Requirements

Team 16

CatepillaDevelopment

Yousif Al-Rufaye
Joe Fuller
William Gracie-Langrick
Jack Hardy
Bailey Uniacke
Ben Young

a)

Our user and system requirements were first drafted when we initially viewed the brief. We collaborated and discussed our initial interpretation of what we thought the customer would require from the project brief, we attempted to abstract the brief and wrote up questions that we had for our customer interview. We then arranged an interview with the customer who provided the brief to answer any missing gaps or ambiguities that we thought that were in it. This interview also assisted us identifying the target audience and any other stakeholders, thus having a key role in deciding how the user should interact with our system. After that, we negotiated and finalised our user and system requirements.

Our user requirements are formatted in a table such that each user requirement has an ID, a description and a priority. We discussed that an ID was necessary so we can uniquely identify our requirements, making it easy to search. A description was also necessary, as no description of the ID may cause some ambiguity and confusion as to what the user requirement actually is. Finally, the priority column is on a scale of importance of not only the task, but whether or not they have to be added in the scenario time is short. This scale is in the order of “May”, “Should” and “Must”. “Must” being the priority of most importance and “May” being the least important.

Our system requirements are split into two tables, functional and non-functional requirements. Functional requirements are the requirements of the system given the following events: transformations, invariants and failures. Our table again consists of three columns: ID, description and user requirement. The ID and description have the same function as stated above. The user requirement column however, describes which user requirement this functional requirement would cover. Our non-functional requirements table has 4 columns: ID, description, user requirement (definition stated above) and a fit criteria. The fit criteria is the quantification of the non-functional requirement.

b)

User Requirements

ID	Description	Priority
UR_COOK_CONTROL	The controls should be easy to use without a tutorial.	Should
UR_COOKING	Cooking should be done with a timer at the stations	Should
UR_SWITCHING_COOKS	Users must be able to switch between cooks	Must
UR_SOUND_TEXT	Should not rely on sound and text	Should
UR_COOK_INTERACTION	May click for the cook to interact with the environment	May
UR_GAME_LOOP	Game may loop when no one is playing to show people how the game works	May
UR_SIMPLICITY	Watching someone else play the game should easily show others how to play	Should
UR_DIFFICULTY_SLIDER	Should be a way to choose difficulty	Should
UR_LEVEL_GENERATION	Should have one fixed level	Should
UR_STAFF	All staff must be cooks and can do any task	Must
UR_RECIPES	Recipes should stay the same	Should
UR_GAME_TIME	Games should be fairly quick (no more than 6 - 8 minutes).	Should
UR_CONTROLS	Must work with keyboard and mouse.	Must
UR_DISPLAY	Should work on a 32 inch screen and must work on a 13 inch laptop.	Should

UR_TIMER	There must be a timer displaying how long it took to complete a level.	Must
LIMITED_TIME	A level may be limited by the timer	May

Functional Requirements

ID	Description	User Requirements
FR_CONTROL_IMPLEMENTATION	Can use point & click, direct control, or drag and drop (whichever is simplest for the user)	UR_COOK_CONTROL
FR_TIMER_POPUP	When food is being cooked, a timer should pop up to show how long is left	UR_COOKING
FR_SELECTING_COOK	Cook can be selected by clicking on it	UR_SWITCHING_COOKS
FR_DISTINCT_COOKS	Cooks should look different but don't rely on text, sound, or colours	UR_SOUND_TEXT
FR_DIFFICULTY_SELECTION	Up to us to choose how to implement. Could be through the main menu.	UR_DIFFICULTY_SLIDER
FR_PLACE_ITEMS	The chef will be able to place their items at stations around the kitchen.	UR_COOK_INTERACTION
FR_PICKUP_ITEMS	The chef will be able to pick up items	UR_COOK_INTERACTION
FR_TIME	Players will be shown their time at the end of the level.	UR_TIMER
FR_PREPARING_FOOD	The chef can prepare food at the stations	UR_COOKING
FR_BIN_ITEM	The chef can use the bin to remove items from their inventory	UR_COOK_INTERACTION

Non-Functional Requirements

ID	Description	User Requirements	Fit Criteria
NFR_CONTROL_ACCURACY	Chef should move to where the player points (in the kitchen)	UR_CHEF_CONTROL	Controls are 99% responsive
NFR_SELECTION_ACCURACY	The correct cook should be selected	UR_SWITCHING_COOKS	Must switch the the chosen cook 100% of the time
NFR_DISTINCT_ENTITIES	All entities (chefs or food items) should be easily distinct from each other without relying on colour/text	UR_SOUND_TEXT	Chefs must have slightly different models (one could have a unique pattern on their apron)
NFR_DISPLAY_SIZE	The game should be adjustable to fit the size of the display	UR_DISPLAY	The game must fit on a 13 inch laptop screen. Should work on a 32 in screen but we may not get access to that.
NFR_TIMER_ACCURACY	The timer should be accurate so that events can sync properly	UR_TIMER	Should be in seconds (with a few decimal places to display milliseconds).
NFR_LOOP	The game should loop when no one is playing to show them what the game is about and how they play.	UR_GAME_LOOP	Only need to show 1 chef doing activities.