Change Report

Assessment 2 Team

Team 12

Richard Liiv

Umer Fakher

William Walton

James Frost

Olly Wortley

Joe Cambridge

a) Briefly summarise the team’s formal approaches to change management, including change management of all deliverables, documentation and code.

We forked Team 15’s GitHub page and started making changes to their code. They had leftover branches from the last assessment. We sorted through these by removing ones we didn’t deem necessary and created new branches based on the new requirements that were then assigned to members of our team. We naturally extended the previous groups source control systems.

Shortly thereafter, we requested editable copies of Team 15’s deliverables from Assessment 1 via email. We established a working relationship with them in order to clarify queries. Much like we had our Assessment 1 deliverables in Google Drive, we imported the copies of Team 15’s documents there as well.

We created a github site (https://umerfakher.github.io/ENG1Project/) which clones the previous group’s site and adds our adjusted deliverables and assessment 2 reports. It is linked to our github repository (https://github.com/UmerFakher/ENG1Project) for code forked from Team 15.

For some of the original team’s documents we highlighted parts that we made changes to which would help the reader distinguish what is original and what are the changes. An example of this is the edited Requirements document where strike-outs and bold has been used. This was not appropriate in other documents with extremely tight limit restrictions. Please see the rest of the Change document for a summary of these.

The previous team had two people working on the implementation of code, much like we did in the first assessment. We had two people from our group take over their (the coders’) role and work on their code base. The rest of our group would fully focus on updating the other team’s deliverables as well as work on documents exclusive to Assessment 2.

After the programming part of Assessment 2 was completed, the remaining team members helped out with documentation as well.

b) i) Requirements

https://github.com/UmerFakher/ENG1Project/blob/website/Documentation/Req1.pdf

The requirements for assessment 1 for Team 15 were mostly not changed as they apply to assessment 2. However, changes were made to the requirements tables to include the additional assessment 2 requirements. There were 3 new requirements for assessment 2: levels of difficulty, saving and resuming the game and implementing 5 powerups to boost attributes of the boat.

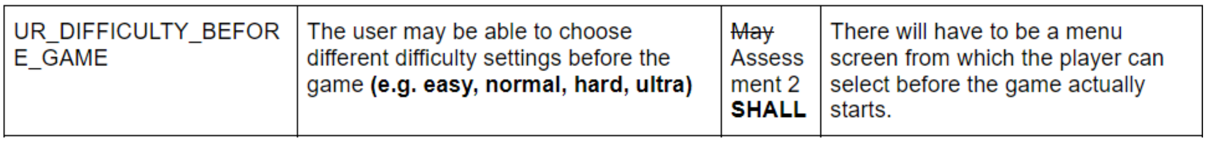
Team 15 had 4 requirements listed in assessment 1's not fully implemented section:

- UR_DIFFICULTY_BEFORE_GAME,
- UR_POWERUPS,
- FR_POWERUPS_RATE,
- NFR_ATTRIBUTES

One not mentioned was FR_DIFFICULTY_SELECTION although we can imply that this is not completed from its parent UR_DIFFICULTY_BEFORE_GAME.

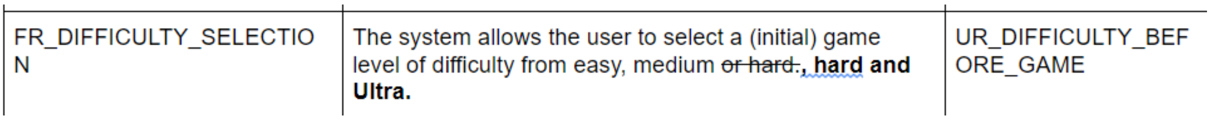
They were very similar to the assessment 2 requirements so we combined them with the new assessment 2 requirements. See the following:

Difficulty



UR_DIFFICULTY_BEFORE_GAME	The user may be able to choose different difficulty settings before the game **(e.g. easy, normal, hard, ultra)**	~~May~~ Assessment 2 **SHALL**	There will have to be a menu screen from which the player can select before the game actually starts.

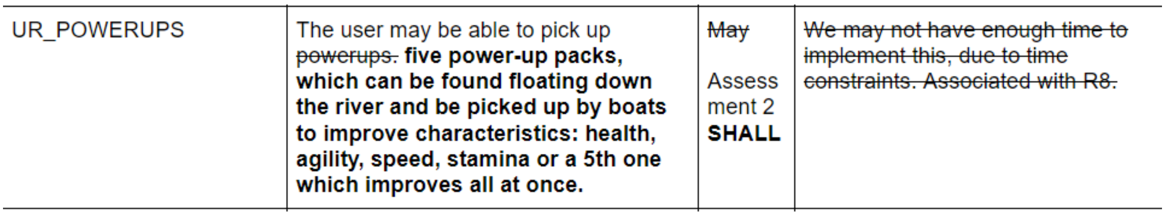
- UR_DIFFICULTY_BEFORE_GAME (User Requirement)
  - Changed description to clearly define difficulties (easy, medium, hard & ultra) inline with the new requirements for assessment 2.
  - Increased priority to shall for assessment 2



FR_DIFFICULTY_SELECTION	The system allows the user to select a (initial) game level of difficulty from easy, medium ~~or hard.~~**, hard and Ultra.**	UR_DIFFICULTY_BEFORE_GAME

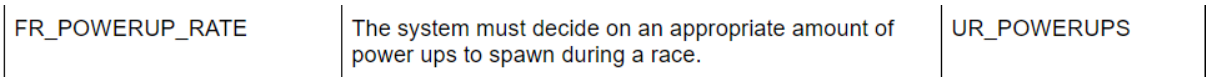
- FR_DIFFICULTY_SELECTION (Functional Requirement)
  - Changed description to clearly define difficulties (easy, medium, hard & ultra) inline with the new requirements for assessment 2.
  - Increased priority to shall for assessment 2

Power Ups



UR_POWERUPS	The user may be able to pick up ~~powerups.~~ **five power-up packs, which can be found floating down the river and be picked up by boats to improve characteristics: health, agility, speed, stamina or a 5th one which improves all at once.**	~~May~~ Assess ment 2 **SHALL**	~~We may not have enough time to implement this, due to time constraints. Associated with R8.~~

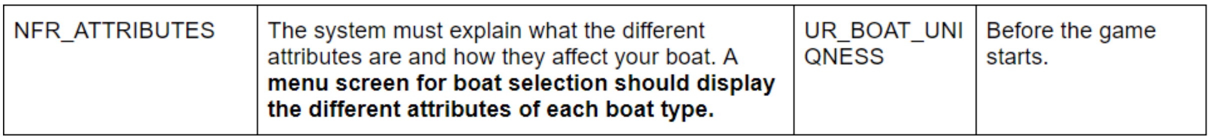
- UR_POWERUPS Changed (User Requirement)
  - Changed description to clearly define details of the power up packs that we have been assigned inline with the new requirements for assessment 2.
  - Increased priority to shall for assessment 2
  - Removed notes



FR_POWERUP_RATE	The system must decide on an appropriate amount of power ups to spawn during a race.	UR_POWERUPS

- FR_POWERUP_RATE (Functional Requirement)
  - This requirement is linked to the previous one and required no change even though the description of UR_POWERUPS was edited. It is relevant to assessment 2 so it is included for context for UR_POWERUPS.
  - It was also not implemented in assessment 1.

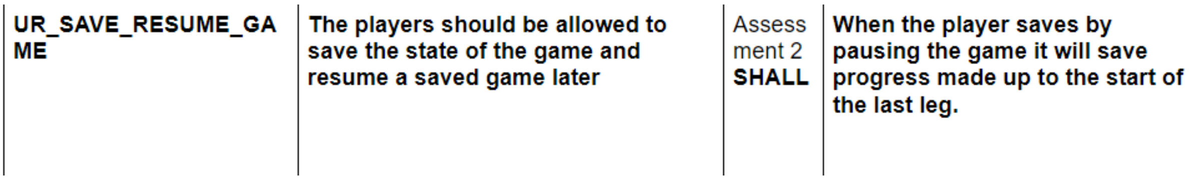
Attributes



NFR_ATTRIBUTES	The system must explain what the different attributes are and how they affect your boat. A **menu screen for boat selection should display the different attributes of each boat type.**	UR_BOAT_UNIQNESS	Before the game starts.

- NFR_ATTRIBUTES
  - This was an incomplete requirement from assessment 1 so will be followed through in assessment 2.
  - Changed description to clearly define the boat selection screen that will respect the NFR_ATTRIBUTES section to display the attributes of each boat type to the user.

Save & Resume Game



UR_SAVE_RESUME_GAME	**The players should be allowed to save the state of the game and resume a saved game later**	Assess ment 2 **SHALL**	**When the player saves by pausing the game it will save progress made up to the start of the last leg.**

- UR_SAVE_RESUME_GAME
  - A new requirement about saving the game state was added, including the description.
  - Priority was increased to shall and notes were added to be in line with assessment 2.

Other

FR_GAME_DURATION: “The system runs from between 3 to 5 minutes”
was changed to
NFR_GAME_DURATION linked to UR_GAME_LENGTH:

Description:

- The game is a short game that runs for a reasonably short length of time and has a short time to complete.

Fit criteria:

- “the game leg can be completed by the user in typically around 30 seconds and the whole game may be completed in around 3/5 minutes”.

See requirements for more details.

As the game duration was not respected by the previous team and would not make sense to strictly close the game within this time range potentially causing user frustration if they haven’t finished. This is probably not what the original team intended here, they probably mean to set this as a typical time basically to say the game is a short game not a medium or long time investing game to complete like the games of today. It is also more fitting to call this a non-functional requirement.

ii) Abstract and concrete architecture

https://github.com/UmerFakher/ENG1Project/blob/website/Documentation/Arch1.pdf

Preface removed to make room for changes to architecture for new requirements.

The architecture UML diagrams seemed to include inheritance relations only and wasn't following typical UML notation fully. As a result, we made some changes and added in more detailed and accurate diagrams. These show packages in our project also by grouping classes to make it a lot easier to understand the project architecture.

The architecture justification part has a lot of explanation as to why and what key design decisions were made and as it is important we respect these decisions to the best of our ability. Some paragraphs were removed due to tight page restrictions so depth of explanations was limited for assessment design decisions.

New justification for the new requirements for assessment 2 was added and this briefly explains the new design decisions made for this second phase of the project. We mention that we build upon the existing framework to implement these new/changed features. We add and reference the relevant requirements that link to this architecture.

For example we build upon existing functionality by making sure that obstacles are distributed according to difficulty (UR_DIFFICULTY_BEFORE_GAME) selected by the user from the difficulty selection screen (FR_DIFFICULTY_SELECTION) and these originally just 'negative' obstacles also coexist with other 'positive' obstacles such as power ups which use the same ObstacleType implementation for scalability in lane class. There are more details in the edited Architecture document as well as Implementation 2 document. Changes to the main game class are made like a difficulty attribute and ability to retrieve and set this value from external classes due to added class protection by our group that the other group did not consider. We added protection to potentially reduce unwanted side effects of classes accessing/changing the wrong attributes or accessing wrong functionality that they should not be doing.

This links to FR_POWERUP_RATE where the system must distribute an appropriate amount of power ups to spawn during a race. We make sure that the 3 'negative' obstacles (originally added obstacles) are distributed and spawned more often in order to have appropriate frequency of obstacles as per the difficulty selected by the user by FR_DIFFICULTY_SELECTION but also so that for example 'positive' obstacles (power ups) don't clutter the game inappropriately and provide an appropriate challenge for the user and make the game playable and respect other existing user-experience and gameplay based requirements.

We add a DifficultySelectionScreen for FR_DIFFICULTY_SELECTION implementing the libgdx Screen interface to let the user select the difficulty level of the game. We again make sure that this respects the user-experience and gameplay based requirements which does actually have a noticeable effect on the difficulty level as per UR_DIFFICULTY_BEFORE_GAME. More difficulty levels should be easily added and existing ones should be editable so that is why we made sure that these were displayed by the DifficultSelectionScreen.

More specifically for UR_POWERUPS we add the new ObstacleType options for new power-ups which required Enums for health, stamina, agility, and speed and ‘all’ attribute powerup and more attributes for these obstacles to allow for these ‘positive’ effects will all be added in ObstacleType class ensuring we respect scalability. These changes were made along with respecting the FR_POWERUP_RATE as mentioned above to make as discussed above to meet this requirement. This does allow for future powerups to be added by simply following these steps and the effect/attributes of existing power ups to be changed in the future for tuning purposes.

Along with the adding DifficultySelectionScreen we made further screen changes by editing BoatSelectionScreen to fully print the values on the display for each attribute for each of the different BoatTypes on BoatSelectionScreen. This was to implement the incomplete requirement NFR_ATTRIBUTES as mentioned from before. This was again respecting scalability of the existing screen and attributes for each BoatType implemented. In case there are more attributes or more BoatTypes that are added in the future this screen would be able to handle dynamically displaying these attributes values for each BoatType.

Finally due to the UR_SAVE_RESUME_GAME requirement we allow the player to pause the game, save the game progress up to the start of the last leg started and resume this game later. User input (e.g. using a button like ‘ESC’) for pausing the game and save in game state (storing the player’s total time, current round and difficulty selected as per UR_DIFFICULTY_BEFORE_GAME) in a text file. The user input button is changeable. This is implemented through the RoundScreen class and saving to file functionality will be implemented in this file. The MainGameScreen will have a new button to load from such a ‘save game’ text file inorder to resume gameplay with the correct round, time and difficulty for the game. As a result we made these additions so that in the future if there are further attributes for a game that are added and need to be saved, then this is possible in the save file.

iii) Methods and plans: software development methods and tools; team management approaches; plan for Assessment 2

https://github.com/UmerFakher/ENG1Project/blob/website/Documentation/Plan1.pdf
See Sprint Updates for assessment 1:
https://umerfakher.github.io/ENG1Project/#section1
For assessment 2 see under Sprint Updates heading:
https://umerfakher.github.io/ENG1Project/#section2
For Project Plan part C see here:
https://umerfakher.github.io/ENG1Project/Documentation/Plan1.pdf#page=5

Since our team and Team 15 mostly used similar mediums for communication and similar programs, libraries and resources to utilise during the project, not a lot of change was needed in the first part of the document.

That being said, there was a small exception to that regarding task management. In the original document, no means of task organisation tools were written about, so we included our team's way of managing tasks by means of Trello boards. We included screenshots of our Trello board and the calendar at the end of the planning document.

Because the usage of tools for UML diagrams wasn't mentioned in the original document, we also added a small paragraph about that.

We edited the paragraph about workload management to reflect our team's methods; for the most part, our workload management style was the same as Team 15's, but over the course of the project we adapted to a different style to use our team's resources better.

Another small change was in explaining the way how the documentation team worked, where the entirety of Team 15 worked together on a single section of the document in a specific timeframe, whereas our team equally divided tasks between members of the group where each person focussed on one or two sections.

A paragraph was added elaborating on the role of the SCRUM master and what we did to follow the SCRUM development process. Please see link at top for details in part B.

As mentioned in part a) of this change report, a change made to methods and planning document was using the same team organization, just taking over the roles and re-assigning responsibilities to our group members.

The Plan Breakdown was removed from the document due to information redundancy and due to lack of space needed for our changes. Below that, the second half of the Gantt chart as well as a critical path and a short description of the diagram was added alongside the original one.

Under the Project Plan part of the document, we also removed specific details about Assessment 1's Gantt diagram to make room for some images.

iv) Risk assessment and mitigation: approach, presentation, risks, mitigations

https://github.com/UmerFakher/ENG1Project/blob/website/Documentation/Risk1.pdf

Removed the preface to increase space.

Changes were made to the “Owner column” of the risk table (every member of our team took over someone else’s risk to oversee), and we added a few risks related to us taking over the project. Since the core of the project remained the same, we didn’t think it necessary to change most of the risks that were already there.

For risk R14 (grouped as a people risk by the previous team) the previous team justified a mitigation strategy of asking for a deadline extension if the coding skill level of everybody in the group was not high enough. We decided to remove this part of the mitigation as we felt it was unrealistic that a deadline extension would be granted and it therefore couldn’t reliably mitigate the risk.

We added an explanation for the type column which grouped the risks into people, project, product and technology as this wasn’t included in the reasoning behind risks methodology by the previous team. We also added an explanation for why this sort of grouping would be appropriate. We continue and follow this structured approach when adding additional risks for the project.