Requirements

Mozzarella Bytes | Team 18

Assessment N°4

Daniel Benison

Elizabeth Hodges

Kathryn Dale

Ravinder Dosanjh

Callum Marsden

Emilien Bevierre

Requirements *

Requirements added for assessment 4 have been highlighted in bold.

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
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
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
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**
UR_DIFFICULTY	**There should be different levels of difficulty in the game**	**Product brief**	**Shall**
UR_MINI_GAME	The game should include a minigame	Product brief	Should
UR_MINI_ GAME_THEME	The minigame should be different in style, but aligned to the theme of the main game	Product brief	Should
UR_REPAIR	Fire trucks can be repaired and refilled at the fire station	Product brief	Should
UR_FIRE_TRUCKS	There must be at least four fire trucks	Product brief	Shall

UR_FORTRESS	There must be at least six fortresses	Product brief	Should
UR_ SCALABILITY	The game should be able to be played on other platforms	Product brief	May
UR_PATROLS	There should be at least 2 ET patrols that the user aims to avoid	Product brief	Should
UR_POWER_UPS	**There should be 5 types power ups that the fire trucks can obtain**	**Product brief**	**Shall**
UR_DESTROY _STATION	At a point in the game the fire station should be destroyed	Product brief	Should
UR_ENJOYABILITY	The game should be enjoyable to play	Product brief	May
UR_PLAYABLE	The game must be playable Dependent on environmental assumptions (see bottom of document)	Product brief	Shall
UR_PLAYER	The game must be a single-player game	Product brief	Shall
UR_CODE	The game must be coded in Java	Interview	Shall
UR_PC	It must be a PC game	Interview	Shall

Functional requirements

ID	Description	Source	Priority	User ID
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
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
FR_WATER	Over time the amount of water needed to flood a fortress should increase	Interview	Should	UR_WIN

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
FR_AI	The ET patrols and ET fortresses are controlled by the computer AI	Product brief	Shall	UR_ PLAYER
FR_TRUCK_ ATTACK	Fire trucks can flood ET fortresses	Product brief	Shall	UR_FIRE_ TRUCKS
FR_FORTRESS_ ATTACK	ET fortresses attack trucks	Product brief	Shall	UR_ FORTRESS
FR_ACCESS _MINIGAME	The mini game should be accessed from within the main game	Interview	Should	UR_MINI _GAME
FR_MINIGAME _SAVE	*The progress of the mini game can be saved*	*Email Communicatio n with Customer*	*Should*	*UR_MINI_ GAME*
FR_MINIGAME _PAUSE	*The mini game can be paused*	*Email Communicatio n with Customer*	*May*	*UR_MINI_ GAME*
FR_ CONTROLS	There should be a screen that explains the controls	Interview	May	UR_ PLAYABLE
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
FR_MENU	There should be a menu screen from which the user has the option to start the game and **choose the difficulty level, view their saves,** see the controls or quit.	Email Communication with Customer	May	UR_ PLAYABLE
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_ ENJOYABILIT Y

FR_SOUND	There could be sound effects with the game	Email Communication with Customer	May	UR_ ENJOYABILITY
FR_SOUND_OFF	If there are sound effects, it should be possible to turn these off	Email Communication with Customer	Shall	UR_ ENJOYABILITY
FR_ ANIMATION	The fortresses and fire trucks should change appearance as they are destroyed	Email Communication with Customer	Should	UR_ ENJOYABILITY
FR_REPAIR_ REFILL	The fire trucks should repair and refill when at the fire station	Product Brief	Shall	UR_REPAIR
FR_DIFFICULTY_ LEVELS	*The game should implement support for three different difficulty levels: Easy, Medium and Hard*	*Email Communication with Customer*	*Shall*	*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*
FR_POWER_ UPS_TIMEOUT	*The power ups will timeout after a set amount of time*	*Email Communication with Customer*	*Should*	*UR_POWER_ UPS*
FR_POWER_UP _STACK	*Power ups should be able to stack*	*Email Communication with Customer*	*Should*	*UR_POWER_ UPS*
FR_POWER_UP _OVERLAP	*Power ups should not occupy the same space at the same time*	*Email Communication with Customer*	*Should*	*UR_POWER_ UPS*
FR_NUMBER _OF_SAVES	*The game should support multiple saves*	*Email Communication with Customer*	*Should*	*UR_SAVE*
FR_SELECTING _SAVES	*The player should be able to select and resume a saved game*	*Email Communication with Customer*	*Shall*	*UR_SAVE*

FR_DELETING _SAVES	*The player may be able to delete a saved game*	*Email communicatio n with customer*	*May*	*UR_SAVE*
FR_SAVE	*The player should be able to save the game at any point*	*Email Communicatio n with Customer*	*Shall*	*UR_SAVE*
FR_TRUCK_ OVERLAP	**Fire trucks should not drive over each other or be able to occupy the same space.**	**Email Communicatio n with Customer**	*Shall*	*UR_FIRE_ TRUCKS*

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