Change report

Group Name
Cohort 1, Group 4

Group Number
Cohort 1, Group 4

Members
Jude Hall
Rosie Hogg
Ishraan Ismail
Sam Wildgust
Ruby Hanson
Tom Devany
Tomas Asllani

Tools & Conventions Used

Implementation of the previous group's deliverables served as a foundation when deciding what processes and tools we would use to make changes in Assessment 2, by following the provided website documents and the arrangement of the code.
Initial allocation of the deliverables in Assessment 2 was decided based on our experience in Assessment 1, and this was set up through a Google documents file where we assigned ourselves based on our personal strengths.

To collaborate, we cloned the repository from Assessment 1 and used this to set up our own private Github, where we used multiple branches to distinguish new versions of the game to be passed between team members and systems for testing. Simultaneous to development, the allocated group members were working on testing the original version of the game, using a very clearly defined version in the Github repository so it would not affect the ongoing game development, and each test was recorded in Google docs with the name of the person responsible, and labelled in different branches of the repository. We have also set automated tests for build and dependencies for each version of the game submitted; this helps to ensure that all of the programs submitted are valid. These tests were used through consulting lecture material on GitHub [1] in addition to online forums [2].

To develop the game, we followed the previous teams' software: LibGDX, IntelliJ and Tiled to ensure the most efficient project-crossover. There were various ways we added assets: through Piskel for more University based events, and online assets for other events. For the sound effects and music, these were found free online.
The people who were developing the game would work on a separate version each, one person per set amount of time, e.g. a week, and then it would be reuploaded to the main branch with the updated date so that it could be followed easily. Additionally, if short for time, multiple people worked on different aspects of the game at the same time: such as events and more functional requirements; while somebody else worked on more user requirements like the visual aspects. At the end of the week they would be combined so code could be reviewed, and the game manually tested with each event (hidden, positive and negative even if they do not directly lead to completion) by somebody else in the group to find issues that were missed and quickly mended.
Using the documentation and separation of systems/entities in the initial code, it was very simple to add new events and changes to the game. Followed the files setup to add our own systems, 'AudioSystem' and 'TextInputSystem' to look coherent with the existing and ensure usability of code so that it can be developed further. Javadoc and comments were written in the same format and style as the previous implementation.
In terms of the previous group's documentation, constant referral to the requirements, accessed through their website, was required to make sure we met the specifications that had been set by the customer before Assessment 1 began.

Requirements

Original Req1 document: https://eng1-group9.github.io/website/Req1.pdf
Excel spreadsheet containing the requirements 1:
https://docs.google.com/spreadsheets/d/1UP-v_Cf7AY_UWcs-ezlvnMIeOC-M6YL_T-Sn0tC_c0k/edit?gid=0#gid=0
Requirements 1 Table Reference document:
Updated Req2 document:

In the original project, the formal requirements were maintained in an external Excel spreadsheet, while the main Req1 documentation contained descriptions and design discussion rather than a structured requirements specification. For assessment 2, this approach was revised to improve clarity and traceability. “Requirements 1 Table Reference document” contains the original requirements extracted from their excel sheet for clarity.
In our updated document, requirements were reorganised into structured tables separating user requirements, functional requirements, non-functional and constraint requirements with clear identifiers and descriptions, following guidance from the lecture slides [3] and IEEE-style requirements practices.

The User Requirements, as seen in page 3 of https://eng1-group9.github.io/website/Req1.pdf and by *Kovolos (2025)*, are defined as statements regarding the tasks that users should be able to carry out using the system.

Figure 1 - Modified User Requirements

ID	DESCRIPTION	PRIORITY
UR_EVENTS	The game shall include 3 positive, 5 negative and 3 hidden events for current iteration.	Shall
UR_SOUND	The player shall hear sound effects when events occur.	Shall

Using this, changes were made to the updated document. ‘**UR_EVENTS**’ and we have added the requirement ‘**UR_SOUND**’ as seen by the above table.
The UR_EVENTS requirement description changed to contain the requirements for Assessment 2, as instead the game should contain an increased number of events from 1 of each type.
The UR_SOUND requirement builds off the system ‘AudioSystem’ that we added and was added to this table to signify that each event should contain a sound effect to indicate to the user what event has occurred - for potential future iterations.
Additionally as **UR_SAVE** was not within the project brief, there was the decision to remove this requirement due to the time constraints of the game production so we could instead focus on improving the functionality of existing and new events.

Functional Requirements

The Functional Requirements, as seen in page 4 of https://eng1-group9.github.io/website/Req1.pdf and by *Kovolos (2025)*, are defined as things a system must do that allows functionality to a user.

Figure 2 - Modified Functional Requirements

ID	DESCRIPTION	USER REQS
FR_SOUND	The game will have sound effects to distinguish positive/negative events and constantly have background music playing.	UR_SOUND
FR_SCORE_CALC	The score constantly updates and is shown on screen so the player can always see how they are progressing.	UR_SCORE

Using this, we have made changes to the updated document so '**FR_SCORE_CALC**' and **'FR_SOUND'** were added.
FR_SOUND corresponds to the added user requirement for sound, to abstract it into a requirement for what the game function should look like to meet that.
FR_SCORE_CALC was modified, as one change made to the game was to have the score constantly update on screen rather than only calculating the score at the end. Therefore the description of the original functional requirement was modified to reflect this.

Non-Functional Requirements
The Non-Functional Requirements, as seen in page 5 of https://eng1-group9.github.io/website/Req1.pdf and by *Kovolos (2025)*, are defined as constraints on functional requirements and qualities a system must have.

We decided that there were no necessary changes required to the non-functional requirements as these were all met in Assessment 1 and no additional requirements were added to be met in Assessment 2 apart from functionality of the game, which this does not reflect.

Constraints
The constraints, as seen by *Kolovos (2025)*, are defined as global issues, usually defined in the beginning, that shape the requirements.

As by this definition, and similarly to the non-functional requirements, there were no necessary constraints to be added or modified as these were all stated clearly in the original brief. Plus, the Assessment 1 implementation reached these requirements in a way that we could inherit them rather than having to change any.

Architecture

The document for this section was updated to reflect the requirements and development practices of assessment 2. While the original agile-based structure was retained in the method selection section, the justification and examples were revised to align with brownfield development rather than greenfield development. The updated document focuses on extending and integrating with existing codebase, while the client requirements still remain relevant. The description of Agile, XP, and Scrum was refined to accurately represent the practices used by our team.

The documentation section of the original document remained unchanged since our group also made use of google docs for this assessment.
The section discussing Apache Subversion as an alternative version control system was removed in the updated document. As assessment 2 focuses on extending and maintaining an existing codebase, Github was already in use. Therefore, evaluating alternative control platforms was therefore no longer relevant.
References to Discord in the communication section of the original document were removed in the updated document, as communication was conducted only through Whatsapp by our team.
The version control section of their document was expanded to include the use of continuous integration (CI) as required by assessment 2.

The original project plan content was replaced with a revised assessment 2 project plan describing project selection, repository setup, early testing challenges, incremental implementation, user evaluation, documentation updates, and preparation for the assessed presentation.
The original project plan table and Gantt chart was removed and replaced with Gantt charts representing the assessment 2 project timeline. These changes were required to ensure alignment with assessment 2 objectives and to reflect actual development practices used by the team.

Risk Assessment & Mitigation

Original Risk1 document: https://eng1-group9.github.io/website/Risk1.pdf
Updated Risk2 document:

The risk document was reviewed and updated to better align with the scope and requirements of assessment 2. The risk management process is described and appropriate, thus remaining unchanged in the updated document as well.

For the change, first of all, the updated risk register introduces colour coding to visually indicate risk severity and impact. Secondly, while the original document categorised risks into project or technical, the updated document introduces a third category, product, to better reflect risks related to gameplay, usability and player experience. Additionally, the risk register format was also revised, so that the risk ID and risk type are presented as separate fields, with risks being ordered from R1-R11 instead of T, P and TP. This change was made to align the risk register structure with the format presented in the *Project Planning and Risk Management* slides [6].

In the original risk register, some risks were assigned to multiple owners. In the updated version, each risk has a single assigned owner who is responsible for monitoring and managing that risk, ensuring clearer accountability. The risk monitoring information was removed from the risk register table for it to be handled through ongoing project activities by their owners rather than being documented per risk in the table.

Several risks from the original document were reviewed for relevance to assessment 2. While some risks like (TP1, P2, T3,T4,T1) from the original document remained unchanged in the updated document, risks P1 and T2 are insufficiently relevant to the inherited codebase and development context of the assessment therefore not included in the updated document.
Some risks were also identified as overlapping in meaning. P2 and TP4 were regarded as representing the same underlying issue. Similarly, TP2, T5, and P3 were reviewed and updated (R11) due to their similarity in scope.
The severity and impact ratings of risks were adjusted to better reflect the context of this assessment. Mitigation strategies for T3 and T4 were also revised (as R4 and R5) to better align with current development and testing practices.
Finally, new risks (R6-R10) were added to the updated document to address risks specific to assessment 2, including risks related to inherited code complexity, issues related to testing, CI builds, and gameplay quality.

Bibliography:

[1] A. Garcia-Dominguez (2025, Sep. 15). Continuous Integration [PowerPoint Slides]. Available at: https://vle.york.ac.uk/ultra/courses/_116044_1/outline/edit/document/_6367049_1?courseId=_116044_1&view=content&state=view

[2] GitHub Docs (2024, Sep. 07). *Building and testing Java with Gradle*. GitHub Docs [Online]. Available at: https://docs.github.com/en/actions/tutorials/build-and-test-code/java-with-gradle
[Accessed: Dec. 1, 2025].

[3] D. Kovolos (2025, Sep. 15). Requirements Engineering [PowerPoint Slides]. Available at: https://vle.york.ac.uk/ultra/courses/_116044_1/outline/edit/document/_6367043_1?courseId=_116044_1&view=content&state=view

[4] Cohort 1 Group 9 (2025, November). Requirements Document. Available at:

[Accessed: Jan, 9 2025].

[5] Cohort 1 Group 9 (2025, November). Architecture Document. Available at: https://eng1-group9.github.io/website/Arch1.pdf
[Accessed: Nov 2025]

[6] D.Kolovos, *Project Planning and Risk Management*, ENG1 lecture slides , University of York, Oct. 2025
https://vle.york.ac.uk/ultra/courses/_116044_1/outline/edit/document/_6367042_1?courseId=_116044_1&view=content&state=view