UNIVERSITY OF YORK
DEPARTMENT OF COMPUTER SCIENCE

Change Report
Engineering 1 - Assessment 2

Group 2, Cohort 1 ("*The JVMs*")

Ben Hatch
Charlotte Sharp
Dan Nicholson
Ethan Andres
Freddie Higham
Joe Silva
Rosie Kern

Change report summary

Once we had inherited all the documents and code from the previous team we spent some time discussing any changes we planned to make to it by reviewing the work, making comparisons to our own and understanding what needed to be finished to complete the full implementation.

To keep track of the changes made in the documented deliverables (requirements, risks, method selection and architecture) we made use of Google Docs. This was appropriate for our team as it meant we were all able to access and make changes while also being able to review all the changes made at any point. This enabled us to ensure our change report would be accurate and provided the necessary means to revert back to previous versions if mistakes were made.

To keep track of any changes made to the code, we made use of GitHub for our version control system. This supplied us with a central repository, ensuring everyone in the team has access to it, and provides a history of all changes that are made in the code through commits. Within our team we ensured that when every commit was made it would be detailed with a description of its functionality so we could easily keep track of what was being changed within the code.

Requirements

The requirements set out for assessment 1 were split into 2 sections of user requirements and system requirements and these system requirements were then split down into functional and nonfunctional requirements.

Inherited requirements document:
https://jjzsilva9.github.io/uoy-eng1-g2.github.io/linkedDocuments/Req1.pdf

Upon looking at the user requirements set out by the previous team we decided to erase UR_MINIGAME as when we conducted our interview with the customer we were told it was not necessary to implement minigames so we had no intention of implementing them. Also we edited the UR_INTERACTION requirement as previously it did not include eating as an interaction and that was stated in the brief that it was needed.

We then extended the user requirements table to include the requirements we had stated in the previous assessment that were missing in this document which included: UR_LENGTH, UR_ACCESSIBLE, UR_MENU and UR_COUNTERS. These were added to ensure that our implementation would be coherent with the findings we had gathered from our customer interview, referring to an inclusive gameplay no longer than 10 minutes, and the basic requirements given in the original brief, referring to having counters for each interaction.

The user requirements also needed to be extended further to accommodate for the changes in requirements given by the updated brief for assessment 2. The brief stated that there would need to be a leaderboard showing the top 10 high scores and player names and a new feature to detect any streaks of activity included in the gameplay. To make this visible in the requirements table 2 new user requirements were added: UR_LEADERBOARD and UR_STREAKS, and also the UR_MENU was edited to include taking in the players name. We also added an additional user requirement after reviewing our feedback, UR_INSTRUCTIONS, which was used to display to the user how the gameplay worked to avoid any confusion with controls.

The final change made to the user requirements was to edit the colour scheme that was laid out in relation to the priority of the requirement. Previously it was set out that red- shall, yellow- should and green-may. While this is useful in differentiating the requirements we thought it made sense to change the scheme to: red-may, yellow-should and green-shall, as this layout follows better convention.

When reviewing the functional requirements laid out by the previous team, we decided to erase 2 of the requirements stated: FR_ITERATIONS, FR_CUSTOM_AV and FR_ASPECT_RATIO. The iterations requirement stated that the game should be designed in a way that enabled further iterations which referred to the UR_MINIGAME user requirement, and as this was also erased it was not needed. The FR_CUSTOM_AV refers to the user having the choice to customise their avatar but on reflection this is not a functional requirement as it does not directly affect gameplay rather it is just an added functionality to the user freedom in the game. FR_ASPECT_RATIO required that the aspect ratio worked on a laptop and desktop. Since it didn't refer to functionality of the gameplay we felt it was a non-functional requirement. Upon inspection of the non-functional requirements table, we

noticed that NFR_SCREEN already covered the requirement of fitting to different screens, hence FR_ASPECT_RATIO was removed.

We also extended the functional requirements further with the ones we had recorded previously including: FR_ASPECT_RATIO, FR_ENERGY_EMPTY and FR_COUNTERS. The aspect ratio requirement was added to ensure the game would present well on a desktop and laptop referring the user requirement UR_PC, and the energy empty requirement ensures that the player cannot perform activities once their energy has run out forcing the day to come to and end after 16 hours which refers to the user requirement UR_TIME_FINISH. The counters implemented were to give a breakdown of how many activities were performed each day linking to UR_COUNTERS.

To ensure the added requirements, given in the updated brief were met we extended the functional requirements to include: FR_LEADERBOARD, FR_STREAKS and FR_COUNTERS. The leaderboard requirement refers to presenting the top 10 scores on the leaderboard with the playernames and keeping it up to date. The streaks requirement states that the chosen streaks should be to do with the frequency of certain activities. The counter requirement is used to keep track of how many times an activity is completed so that the streaks functionality can be implemented.

When reviewing the nonfunctional requirements given by the previous team, we decided to erase the following: NFR_SLEEP, NFR_EXAMTIME, NFR_STUDYING and NFR_TIME_FINISH. These were removed from the nonfunctional requirements as they are more likely to be considered as functional requirements as they are needed for the user to play the game. In order to pass to the next day the user needs to sleep and in order to score in the game the user needs a place to study. It was also given that the game ended after 7 days and we did not need to implement the exam so the NFR_EXAM_TIME requirement was not coherent. NFR_TIME_FINISH was unnecessary as the ideal length of gameplay time is explicitly stated in UR_LENGTH.

We updated NFR_PC to specify that the game should be playable on PCs. To further elaborate on the system requirements, we added NFR_PLATFORM and NFR_PERFORMANCE to specify that the game should exclusively be playable on PCs and that the game should perform well even on low-spec computers.

We then extended the nonfunctional requirements to include some desirable features we had pre planned in our requirements gathering which included: NFR_DELAYS, NFR_COLOURS and NFR_ARTSTYLE. The delays were used to ensure the game was simple to follow and the art style was used to make it appealing to the correct market, referencing UR_AUDIENCE. The colours were used to ensure that the display of the game was visually appealing and not difficult to understand, referencing UR_ACCESSIBLE. We also added an additional requirement NFR_INSTRUCTIONS, which displayed the instructions to the user about how to play the game which linked to the user requirement UR_INSTRUCTIONS.

To ensure that the added requirements for the updated brief were met the nonfunctional requirements were also updated with: NFR_points and NFR_STREAKS. The points requirement refers to the algorithm in which points were awarded used to calculate a final

score at the end and the streaks requirement refers to choosing streaks of activities to unlock achievements that could be tracked by the counters and be appealing to the market. These are linked to the user requirements UR_SCORE, UR_STREAKS and UR_AUDIENCE.

Furthermore, NFR_TIME_PASS, NFR_POINTS and NFR_IDLE were moved to the functional requirements table because we felt that they are functional requirements and were previously wrongly categorised as non-functional.

The updated version of the requirements document can be seen here:

Updated requirements document:
https://jjzsilva9.github.io/uoy-eng1-g2.github.io/linkedDocuments/Req2.pdf

Architecture

When reviewing the architecture document of the previous team we also followed a similar approach using responsibility driven design, CRC cards and creating entity component diagrams.

Inherited architecture document:
https://jjzsilva9.github.io/uoy-eng1-g2.github.io/linkedDocuments/Arch1.pdf

However changes were still needed in this part of the document because of the new requirements. In order to ensure that our final system met the requirements stated, the UML diagrams needed to be extended for extra functionality.

Upon viewing the architecture provided to us, we felt that the structural diagrams consisting of packages weren't very useful as they just listed the classes with their methods and attributes without explaining what their purpose or functionality was. For this reason, we decided to remove these diagrams and replace them with ECS diagrams for all the systems of our code. The reasoning behind creating ECS diagrams was that they are very helpful when making sense of complex interactions by detailing which components are involved with each functionality of the game.

When reviewing their architecture we also found their single sequence diagram based on the score calculation to be fairly redundant. Sequence diagrams are best used to present the interactions between multiple different systems and components, rather than their diagram which only contains two classes. Therefore we opted to remove this diagram from the document and instead created new UML sequence diagrams for streaks, player name input and the leaderboard. We felt these new sequences had much more depth to them and are essential functionality in our implementation.

To ensure we had appropriate architecture coverage of our implementation, we added a state diagram to the document. Previously there weren't a large number of states, therefore a state diagram would not provide any useful information, but now with the added screens and new functionality we have enough states to warrant a diagram.

Along with adding new UML diagrams and changing old ones, for each set of UML we have included a brief introductory paragraph about the type of diagram and why it is used to represent each functionality. This change was made to help improve the readability of the document and explain our reasoning.

To form our UML diagrams we made use of Plant UML, which the previous team also did but did not justify their reasons for choosing it as a tool. We felt that PlantUML performed well for us as a tool in the previous assessment because of its versatility and ease of use when editing and changing across multiple people in the team, so we added this reasoning into the architecture document.

Due to the change in brief for assessment 2, the architecture decisions had to be updated to explain the constructs that fulfilled the new requirements. This was done at the end of the Architecture Decisions section where the implementation of UR_LEADERBOARD and

UR_STREAKS is described. This section was also updated to specify that the score is no longer calculated in EndScreen but in ScoreCalculator. This was also mentioned in the little update to the System Development Cycle section.

We also updated the System Structure section of the document to include the additional screens we created in the Screens package when developing the code.

Once these changes had been made, the architecture diagrams were representative of our final implementation.

Updated architecture document:
https://jjzsilva9.github.io/uoy-eng1-g2.github.io/linkedDocuments/Arch2.pdf

Method selection and planning

Inherited method selection and planning document:
https://jjzsilva9.github.io/uoy-eng1-g2.github.io/linkedDocuments/Plan1.pdf

When reviewing the method selection and planning document of the previous team we realised that our approach was very similar to theirs. We both adopted the agile framework, more specifically using Scrum and held meetings weekly as well communicating throughout the week. Our choices in methodology remain the same during assessment 2 as we are still a small team developing a game with known requirements and accessibility to the customer and each other to conduct meetings frequently.

We also both used Google Drive to collaborate with each other on work, Github to host our repository and create our website and discord to keep in contact with each other. For the development of the game we both used the same game engine LibGDX as it is very well suited for game development, with provided libraries for 2D games. As these tools had performed well for us during assessment 1 we decided to keep on using them in the process during assessment 2.

Keeping all these similarities in my mind, we decided to not change the first part of the method selection and planning document as our previous approach and intended future approach is very similar to what has been provided.

In terms of team organisation we followed a similar structure of adopting a flexible approach based on the strengths and weaknesses for each individual. However a difference we did have was that we did assign specific tasks to members of the team to ensure that there was an even split of work and to promote productivity. We then shared our work with each other during the meetings and also by committing changes frequently. This change can be observed in the new method selection and planning document under the team organisation description.

In order to continue working on implementation in the most productive way we followed the same structure of the previous team of producing Gantt charts to visualise the progress we were making. We extended the document to include weekly breakdowns of the work following the same procedure as the previous team, which can be seen at the bottom of the method selection and planning document as well as updating the Gantt chart weekly with any progress that had been made. This was done to ensure that we would be working on a realistic time frame while completing the implementation and deliverables needed.

Upon reviewing feedback we ensured that the Gantt charts we created were more detailed with the tasks we were completing each week. For example we broke down the implementation stage into stages of implementation like map changes and the score function. We also added a dependency table which listed all the tasks we were required to implement for assessment 2 with an ID, description, start date, end date and any dependencies around that task. The previous team did not do this and we thought it was appropriate as it outlines an order of how tasks should be completed based on existing dependencies. For example we knew we would have to implement a method of getting a

player name before the leaderboard would be fully implemented. This ensured that we were always able to make progress each week.

Updated method selection and planning document:
https://jjzsilva9.github.io/uoy-eng1-g2.github.io/linkedDocuments/Plan2.pdf

Risk assessment and mitigation

Upon reviewing the adopted risk management and mitigation plan of the adopted project, it came to our realisation that the development and management of the risk register was very similar to our own risk register. Like us, they followed the risk management process layed out in Ian Sommerville's book, Software Engineering. This meant that our group was very familiar with the layout and the maintenance techniques used in the risk register.

Inherited risk assessment and mitigation document:
https://jjzsilva9.github.io/uoy-eng1-g2.github.io/linkedDocuments/Risk1.pdf

Similar to us, the previous team also allocated time in their meetings to discuss the risk register. The previous team specifically mentioned that the risk owners would bring the group's attention to a risk if the owner felt that there needed to be a greater mitigation strategy for that risk. We decided that this would be a useful protocol to adopt but we changed the wording because we felt that the need for "greater mitigation" was quite limiting. This was changed to a need for a "change in mitigation strategy" because it better reflects our flexible mindset when it comes to reassessing our risk register; unnecessary mitigation strategies can be updated to be less drastic.

Both of our risk registers were very similar so there was no need for big changes in structure as we were already familiar with it. The only change made to the risk register structure was that the owners column was split into separate primary and secondary owners columns. This was done in order to keep the bus factor up while also ensuring a single point of resolution when monitoring the risks as the secondary owner would only be involved if the primary owner became unavailable.

One difference between our risk registers was the use of a "Last Assessed" column which we were happy to adopt because of its ability to identify neglected risks. However, we felt that the list of risks the previous group provided didn't fully encapsulate the range of potential risks involved with developing a video game. Due to this, we added our own risks until we felt that the risk register sufficiently covered the relevant risks that may occur during the project. Furthermore, we added the "Legal" category for the risks as that had been completely overlooked by the previous team.

The only update we made to the risks of the previous risk register was the likelihood and severity ratings for R9. We felt that the chance of the group overlooking the requirement of interactable locations was quite low as we're all familiar with the requirements of the game so the likelihood rating was changed from M to L. In addition to that, if that risk were to occur, it wouldn't be detrimental to the project and could be fixed with a discussion on how to meet that requirement so the severity rating was changed from H to M.

Finally, the risk owners were reallocated where team members took ownership of risks that related to the risks they owned in the previous project.

Updated risk assessment and mitigation document:
https://jjzsilva9.github.io/uoy-eng1-g2.github.io/linkedDocuments/Risk2.pdf