Change Report

Group Number: Cohort 2 Team 7

Group Name: pickNmix

Group Member Names:

David Lun

Sameer Minhas

Harry Muir

Phyo Lin

Alex McBride

To effectively plan, manage and review changes to the Assessment 1 deliverables, documentations, and code inherited from the previous team, our team used a systematic and cooperative approach, utilizing thorough procedures, reliable tools, and well-defined conventions. The process began with an in-depth review of the inherited deliverables, including requirements, architecture, risk assessments, method selection and planning documents. Firstly, we reviewed all documents with the purpose of finding any inconsistencies, wrong information and areas where the additional clarity or alignment with project goals are needed.

Based on the things that we learned from the review, we updated the requirements document to reflect the game that includes the new feature requirements without any ambiguities. The architecture document is also changed with some new changes in order to make sure that it aligns with the updated requirements. The risk assessment was revisited to account for newly identified risks and mitigation strategies, with additional detail provided for critical areas. Moreover, the method selection and planning is also updated to reflect the actual our team planning strategy.

We organized the documentation for these improvements using tool like Google Docs, which allowed for real-time collaboration, and simple access for any team member. For code, we used Git to track changes, manage branches and and facilitate the reviews. Automated CI is also implemented in order to make sure the changes to the codebase were rigorously tested before the integration. For the communication between the team members, our team mainly used discord for the collaboration as it makes it easier to communicate for the changes that we made and the things that we need to discuss about the project between the team members.

To maintain quality and consistency, we followed standardized conventions, including structured naming patterns, descriptive commit messages, and code formatting guidelines. Then, we also conducted peer reviews for every update in both documentation and code in order to make sure that those changes were followed by the project objectives. Additionally, the team was able to assess progress, resolve issues, and improve procedures through regular review meetings held at the end of each phase. These steps ensured the deliverables and documentation were not only updated but also enhanced, setting a strong foundation for the next phases of the project.

Requirements:

Brief explanation of overall requirements change:

The requirements deliverable has been changed to better reflect the new product brief. This updated product brief includes two additional objectives: One is that the game must contain a **leaderboard** with the name and score of the top 5 scores and the second objective is that within 'winning' the game through achieving the best student satisfaction, **achievements** can change the final score after the game ends. In order to apply the above objectives the requirements deliverable needed to be updated and include adding to both the User Requirements table and the Functional Requirements table. The Non-Functional Requirements table was not necessary to add to as there were no additional qualities the system must have to best represent itself

Also updated the requirements deliverable in other ways to make it clearer, easier to read and better align with the Group Assessment Document such as changing the introduction to include why the requirements are presented the way they are and formatting the tables themselves

Added Requirements

The **UR_LEADERBOARD** requirement "The player shall be given a leaderboard with the names and score of the top 5 scores at the end and start of the game". It was necessary to include as it was a part of the "Your game must contain:" section of the product brief. This is a user requirement since the user is affected by this leaderboard as they will be able to see it at the end of the game and so this requirement should meet what the user wants

The **FR_LEADERBOARD** requirement represents the actions the system needs to implement in order to meet the user requirement **UR_LEADERBOARD.** We decided as a group to make the leaderboard into a table as that was the most logical option. Initially we decided to only have 2 columns: username and score and to deal with achievements in a different manner. However as achievements were affecting the student satisfaction i.e the score, we thought it would be appropriate to include achievements into the leaderboard table.

The **UR_ACHIEVEMENTS** requirement "The player shall be able to obtain achievements during their playthrough of the game". This was necessary to include as it was a part of the "'win' the game" section of the product brief and the user is affected by achievements as a visual element and it changes their final score.

The **FR_ACHIEVEMENTS** requirement represents the actions the system needs to implement in order to meet the user requirement **UR_ACHIEVEMENTS**. In order to implement achievements we decided to award achievements while the game is being played so that more interesting things can happen on screen and give the user a rush and an indication that their actions matter. We decided to only show the total number of achievements at the endscreen of the game to purposefully hide information from the user in

hopes of increasing the game's replayability so that the player will want to collect more achievements the next time they play the game.

Reducing Requirements

UR2_EDUCATIONAL_BUILDINGS, UR3_RESIDENTIAL_BUILDINGS, UR4_RECREATIONAL_BUILDINGS, UR5_DINING_BUILDINGS. All of these requirements have now been reduced to a single requirement called **UR_BUILDING_VARIETY**. This requirement states the player shall be able to place at least one building for each category such as: educational, residential, recreational and dining. The reason that the previous requirements have been reduced into one is because they are too specific and unnecessary to cover in so much detail.

FR2_EDUCATIONAL_BUILDING, FR3_RESIDENTIAL_BUILDINGS, FR4_RECREATIONAL_BUILDINGS, FR5_DINING_BUILDINGS. All of these requirements have now been reduced to a single requirement called **FR_BUILDING_VARIETY**. This requirement states that the system will allow the user to place down buildings. These buildings will affect student satisfaction either positively or negatively from its distance to other buildings.

NFR2_AUDIO_CONTROL has been removed from the Non-Functional requirements table as it is too specific and derivative of **NFR4_IMMEDIATE_RESPONSIVENESS**. Updated the description of **NFR4_IMMEDIATE_RESPONSIVENESS** to reflect the change by including "or mute/unmute the game".

NFR5_MINIMAL_LOADING has been removed from the Non-Functional table. This is because load times has already been mentioned by **NFR11_TRANSITION_LOAD_TIMES** as the startup or transition between screens but if **NFR5_MINIMAL_LOADING** is referring to load times between the player's input and when the action is visually displayed, this has also been discussed through the **NFR4_IMMEDIATE_RESPONSIVENESS** requirement. As a group we discussed the suitability of this requirement and thought it best to remove it as it is covered by one of the two requirements above.

NFR9_BUILDING_RESTRICTION_ENFORCEMENT has been removed from the Non-functional table as it is not a non-functional requirement and does not accurately represent the definition of a non-functional requirement "a quality that the system must have", it is closer to the definition of functional requirement "an action that the system has to take to provide useful functionality to its user". There already exists a functional requirement to do with building restrictions **FR11_BUILDING_RESTRICTIONS** and so **NFR9_BUILDING_RESTRICTION_ENFORCEMENT** is not needed and has subsequently been removed.

Changes to existing requirements

Only one of the original requirements was changed; it was in the user requirements table **UR12_BUILDING_RESTRICTIONS**. The part of the description we are changing is the "and

will have limits to the number they can place". As a team we interpreted this requirement as using some method to try to limit the user from placing too many buildings. There was no more description as to what the "limit" could mean and the requirement was not implemented by the end of Team 10's Assessment 1. We decided that since the "limit" was not defined we would change this to mean the "cost of buildings" i.e buildings will have a cost in order to limit the number of buildings the user can place.

Most of the other changes were minor and had to do with the tables themselves. One thing we changed was the introduction more specifically the "Presentation of Requirements" to include the fact that the requirements were presented in a table format as well as what each column means for each table.

Another change that was done to the requirements was the ID's of the requirements. Each of the requirements was numbered within the ID column however when referring back to the ID's the number was not mentioned e.g the user requirements column within the functional requirements and non-functional requirements table do not have a number whereas the user requirements table has numbers for the ID column. As a team we decided to remove the numbers from all of the requirements ID as it would still be unique but now easier to read.

A minor change was done to all of the requirements tables and that was to format the table properly. We increased the size of the ID and Description columns to the point that they took up the page's width so that the tables were more readable instead of being cluttered like before.

Inherited deliverable URL: https://grumv2.github.io/Eng1Ass2Web/old_assets/Req1.pdf
Updated deliverable URL: https://grumv2.github.io/Eng1Ass2Web/WebDocs/pdfs/Req2.pdf

Architecture:

Quite a few changes to the deliverable have been made, firstly and unintentionally the previous groups images have been erased accidentally when converting it from a PDF to an editable medium like markdown. For clear separation from the previous teams architecture deliverable content there is a section at the end of their work called ‘Changes to the previous groups architecture’.

Some things regarding the architecture the developers deemed unsuitable for their approach to development so they made changes to it, so with some explanations and numerous diagrams to reflect the changes made up to that point.

The first change done was restructuring the codebase to a ‘clients’ and ‘entities’ package which made use of all of the previous groups code with additional classes for extra abstraction. The respective abstract classes which accomplish this are ‘ForegroundEntity’, ‘Button’ and ‘Obstacle’ for their respective child classes.

There is a whole new ‘Server’ package which carries out the requirements not fulfilled by the previous groups game like a scoreboard and game end state. The server also fulfils the new requirements that were introduced in part 2 of the project like achievements. The new classes added to interact with the Server are ‘End’ and ‘Menu’ classes created in the ‘Entities’ package. The is another important package ‘Customs’ and its class ‘Border’, now named ‘Broker’ will be the core interaction between the client and server classes which fulfils the data transferring needed between them without direct interaction. This ensures that the code is easier to allocate between the two developers.

There are a handful of new diagrams for the architecture which serve as a snapshot of the progress at the time of programming. The diagrams also represent the changes to the architecture from start to finish in a visual way which makes it easier to convey the intent than in plain text.

Inherited deliverable URL: https://grumv2.github.io/Eng1Ass2Web/old_assets/Arch1.pdf
Updated deliverable URL: https://grumv2.github.io/Eng1Ass2Web/WebDocs/pdfs/Arch2.pdf

Method Selection and Planning:

The changes made were mainly to indicate who had taken over what deliverables as what different tools we used that the other didn’t. There was a section called ‘Team Organisation Approach’ which one of the first things they wrote about was on who was working on which deliverable. This was changed adding an indication of who in our team is taking over which deliverable.

They used Instagram as their method of communication whereas our team uses Discord so an additional point of what we used was added.

Finally some things like new snapshots using a gantt chart tracking our teams progress was set up. There is now a ‘Weekly Planning section at the end which will be updated to reflect decisions made regarding what tasks are being done.

Things noted down in the weekly planning section include the changing of architecture in a specific area as well as reallocation of roles for the deliverables after some of the members have done their respective tasks, namely the introduced Continuous Integration and Testing docs.

Inherited deliverable URL: https://grumv2.github.io/Eng1Ass2Web/old_assets/Plan1.pdf
Updated deliverable URL: https://grumv2.github.io/Eng1Ass2Web/WebDocs/pdfs/Plan2.pdf

Risk Assessment and Mitigation:

The Risk Assessment and Mitigation report from the previous team has been updated to better align with our current project requirements, including new client-requested features and improvements to our risk management strategy. These updates include adjustments to existing risks, the addition of new risks, and the removal of less impactful risks.

It is important to note that no significant changes are made into the introduction section of the Risk Assessment and Mitigation. It is because we believed the existing introduction section effectively describes the purpose of the risk assessment and also clearly explains how the risks are presented in the specific format by covering all the necessary things in a clear and concise way. However, we added a new category called project in risk categories because we believe that it would be beneficial to have the risks related to the project timeline, resources and requirements clarity.

Changes to the existing risks

For P1, the mitigation plan was changed to the one in our previous report in order to reflect our actual team task distribution plan which is more flexible than the existing one. Similarly, P6 has been also updated with the new mitigation plan to show the decision making process that we used in our team and how we solved the disagreement among the team members. Insteading of having a leader to make decisions, our team used a voting system to make the important decisions for the project in order to take account of every member’s opinions.

Newly Added Risks

Several risks are also added in the risk assessment table in order to cover the areas that are unaddressed by the previous team. In the Person category, we added a new risk P8, which is related to the misunderstanding between the team members, with the efficient mitigation plan. That risk is added because the misunderstanding is one of the most common things that can happen during the team project and it is good not to underestimate that factor in working with others.

For the Technology category, we added four new risks with the purpose of covering all the required things that should be accessed under the risk assessment. The new added Technology risks are T6, T7, T8, and T9 which are describing the risk of the code conflicts, the skill gaps, accident code overwrites on each other's code, and the loss of data because of the system error. So, these risks emphasise on the importance of technical preparedness and robust version control practices.

The Gameplay category was also expanded with the risks which are directly related to the new-client requested features, leaderboard and achievements. There are three risks related to the new features. G8 concentrates on possible mistakes in the leaderboard, G9 deals with accomplishments that don't activate properly, and G10 makes spotlight on the possibility that players would misinterpret the achievement requirements. All these risks are presented with clear mitigation plans.

The newly-introduced Project category includes four risks. All these risks are showing the potential risks that can happen during the project and details about all these risks can be seen in the risk assessment table. This category strengthens the focus on the project management and proactive planning.

Removed Risks

Three dangers were eliminated from the Gameplay category because they were considered to be more minor. The risk of controls being annoying, the risk of the game being too difficult and the risk of language not being understood are removed from the risk assessment table because their potential impact was considered to be minor compared to others and we want to focus more on the risks which can cause the significant impacts on our project.

The updated risk assessment now provides a more comprehensive and focused approach to risk management. The risk table has been updated to include newly identified risks related to client-requested features, while less impactful risks have been removed to streamline the focus on critical areas. However, the clarity and effectiveness of the risk assessment table is maintained by keeping the original structure of the table in presenting.

Inherited deliverable URL: https://grumv2.github.io/Eng1Ass2Web/old_assets/Risk1.pdf
Updated deliverable URL: https://grumv2.github.io/Eng1Ass2Web/WebDocs/pdfs/Risk2.pdf