Requirements Evaluation

Mozzarella Bytes | Team 18

Daniel Benison

Elizabeth Hodges

Kathryn Dale

Ravinder Dosanjh

Callum Marsden

Emilien Bevierre

Requirements

Single statement of need: Build a single-player game suitable for prospective students and their guardians to play on open days, that involves moving fire engines between a Fire Station and alien fortresses, avoiding alien patrols on the way, and attacking alien fortresses when the fire engines' water cannons are within shooting range.

The description of the requirements engineering process, elicitation of requirements, requirements change management, validation of requirements, requirement presentation and our use case can be found at: https://emhodges.github.io/SEPR-game/assessment1/Req1.pdf

Colour Coding Key

🟥	Shall
🟧	Should
🟩	May

User requirements

User ID	Description	Source	Priority	Updates	Evaluation
UR_WIN	The player wins if they flood the ET fortresses before the ET fortresses and ET patrols destroy all of the player's fire trucks	Product brief	Shall		*This works exactly as expected. When the game is played and all fortresses are destroyed by the user before the patrols and fortresses can destroy all the trucks the game ends and the user is informed that they have won.*
UR_LOSE	The game is lost if all the player's fire trucks have been destroyed before the player has flooded all of the ET fortresses	Product brief	Shall		*This works as intended. When all the fire trucks are destroyed – either by a fortress or patrols, the game is ended and the player is told that they have lost.*
UR_SAVE	***The player should be able to save the state of the game at any point and resume a saved game later***	***Product brief***	***Shall***	***New requirement due to change in product brief***	*Players are easily able to save and resume games – however they are not informed when a game has successfully saved. This would be a useful addition as users may accidentally save more than one copy of the current game that they are playing by clicking multiple times.*
UR_DIFFICULTY	***There should be different levels of difficulty in the game***	***Product brief***	***Shall***	***New requirement due to change in product brief***	*There are three difficulty levels in the game. Each level is distinctly different from each other in terms of how hard they are to complete.*

UR_MINI_GAME	The game should include a minigame	Product brief	Should		*There is a minigame included in the main game, initiated when a truck collides with a patrol.*
UR_MINI_GAME_THEME	The minigame should be different in style, but aligned to the theme of the main game	Product brief	Should		*The mini-game is indeed a different style, as it is a dance off game and not too similar to the main game. However it is in the same theme of the main game as it is an 'ET' dancing with a fire fighter and it has the same style graphics and colours.*
UR_REPAIR	Fire trucks can be repaired and refilled at the fire station	Product brief	Should		*This works very well as when trucks are returned to the fire station they refill at varying speeds. How full a truck is of both water and health is clearly visible.*
UR_FIRE_TRUCKS	There must be at least four fire trucks	Product brief	Shall		*There are four fire trucks in each difficulty level.*
UR_FORTRESS	There must be at least six fortresses	Product brief	Should		*Each difficulty level has 6 fortresses.*
UR_SCALABILITY	The game should be able to be played on other platforms	Product brief	May		*This requirement has not been met. When playing on different machines the game does not always fit on the screen – for example, on a 14 inch laptop some of the fortresses in the easy mode are not visible and so the game is very difficult to complete.*
UR_PATROLS	There should be at least 2 ET patrols that the user aims to avoid	Product brief	Should		*Each difficulty level has a minimum of two patrols, with this number increasing in the more difficult levels.*
UR_POWER_UPS	***There should be 5 types power ups that the fire trucks can obtain***	***Product brief***	***Shall***	***New requirement due to new change in product brief***	*This requirement is somewhat met. There are 5 different types of power up, however the user has no indication of what each one does and so does not have any way of knowing if they are worth getting.*
UR_DESTROY_STATION	At a point in the game the fire station should be destroyed	Product brief	Should		*The fire station is destroyed at a reasonable point in the game.*
UR_ENJOYABILITY	The game should be enjoyable to play	Product brief	May		*This is a subjective requirement and so difficult to evaluate. As the evaluator at this time, I do not think that this requirement has been met, as the fortresses aren't all visible to me and the minigame is not functioning correctly – it regularly tells me I am wrong when I am pressing the correct arrow or that I am late when the arrow is in the box.*
UR_PLAYABLE	The game must be playable Dependent on environmental	Product brief	Shall		

	assumptions (see bottom of document)				
UR_PLAYER	The game must be a single-player game	Product brief	Shall		*It is a single player game.*
UR_CODE	The game must be coded in Java	Interview	Shall		*It has been coded in java.*
UR_PC	It must be a PC game	Interview	Shall		*The game is playable on PCs.*
UR_TRUCK_SPACE	Fire trucks should not drive over each other or be able to occupy the same space.	Email Communication with Customer	Shall		*It is not possible for fire trucks to drive over each other or occupy the same space. If you try to do so, both trucks in question stop where they are.*

Functional requirements

ID	Description	Source	Priority	User ID	Updates	Evaluation
FR_FIRE_TRUCKS	Each fire truck must have a unique spec in terms of its speed, amount of damage it can take before being destroyed, the volume of water it can carry, the range and delivery rate of its water cannon	Product brief	Shall	UR_FIRE_TRUCKS		*Each fire truck is different in terms of its speed, water reserve, maximum health and water delivery range. These statistics are clearly displayed to the user in the top left corner of the screen when the fire truck has been selected.*
FR_FORTRESS	Each ET fortress must have a unique spec in terms of the range of its defensive weapons, the amount of damage these weapons can deal to Fire trucks over a period of time, and the volume of water it takes to flood	Product brief	Should	UR_FORTRESS		*Each fortress is different. Again, these statistics are clearly displayed to the user in the top left corner of the screen when the fortress in question has been selected.*
FR_WATER	Over time the amount of water needed to flood a	Interview	Should	UR_WIN		*This requirement has not been met in each individual difficulty level, however the amount of water needed to flood the*

	fortress should increase					*fortress in each different difficulty level is different.*
FR_MOBILITY	The user can move the fire trucks. Patrols and fire engines should be mobile; fortresses should be immobile	Product brief	Shall	UR_PLAYABLE		*The user can operate all four fire trucks effectively. The patrols move around without user interaction and the fortresses are immobile.*
FR_AI	The ET patrols and ET fortresses are controlled by the computer AI	Product brief	Shall	UR_PLAYER		*The fortresses and patrols are controlled by the computer AI and the user cannot influence their movement or replenishment of their health.*
FR_TRUCK_ATTACK	Fire trucks can flood ET fortresses	Product brief	Shall	UR_FIRE_TRUCKS		*Fire trucks are able to flood ET fortresses and destroy them.*
FR_FORTRESS_ATTACK	ET fortresses attack trucks	Product brief	Shall	UR_FORTRESS		*Fortresses attack fire trucks when the fire truck is within range of the fortress.*
FR_ACCESS_MINIGAME	The mini game should be accessed from within the main game	Interview	Should	UR_MINI_GAME		*The minigame is accessed when a patrol collides with a truck inside the main game.*
FR_MINIGAME_SAVE	***The progress of the mini game can be saved***	***Email Communication with Customer***	***Should***	***UR_MINI_GAME***	***New functional requirement linking to the new product brief and UR_MINI_GAME***	*The progress of the minigame can be saved in the same way that you would save the progress of the main game. It is not possible to save the minigame and continue with the main game.*
FR_MINIGAME_PAUSE	***The mini game can be paused***	***Email Communication with Customer***	***May***	***UR_MINI_GAME***	***New functional requirement linking to the new product brief and UR_MINI_GAME***	*It is possible to pause the minigame.*
FR_CONTROLS	There should be a screen that explains the controls	Interview	May	UR_PLAYABLE		*There is a screen that explains the controls, however it misses our key features. For example it says nothing about how to attack fortresses (the user is left to figure that out on their own) and the power ups are not mentioned.*

FR_ STATION _DESTROY	Fire trucks cannot be repaired or refilled ***at the fire station*** after the fire station has been destroyed	Product brief	Should	UR_ DESTROY _STATION		*Once the fire station is destroyed, the trucks cannot go back there to repair or refuel.*
FR_MENU	There should be a menu screen from which the user has the option to start the game and ***choose the difficulty level,*** see the controls or quit.	Email Communication with Customer	May	UR_ PLAYABLE		*The menu screen is very clear and allows the user to start the game, choose their difficulty, see the controls or, if they want to, quit the game.*
FR_GAME_ OVER	There should be a 'game over' screen once the game is ended telling the player if the game is won or lost.	Email Communication with Customer	May	UR_ ENJOYABILITY		*There is a game over screen, however it is almost indistinguishable from the winning screen.*
FR_SOUND	There could be sound effects with the game	Email Communication with Customer	May	UR_ ENJOYABILITY		*There are enjoyable sound effects with the game.*
FR_SOUND_OFF	If there are sound effects, it should be possible to turn these off	Email Communication with Customer	Shall	UR_ ENJOYABILITY		*The sound effects can easily be turned off.*
FR_ ANIMATION	The fortresses and fire trucks should change appearance as they are destroyed	Email Communication with Customer	Should	UR_ENJOYABILITY		*The fire trucks do not change their appearance as such, however their health bar goes down. The fortresses visibly change as they are destroyed.*
FR_REPAIR_ REFILL	The fire trucks should repair and refill when at the fire station	Product Brief	Shall	UR_REPAIR		*The fire trucks do repair and refill successfully while at the fire station.*
FR_DIFFICULTY _LEVELS	***The game should implement support for three different difficulty***	***Email Communication with***	***Shall***	***UR_ DIFFICULTY***	***New functional requirement linking to the change in***	*There are three different difficulty levels, easy, medium and hard, all of which are distinct from one another.*

	levels: Easy, Medium and Hard	Customer			product brief and the new user requirement UR_DIFFICULTY	
FR_DIFFICULTY_SELECTION	The player should be able to select the difficulty level before the game starts	Email Communication with Customer	Shall	UR_DIFFICULTY	New functional requirement linking to the change in product brief and the new user requirement UR_DIFFICULTY	The player is easily able to select their difficulty from the main menu screen before the game starts.
FR_POWER_UPS_TIMEOUT	The power ups will timeout after a set amount of time	Email Communication with Customer	Should	UR_POWER_UPS	New functional requirement linking to the change in product brief and the new user requirement UR_POWER_UPS	The power ups are not available at all times and the user only has a short window in which to collect and use them.
FR_POWER_UP_STACK	Power ups should be able to stack	Email Communication with Customer	Should	UR_POWER_UPS	New functional requirement linking to the change in product brief and the new user requirement UR_POWER_UPS	If you are able to collect more than one power up at a time then they stack on top of each other for use. Again it would be good if the user was informed of this in the controls screen.
FR_POWER_UP_OVERLAP	Power ups should not occupy the same space at the same time	Email Communication with Customer	Should	UR_POWER_UPS	New functional requirement linking ot the change in product brief and the new user requirement	Having played the game multiple times, the power ups never appear to occupy the same spac at the same time

					UR_POWER_UPS	
FR_NUMBER_OF_SAVES	The game should support multiple saves	Email Communication with Customer	Should	UR_SAVE	New functional requirement linking to the change in product brief and the new user requirement UR_SAVE	The game is more than able to support multiple saves.
FR_SELECTING_SAVES	The player should be able to select and resume a saved game	Email Communication with Customer	Shall	UR_SAVE	New functional requirement linking to the change in product brief and the new user requirement UR_SAVE	The user is easily able to resume any of their saved games.
FR_DELETING_SAVES	The player may be able to delete a saved game	Email communication with customer	May	UR_SAVE	New functional requirement linking to the change in product brief and the new user requirement UR_SAVE	The user is easily able to delete any of their saved games.
FR_SAVE	The player should be able to save the game at any point	Email Communication with Customer	Shall	UR_SAVE	New functional requirement linking to the change in product brief and the new user requirement UR_SAVE	The user is able to save the game in either the main game or the minigame, even though some more feedback would be appreciated.x1
FR_TRUCK_OVERLAP	Fire trucks should not drive over each other or be	Email Communication with	Shall	UR_FIRE_TRUCKS		It is not possible for fire trucks to drive over each other or occupy the same space. If you try to do so, both trucks in question stop where they are.

	able to occupy the same space.	**Customer**				

Non-functional requirements

ID	**Description**	**Rational**	**Fit criteria**	**User ID**
NF_PC	The game must be playable on engines/things that can be played on PC	Product brief	The game must use libraries /function that could be used on other platforms	UR_ PLAYABLE
NF_ RESPONSE	The game must respond quickly to user input	Improved user experience	Average response time >1 second, maximum response time >2 second	UR_ ENJOYABILITY
NF_ CONTROLS	The controls should be easy to learn	Prospective students should be able to play the game	The player should be able to grasp the controls in under 2 minutes	UR_ ENJOYABILITY

Environmental assumptions: 1) The player is assumed to be playing on a modern computer that is of reasonable specifications. 2) The user will have standard hardware such as a keyboard and mouse. 3) The user will have java installed to run the program.

Risks: The main risks relevant to the requirements are R4 & R7 (see risk assessment table in risk management section).

Use Cases:

Use case 1:

- Name: “FortressDestroy”
- Context: The user destroys a fortress
- Primary Actor: The user
- Stakeholders: Us - with our interests being the user enjoying the game
- Precondition: The system is working as intended

- Minimal Postcondition: The fortress disappears
- Trigger: The fortress health becomes 0
- Main Success Scenario: 1.The user attacks a fortress using a fire truck
    - 2.The fortress health depletes
    - 3.The fortress health becomes 0
    - 4.The fortress disappears
- Secondary scenarios:
    - The fortress's health doesn't deteriorate, leading to the user being unable to destroy it
    - The fire truck does not attack the fortress, leading to the game being unplayable
- Success Postcondition: The user enjoyed attacking the fortress (UR_ENJOYABILITY) and the fortress was successfully destroyed by the user flooding it with a fire truck (FR_TRUCK_ATTACK).

Use case 2:

- Name: "Win"
- Context: The user destroys all fortresses (see Use Case 1) before all of their fire trucks have been destroyed
- Primary Actor: The user
- Stakeholders: Us - with our interests being the user enjoying the game
- Precondition: The system is working as intended
- Minimal Postcondition: The user destroys all the fortresses
- Trigger: The user destroys the last fortress
- Main Success Scenario: 1.The user destroys the last fortress
    - 2.The win state is displayed
    - 3.The game ends
    - 4.Goes back to the game menu
- Secondary scenarios:
    - 2.1 The game continues to run as if the end isn't reached, e.g the
    - fire station timer continues to run so the fire station is destroyed
    - 4.1 The game starts again
- Success Postcondition: The user enjoyed the game (UR_ENJOYABILITY) and did not find it too easy