Requirements

Cohort 4 Group 6

Javengers

Braithwaite, Max
Faruque, Amber
Fu, Zhuoran
Kocaman, Melike
McDermott, John
Rissen, James
Scott, Charlotte

As a group, we researched different methods of coming up with questions in the process of eliciting user requirements. The approach we decided to follow was brainstorming as a team, with each team member contributing and giving their opinions. Brainstorming encourages creativity from every team member, as well as being a very swift and efficient process. The product of the brainstorming was then condensed and refined by a couple of team members while other members were completing tasks succinctly. This set of questions was then reviewed by the other members of the team, with each question being discussed and reasoned as to its importance. We aimed to ask as many questions as possible during the interview to maximise the amount of information conveyed to us by the customer.

Our research was conducted with our client in an interview. Our approach to the interview was to have one team member primarily asking questions, whilst another team member was noting down key fundamental requirements. We also voice recorded the interview to ensure we didn't miss any key information. Our client initially covered the primary details that they wanted for the game and then we followed up with some additional functional and non-functional questions. Post-interview, our team organised a meeting to discuss the user requirements which we had gathered from the client, ensuring that everyone was familiar and comfortable with the plan for the project.

In order to document our requirements post-interview we broke the requirements down into three tables, user requirements, functional requirements and non-functional requirements. This was beneficial as splitting the requirements into categories ensures that each type of requirement is clearly defined and easy to locate while reducing confusion in regards to task allocation. This allows our team members to quickly scan the tables and prioritise requirements. Additionally, by including priorities in the table we can make sure that nothing vital is missed in the implementation. We decided to use IDs to allow the requirements to be easily referenced between documents and in project tracking. Finally this separation improves the ease of testing as test cases can be mapped directly to requirements.

ID	Description	Priority
UR_DISPLAY	The game should display to the user over a screen of varying sizes and should be accommodating for visual impairments.	Shall
UR_MAP	The game should have a university maze style map which the player must navigate to escape - this could resemble the UOY campus to appeal to stakeholders.	Shall
UR_PLAYER_CONTROL	The User should be able to control the player using an input system (Mouse and keyboard).	Shall
UR_GAME_COMPLETION	The User should be able to complete the game - winning or losing.	Shall
UR_PLAYER_ABILITIES	The User should be able to interact with the surroundings in specific ways that allow them to overcome a challenge .	Should
UR_CHALLENGE	There should be interactions in the game that challenge the user, preventing them from completing the game too easily.	Should
UR_SYSTEM	The system should work well on multiple devices	Should
UR_STYLE	The game should be a top down 2D style maze game, themed around escaping from university - user group ought to be considered.	Shall
UR_TIME_LIMIT	The game should have a limited duration.	Shall
UR_MENUS	The User should have access to several menu screens within the game that allow them to access key changeable variables - they must be easy to navigate.	Shall
UR_UI	There should be a pleasant and understandable UI.	Should
UR_ENEMY	The user can be slowed down by enemies that patrol the map.	May
UR_POINTS	The user can collect and earn points that increase their overall score.	Should
UR_LEADERBOARD	The user should be able to view their past scores via a leaderboard table accessible through the main menu - this should be clear and easy to access.	Should

ID	Description	User Requirements
FR_RESIZE	The window will resize to fit the screen size of the Users device.	UR_DISPLAY
FR_COLLISION	The player character will collide with the sections of the map - limiting their movement.	UR_MAP
FR_MAP_LIMIT	The map will have an “end” which the play cannot move over.	UR_MAP
FR_KEYBOARD_CONTROL	The player must be able to move up, down, left and right with the keys WASD, and the player character must alter its sprite in situ with this movement.	UR_PLAYER CONTROL
FR_MOUSE_CONTROL	The player must be able to use the buttons on their mouse to interact with the game and it must implement some functionality in the design.	UR_PLAYER_CONTROL
FR_WIN_LOSE	The User must be able to find a way to complete the game within the constraints of the game.	UR_GAME_COMPLETION
FR_WIN_LOSE_SCREEN	Upon the user completing the game, or the player running out of time, the game should display a screen or image to inform the user of their success or failure.	UR_GAME_COMPLETION
FR_TORCH	The user must use a torch mechanic to light their way in the game in order to escape the maze. There must be a way for the player to interact with the torch via an input device.	UR_PLAYER_ABILITIES
FR_KEYCARD	The user must be able to pick up, hold, drop and use a keycard to access new areas of the map.	UR_PLAYER_ABILITIES

FR_FOOD	The User should be able to collect and consume a food item that temporarily boosts certain variables associated with the player character.	UR_PLAYER_ABILITIES
FR_GOOSE	The player Should be hindered by a goose - per the game being set on UOY campus. The Goose should be able to “steal” an item from the player and hinder them from completing the game.	UR_CHALLENGE
FR_LAKESIDE_EVENT	The player should be able to interact with the area around the lakeside, including the buildings. Completing tasks in each	UR_CHALLENGE
FR_MAZE_EVENT	The player should be able to complete small “side tasks" in the maze in order to earn extra points or increase their “time remaining” during the gameplay - these need not be completed in order to complete the game.	UR_CHALLENGE
FR_HIDDEN_GOOSE_EVENT	The play should be able to interact with the goose in some hidden way which does not hinder or benefit the player.	UR_CHALLENGE
FR_LECTURER	The player should be chased by a lecturer and forced into an UNSKIPPABLE cutscene - the timer should continue to run during this - worsening the player's score.	UR_CHALLENGE
FR_VISUALS	The game should resemble a university, but not with realism, a cartoonish style that is appealing to the general clientele, possibly a pixellated style that is appealing to younger users.	UR_STYLE
FR_COUNTDOWN	There should be a timer that limits the length of the game, this should be visible at all times when a game is in effect, it should limit the game to last 5 minutes.	UR_TIME_LIMIT
FR_PAUSE_COUNTDOWN	When the game is paused the countdown should be paused as well.	UR_TIME_LIMIT
FR_MAIN_MENU	There should be a main menu when the game is first opened which allows a player to access all other menus of the game, as well as a way to initiate a sequence of the game - a “PLAY” button.	UR_MENUS
FR_SETTINGS	There should be a settings menu where the user can alter key variables within the game - such as game volume, music, player sounds. There should also be a return button to the main menu.	UR_MENUS
FR_PAUSE	When the player pauses the game the pause menu should appear. This should stop the timer countdown, as well as offer the player the option to return to the main menu, as well as the option to return to the game.	UR_MENUS
FR_RETURN	The player should be able to return to the previous menu from the menu they’re in now.	UR_MENUS
FR_LOCAL_SCORE	There should be a menu where the user can see a set of saved game scores - only the top ten should be shown along with the details relevant to them.	UR_MENUS
FR_SCORE	There should be a constantly displayed score which is based on the player's timer, as well as interactions that a player completes within the game.	UR_UI
FR_HOTBAR	The player should be able to collect items and store them in an inventory style hotbar. The hotbar should be visible at the top of the screen - could have a size limit	UR_UI
FR_ACTIVE_CHALLENGE	There should be a block of text that describes to the player what they should be trying to do - so the player does not get confused and lose interest.	UR_UI
FR_CLOCK	Part of the UI should simply show the timer countdown.	UR_UI

FR_E_PATROL	Enemies that patrol the map should be able to hinder or slow down the player without a specific cutscene or interaction.	UR_ENEMY
FR_BONUS	There should be a collectable item that gives you some bonus points towards your final score that is collectable throughout the game.	UR_POINTS
FR_INTERACTION_BONUS	Upon completing specific interactions the player should gain a specific set of bonus points depending on how quickly they completed the interaction or task.	UR_POINTS

ID	Description	User ID	Fit Criteria
NFR_NAVIGATION	The map should be easily traversed.	UR_MAP	It should not take the player more than 2.5 minutes to cross the entire map.
NFR_MAP_LOAD	The time to load from one “map” in the game main maze to side rooms etc should be quick.	UR_MAP	It should not take more than a second to load in and out from maps.
NFR_PLAYER_RESPONSE	The player's character should respond quickly to user inputs.	UR_PLAYER_CONTROL	The player's character should respond to an input in <1ms.
NFR_INTUITIVE	The game should be easy for someone new to learn.	UR_PLAYER_CONTROL	The game controls should be easy for someone to learn within the first 2 minutes of gameplay
NFR_FEEDBACK	The game should provide feedback is the player completes an action	UR_PLAYER_CONTROL	And audible feedback shall inform the player they have completed an action
NFR_GAME_END	The game should finish within a set time.	UR_GAME_COMPLETION	The game should last < 5mins.
NFR_MIN_TIME	The game should not be completable under a certain time.	UR_GAME_COMPLETION	The game should last > 3mins.
NFR_MENU_USE	Menus should be usable	UR_MENUS	The font on the menus should be readable.
NFR_MENU_TO_GAME	The switch from the menu screen to the actual game should be quick.	UR_MENUS	Should move from menus to game < 3 seconds.
NFR_INTER-MENU	Moving from menu to menu should not be slow, allowing the player to navigate easily.	UR_MENUS	From one menu to another should take < 1 second.
NFR_PAUSE_MENU	Once in the game the player can pause the game - but should be able to do this quickly	UR_MENUS	The switch from game to pause menu should take less than a second.
NFR_GAMESPEED	The game should run smoothly at all points	UR_DISPLAY	The game should run at >60fps 90% of the time
NFR_CONSISTENT_GAMESPEED	The game should run consistently even if there are many objects	UR_DISPALY	The game should run at >30 fps at all times
NFR_GAMEPLAY	The game shall not crash or freeze during gameplay	UR_DSIPLAY	The game will only crash or freeze a maximum of 1 in every hundred plays of the game
NFR_ACCESSIBILITY	The game should have options for those who are colourblind	UR_DISPLAY	Deuteranomoly, Protanomaly filters should be added at a minimum.
NFR_OS_VERAIONS	The game should run on multiple operating systems	UR_SYSTEM	The game should run on Windows, Linux and MAC OS
NFR_GAME_SIZE	The game shall not be too large to inhibit gameplay	UR_SYSTEM	The game shouldn’t exceed 500MB during gameplay