Change report

Cohort 1 group 11

Freddie Aberdeen
Mutaz Ghandour
James Given
Jasper Owen
Jungwan Park
Joe Reece
Ivan Shestakov

Process

For each deliverable we inherited from the other team, we reviewed it and established what we thought was good about it and what was bad about it. We then went over each deliverable and made any changes that we felt were necessary.

The deliverables were Google Documents so as we made any changes, we commented what the changes were so we would know what to write about in the change report. While we were doing this, we made sure that we still had access to the original documents as a reference.

We also used the version history feature of the google document to give us the option to reverse any changes we made, just in case we decided the other team's version of a deliverable was better than the changed version.

Once we were satisfied with the changes we had made to a deliverable, we reviewed the comments detailing the changes made and wrote about them in the change report, including why we changed it.

For some deliverables, such as Method Selection and Planning, we read about the techniques the other group used in Assessment 1. While these techniques were all valid, we preferred to use our own techniques from Assessment 1 in Assessment 2, because they were what we were familiar with so we altered the documents to reflect this

Tools and Conventions

All deliverables were edited in Google Documents. This allowed us to store the documents on our Google Drive, where any group member could access them and also allowed us to use comments, meaning we could clearly mark any changes made to the deliverables, which made writing the change report much easier.

The change report was also made in Google Documents to allow any group member to access it if they had a change that they wanted to describe.

As we wrote our changes we included what the deliverable was like before the change, what it was like after the change and why we changed it. This allowed us to clearly describe each change and justify it.

Also, we grouped the changes made to each deliverable together to make the document organised and easy to follow. We also described the changes in chronological order so the changes we made to each document would be easy to follow.

Requirements

Requirements change 1 - Leaderboard requirements added

Before: The previous team did not include any requirements for a leaderboard as it was not included in the Assessment 1 brief

After: The requirements UR_LEADERBOARD, FR_LEADERBOARD and NFR_LEADERBOARD were added to the requirements document

Reason: The game having a leaderboard was added to the brief as part of assessment 2 so we needed requirements to account for this.

Requirements change 2 - Achievement requirements added

Before: There were no requirements relating to there being achievements that the player can complete in the game

After: We added the requirements UR_ACHIEVEMENT, FR_ACHIEVEMENTS, NFR_ACHIEVEMENTS and UR_ACHIEVEMENT_SCORES to the document

Reason: The Assessment 2 brief added that there must be achievements that the player can achieve while playing the game, so we added requirements to reflect this

Requirements change 3 - Data storage requirements removed

Before: Some of the original requirements mentioned that the game could not store any data about the user such as their score

After: These requirements were removed from the document

Reason: Not storing data about the User contradicted the requirements for having a leaderboard, which was part of the brief, so we got rid of these requirements.

Requirements change 4 - UR_COUNTER edited and FR_COUNTER and NFR_COUNTER added

Before: The requirements UR_COUNTER mentioned that the game should contain an event counter

After: UR_COUNTER now states that the user can see how many of each event they completed and the new requirements FR_COUNTER and NFR_COUNTER describe the functionality of the event counter

Reason: We did not feel that mentioning what the game should contain was suitable for a User Requirement.

Requirements change 5 - Corresponding User Requirement for Functional Requirements changed.

Before: The corresponding requirement for the requirements FR_VISIBLE_HINDRANCES, FR_VISIBLE_BENEFITS and FR_HIDDEN_EVENTS was UR_DIFFICULTY

After: The corresponding requirement for the requirements FR_VISIBLE_HINDRANCES, FR_VISIBLE_BENEFITS and FR_HIDDEN_EVENTS was UR_EVENTS

Reason: We felt that UR_EVENTS was a more suitable requirement than UR_DIFFICULTY because UR_EVENTS describes how the user shall encounter events, while UR_DOFFICULTY does not mention any events.

Requirements change 6 - FR_SCORING_SYSTEM edition

Before: The requirement FR_SCORING_SYSTEM mentioned that their score would be calculated using events completed and time taken, but did not mention how the score would be calculated.

After: FR_SCORING_SYSTEM mentioned how events can affect the score and that completing the game more quickly results in a higher score.

Reason: We wanted the description of how the score would be calculated to be clearly reflected on our requirements, as well as for all our requirements to be as accurate as possible.

Requirements change 8 - NFR_ACCESSIBILITY criteria changed

Before: The requirement NFR_ACCESSIBILITY said that the game would be ‘universally understandable’ and did not have good criteria to test it against

After: NFR_ACCESSIBLITY describes how most players shall understand the UI without assistance and has criteria of 90% of players successfully understanding it

Reason: Calling a game ‘universally understandable’ felt vague and we needed criteria that we could reliably test the requirement against for it to be valid, so we changed the wording and added suitable criteria.

Requirements change 8 - NFR_IMMERSION, UR_ENJOYABILITY and NFR_ENJOYABILTIY added

Before: There were no requirements to account for the player feeling immersed in the game or enjoying the game

After: The requirements NFR_IMMERSION, UR_ENJOYABILITY and NFR_ENJOYABILTIY were added

Reason: Our Assessment 1 feedback recommended that we could include a Non-Functional requirement for user immersion and enjoyability.

Method Selection and Planning

Method selection change 1-Trello to GitHub Issues

Before: The original document from group 5 used trello to track the status of tasks and to monitor the sprint progress through Kanban boards.

After: We replaced all references to Trello with GitHub Issues, as our team actually used GitHub for the Assessment 1 project. Tasks were assigned to individual team members through GitHub Issues with assigned deadlines every each week. This allowed us to track progress, manage the list of tasks, and make sure the development work was organised and running smoothly.

Reason: The original document described the tools used by the previous team, but our team worked differently in Assessment 1. We used GitHub Issues instead of Trello for task management, including assigning tasks and tracking progress. This adjustment ensures the method represents how our team actually worked.

Method selection change 2 - Documentation Tool Updated(Google Docs added)

Before: The original document did not specify which tools the previous team used for documentation.

After: We added Google Docs for documentation, since our team used it throughout Assessment 1 for writing, editing, and sharing documents collaboratively.

Reason: All of our team members have experience using Google Docs and used Google Docs for Assessment 1 as well.Furthermore, it was easy to share and access documents and was more familiar for our team compared to alternative tools such as OneDrive Word, therefore we decided to add Google Docs.

Method selection change 3 - Communication Tool Updated(WhatsApp Community added)

Before: The original document did not specify any communication tools used by the previous team.

After: We added a WhatsApp Community as a communication tool because our team used WhatsApp to communicate with each other, share important information, and adjust the meeting schedules.

Reason: Our group used WhatsApp for communicating with each other, sharing important information, and adjusting the group meeting schedules during Assessment 1. All team members preferred WhatsApp rather than other tools such as Discord, so we added WhatsApp to accurately reflect our group's communication workflow.

Method selection change 4 - Team Organisation Updated(OSRC deleted)

Before: The original document mentioned team organization by dividing into two main groups, which were technical and non-technical. This method was suggested by the Design Team Alliance in OSRC and they decided to use it to maximize the efficiency of assigned tasks based on their individual strengths.

After: Our group did not use the OSRC method to divide into subgroups, so the team organisation was updated. Instead, we divided into two groups, which were documentation and coding, and further subdivided tasks in each group. Furthermore, we conducted two weekly meetings to ensure that each member's work stayed consistent and on schedule.

Reason: The biggest reason we updated is that our group did not use OSRC for team organisation. Instead we divided into documentation and coding groups based on our actual workflow. Therefore, we removed the OSRC reference and updated it to align with our team organization.

Method selection change 5 - removed evolution of the plan

Before: The original document included an "Evolution of planning" at the end, where the previous team described how a group member became ill and this affected their project schedule.

After: This section was removed.

Reason: Illness-related planning information reflected the previous team's situation, which is not related to our group's planning. To maintain consistency and accurately represent our workflow, the "Evolution of the Plan" section was removed.

Method selection change 5 - Systematic plan added

Before: There was no systematic plan for the whole project mentioned in the document

After: A plan for the project with our objectives for the whole project, as well as our objectives for the Assessment 1 and Assessment 2 sections of the project was added. This included their start and end dates.

Reason: Including a systematic plan was mentioned in the assessment document. In addition, the lack of a systematic plan in our original document was mentioned in our Assessment 1 feedback so we made sure we included one in our new Method Selection document.

Risk Assessment

Risk Assessment change 1 - Likelihood/Impact Criteria added(L/M/H)
Before: The original document used Low, Medium, and High for rating for the likelihood and impact, but it did not explain what each level represented.

After: We added definitions of each level to represent likelihood and impact.

Reason: Without defined Criteria, the risk assessment was subjective and could lead to different interpretations. To maintain consistency, we add these definitions to make it more reliable.

Risk Assessment change 2 - Risk Ownership Updated
Before: The ownership in the Risk Register was assigned to members of the previous team.

After: We updated the ownership in the Risk Register to our team members' names. Based on the updated ownership, we ensured that each member is clearly responsible for monitoring their assigned risks.

Reason: The original ownership was relevant to the previous team, so we updated it to make the risk register accurate and clearly show the responsibility for each risk.

Risk Assessment change 3 - Risk Register Improved
Before: The original document did not include a Risk ID column and did not mention risks related to us taking over the project.

After: To keep track of the risks clearly, we added a Risk ID column to each entry and included three additional risks at the bottom of the risk register to account for challenges we faced when taking over the project.

Reason: To identify the risks more easily, the Risk ID column is important for organising and referencing each entry, so we added a Risk ID column to the register.

Risk Assessment change 4- Risk Level Terminology and Colour Coding updated
Before: Risk levels were evaluated as Middle/Normal in the original document by the previous team there was no clear method to distinguish the different levels.

After: We standardised the terminology by changing "Middle/Normal" to "Medium" and updated the colour coding.

Reason: To keep consistency in the Risk Register, We updated the "Middle/Normal" to the "Medium" and colour coding to minimise confusion when interpreting the risk levels and make it easier for our team to understand as we take over this project.

Risk Assessment change 5- Status Column Removed

Before: There were values such as “Finished” and “In progress” in the Status Column in the original document.

After: We decided to remove the status column from the original document because it is not suitable for Risk Register.

Reason: A risk can not be marked as “Finished” before the project is submitted. For example, an illness of a team member, potential technical problems could be ongoing when the project is finished. Therefore, we removed the status column from the original document

Risk Assessment change 6- Extra risks added

Before: There were no risks in the assessment to account for us taking over the project and any risks that would entail.

After: Risks 15, 16 and 17 were added to the risk assessment

Reason: We added Risk 15 because there was a lot of documentation work that needed to be done, but at the same time we needed to have a presentable game that worked. Therefore maintaining a balance between both aspects of the project was important, so we decided to add the possibility that we would not to our risk assessment.

We added Risk 16 because there was a possibility that we would struggle to understand and edit the code made by the other team and this felt like a genuine risk that needed to be included in the assessment.

Also, we added Risk 17 because our group had divided into coding and documentation in Assessment 1 and we intended to continue with this approach in Assessment 2. A lack of communication between these subgroups could result in inconsistencies that would cost us marks. Therefore this risk needed to be included.

Risk Assessment change 6- Impact re-evaluation

Before: Risks 3, 6, 7, 10, 11 and 17 were given an impact of high

After: The impact of the above risks was switched to medium

Reason: Following our Assessment 1 feedback, in which we were told to save High/High Medium/High and High/Medium for the most likely/impactful risks, we re-evaluated the risks with High designations and decided to switch the impacts of these risks to medium to make it easier to manage the remaining High impact risks and emphasize their impact.

Risk Assessment change 6- Usability Risks added

Before: There were no risks considering what the users may think of our game

After: Risks 18 and 19 were added which concerned enjoyability and difficulty
Reason: In our Assessment 1 feedback we were advised to include usability risks

Architecture

Architecture Change 1- Scoring and Tutorial Explanation updated

Before: The architecture only explained that scoring and tutorial features were supported in a vague way without explaining more details about the scoring system and the tutorials.

After: We updated the architecture description in more detail, such as how the score is calculated during gameplay based on time and events, and how the final result is shown on the ending screen. Moreover, we also clarified that the tutorial system provides instructions on controls and objectives.

Reason: In the requirement document, we updated the scoring system and the tutorial behaviour was clarified, but this architecture document still described these features in a vague way. Therefore, this change could be made to improve clarity and maintain consistency between the two documents.

Architecture Change 2- Mention of Plant UML

Before: The previous group only used UMLet to make their diagrams

After: We mentioned that some interim diagrams were created in UMLet but we also mentioned that other diagrams were created using PlantUML.

Reason: We decided to use plantUML to make any new diagrams for Assessment 2, as that was what we used in Assessment 1 and we found that it was simple to use and helpful for making diagrams

Architecture Change 2 - Updated diagrams

Before: The diagrams in the document showed the architecture of the game as it was when the other group submitted it

After: The diagrams in the document were new ones that were up to date with our continued development of the game

Reason: If we had kept the original diagrams, our architecture would have been outdated, so we created updated versions

Architecture Change 3 - Architecture description changed from ECS to Object Oriented

Before: The original group said that their architecture was ECS

After: We described our Architecture as Object-Oriented

Reason: We felt that the architecture of the game did not match the group's description of ECS. This was because many of the entities and components in the game contained logic, which should not happen in ECS. Therefore we decided it would be more accurate to describe our architecture as Object-Oriented than ECS

Architecture Change 4 - Specific requirements assigned to specific paragraphs in related requirements section

Before: In the related requirements section of the architecture, the original group would include the requirements relevant to each sub-section at the top of each sub-section, and then write about the features that met these requirements without actually linking these features to the requirements.

After: Each paragraph that describes a feature in the related requirements section is directly linked to the requirements it meets

Reason: Not linking the requirements to the features it met made the description unclear as to how the requirements were met and what features of the game actually met the requirements, whereas directly linking requirements to each feature makes it clear how each requirement is met.

Architecture Change 5 - Architecture Refinements section added

Before: The previous architecture document did not include several architectural refinements made during development.

After: To document the previous team's architectural development decisions, an Architectural Refinements section was added, which explains package reorganisation, separation of responsibilities within core classes, introduction of the Engine class, removal of static variables, and improvements to asset and rendering structure.

Reason: The Architectural design decisions were already present in the codebase but not documented in the previous architecture document. This section was added to avoid the inconsistencies between the documentation and the implementation of the codebase.