2. Change Report

Group 11 - 11 Musketeers

Osama Azaz
Adam Dawtry
Tom Jackson
Brendan Liew
Holly Reed
Harry Ryan

2.a

- Trello was used to create an extensive to-do list, consisting of separate boards for art work, the change report, implementation, and the general plan.
    - Each of the tasks on the list are assigned to one or more team members.
    - When a team member begins work on a task, the task can be moved from the *To Do* section into the *Doing* section, and when it is completed it can be finally moved into the *Done* section.
- When making changes to previous group's deliverables, the *Suggesting* function of Google Docs is used to plan and suggest additions/deletions which can then be reviewed by the rest of the group.
- Changes in code are tracked using Trello and its *Label* function.
    - Each task is given one or more labels relating to the type of change (like a bugfix), or in which sprint the task has been created.
    - A comment is added to each task when each task has been completed and merged into the main Github repository.
- All changes are compiled in part **2.b** of this deliverable, each with the date implemented and an explanation and/or justification for the changes made.
    - This section is split into changes made to the deliverables: requirements, abstract and concrete architecture, methods and plans, and risk assessment and mitigation.
- Many resources [1][2] state the need of a change request form or change control form to track changes, however Ian Sommerville makes a good point in [1] that this method of recording changes in detail may be more useful for large projects which have many people working on it.
- Somerville also recommends focusing on "describing the change required, with less emphasis on implementation issues". Therefore, we have decided to simply state the date of the change, a description of what was changed, and why. We believe that this is enough information to accurately track changes without overwhelming team members with too much information.
- When changes are being considered, key points are considered as stated in [3].
    - Any risks, alternatives, resources, and reasons relating to the change are reviewed before a team member suggests a change.
- A more streamlined process of change management has been created by combining principles and steps of methodology discussed in [1] and [3].
    - A request for a change is created on the Trello board.
    - The team member assigned to the task reviews the request and makes changes.
    - Once a change has been made, the rest of the team is notified by updates made to the board.
    - If the team is happy with the change, it is implemented and the request is closed.

2.b

Requirements:

- **04/03/22:** Requirements deliverable is updated with the new requirements required for Assessment 2.
    - Requirements UR.OBSTALCES, UR.WEATHER, UR.SPEND_LOOT, UR.POWER_UPS, UR.DIFFICULTY, UR.SAVE_LOAD and UR.ATK_SHIP have been added to the User Requirements table on page 3 to ensure the requirements are up to date.
- **22/04/22:** Removed the 'Difficulty' column for each requirements table and removed the 'Priority' column for the Functional, Non-Functional, and Constraints tables.
    - This was done as these columns were seen as redundant and did not offer any additional useful information to the document.
- **22/04/22:** Added the 'User Requirements' column to the Functional requirements table.
    - In the first half of the assessment, these columns helped us to make sure that our implementation was fully completed, so we have decided to include it in the new Requirements deliverable. It also allowed us to see if any changes to a user requirement or functional impacted any other requirements.
    - This also allowed us to see if any user requirements were missing. We found that a user requirement rendering enemy ships was missing, therefore UR.ENMY_SHIP was added.
- **22/04/22:** FR.ATTACKCURSOR was updated to include attacking ships.
    - A complete check was done over the Functional requirements table to ensure that they were fully up to date for the new assessment and completely linked back to the user requirements.
- The remaining unchanged requirements and section 2.(a) were deemed to be satisfactory and therefore were not changed.

Abstract and concrete architecture:

- **04/03/2022:** Sound Manager and Texture Handler is added to the justification for abstract and concrete architecture for assessment 1 that is located on page 5 and 6.
    - **Texture Handler** class was added because the assessment 1 implementation had no way to dispose of the textures after the game was closed. Without texture disposing, the code could cause the game to have problems such as memory leak.
    - **Sound Manager** class was added to the game because it was listed as a functional requirement in assessment 1 deliverables which the previous group didn't have time to implement.

- Abstract architecture diagram was left unchanged as the diagram has already captured the core classes and system states.

Methods and plans:

- The precise URL to the updated plan:
- The reason for most of the below changes were to ensure that the entire document is relevant to our team. This included small adjustments to some sections and the

removal of some content which have not all been listed here to prevent the recording of minor changes.

- **22/04/22:** ‘Methodology’ on page 2 has been updated to match our agile approach.
    - Further explanation has been provided in the section which explains our choice and how we used the approach.
- **22/04/22:** ‘Development environment’ on page 3 has been updated to the correct environment our group has used and the reasons why it was chosen.
- **22/04/22:** The sections: ‘Communication’, ‘storing our documentation’, ‘Trello’, and ‘version control system’, were mostly unchanged as we used the same software and methods as the previous team.
    - Some small changes were made, for example, the tags we used on our Trello boards were included in the document.
    - Sections that were not relevant to our group were removed.
- **25/04/22:** Section 4.(b) has been completely updated so that it is relevant to our team.
    - This involved deleting the entire section as we think that it had no relevance to us.
    - Replaced with the previous assessment’s section 4.b with some minor changes.
- **29/04/22:** Section 4.(a) has been updated to include mention of our use of IntelliJ at the end of the project for testing purposes. This is in the development environment section.
- **29/04/22:** Section 4.(c) has been completely rewritten to include our own systemic plan and the evolution of the plan.
    - The previous group’s plans were removed from the document to ensure that the deliverable did not go over the page limit.

Risk assessment and mitigation:

- **25/02/2022: Replaced an old risk in the risk catalogue ID:005**
    - **OLD: ID:005,** Misunderstand which features our customers truly care about
    - **NEW: ID:005,** Project deliverables delayed due to other assessment within the term
    - Why: Because **OLD** is the same with ID 013, which is Misunderstanding requirements of project,in which the requirements of our customers and our project listed is the same.which is why replacing it would reduce its redundancy

- The remaining risk assessments were not changed as they were deemed satisfactory.

References

[1] I. Somerville, *Software Engineering*, Global Edition. Pearson Education, 2016. [E-book] Available: https://ebookcentral.proquest.com [Accessed: 11 March 2022].

[2] M. Martin (2022, Feb. 18). *Change Control Process in Software Engineering with Steps*. Guru99. [Online]. Available at: https://www.guru99.com/change-control-business-analyst.html [Accessed: Mar. 11, 2022].

[3] S. Jena (2021, Aug. 18). *Change control | Change Management in Software Engineering*. GeeksforGeeks. [Online]. Available at: https://www.geeksforgeeks.org/change-management-in-software-engineering/ [Accessed: Mar. 11, 2022]