2] Easy Engineering Classes (18th November 2017). *Software Requirement Engineering [Requirement Documentation] SE Lectures Hindi English*. *Youtube.* [online] Available at: https://www.youtube.com/watch?v=ABDIyiW3b7g [Accessed 31 Oct. 2019].

[3] Requirements Engineering Lecture, Lecture 2 of SEPR, Dimtris Kolovos, University of York, October 2019