<u>Changes to Requirements</u>

After reviewing the feedback from the first assessment, the requirements for the project remained unchanged at the core as the architecture defined allowed us to sufficiently meet the initial requirements of the client. Therefore all changes further from there were made either due to the complexity of their implementation or due to the client requesting certain changes. As part of the risk mitigation document, requirements that were difficult to implement were negotiated with the client and kept,removed or changed based on those discussions. There were however, quite a few changes to the nonfunctional requirements as the original versions of them were not measurable/quantifiable.

(Underlined words are additions, Strikethroughs are removals.)
The major requirements removed from the project were to do with the visibility in different rooms:
- Props will only be visible whilst in the same room as the player or for a short period after they have left the room
- They will only be visible whilst in the same room as the player or for a short period after they have left the room

The complexity of implementing these particular requirement was far too great as it interfered with the way the library generated the game loop. Two mitigation techniques (one for potentially poor library choices, the other for changing game requirements) were discussed with the team and then a meeting was carried out with the client before removing the requirements.

The next requirements modifications were for visual elements of the game.
- Murder-Mystery at a Lock In Costume Party [~~in a early 20th century setting~~] where the party is still taking place despite the murder

This requirements were changed so that the project could have a unified theme (while also not limiting ourselves to one art style) across the project which fits the core requirements of the project. There were additional very minor requirements that were slightly edited to suit this unified theme.

The following nonfunctional requirements were changed in order to make them more quantifiable while the method of collecting these requirements was specified as well.

<u>Nonfunctional requirements regarding users will have their data gathered from the users after they play the game through the means of a questionnaire</u>

- The game must be deemed to have a <u>unified story theme</u>[~~art style influenced by pixel art with a sepia hit~~] by at least 80% of users <u>who have played the game repeatedly.</u>
- The game must be deemed to have a [~~cohesive~~] <u>dynamic</u> narrative by 80% of users who were able to play the game <u>repeatedly</u>.

These were changed so that the requirements could actually be achieved instead of leaving the procession of them ambiguous (while also not limiting ourselves to a single art style as written above).

<u>Changes to Planning and Method Selection</u>

The general method and planning listed in the document was followed and left reasonably unchanged, as it was well suited to our team (A team meeting was carried out about this within the first weeks of Assessment 2), with the only additions being the extra tools (Tiled, Slack) that were required as part of the development of the project, and some modifications to the planning being required due to availability of team members as well as the client over the holiday period.

Although the plan of using a Scrum inspired development process was followed,it was not possible (and feasible) to keep in contact with the client every week as listed in the process due to a combination of personal obligations and holiday period planning for the team. Therefore since communication with the client was at a lower-than-expected level, the team switched to contacting the client on a weekly basis once the holiday period had ended to ensure the project was set up as per the requirements of the client and if any new modifications needed to be added. This allowed the team to keep the requirements updated while showing the client a version of the project to ensure that it was inline with the client's vision of the project.

Communication between our team was found to be lacking with the team using only Discord as a communication client so there was a need to ensure that everyone was updated with the development of the project. To do this, we implemented the use of "Slack", another communication client. Slack allowed us to communicate effectively as part of our team and allowed us to split off into separate sub-teams for different classes and modules while developing the game. Additionally the use of github-integrated bots in Slack allowed us to be notified when significant changes and merges were made to the central repository. This additional communication line gave us a huge speedup in the time taken for a team member to review the changes to the repository, allowing the team member to point out errors and fix them or to point out potential improvements that could be made, before a change was merged into the central repository, leaving us with a more robust version of the project.

Other minor changes included the use of Tiled, which was necessary in order to create the background map of the project, and the use of CC0 (no copyright) assets in order to facilitate faster development of the project, while making the game look aesthetically pleasing.

The original plan of the Scrum inspired development process was followed as it was suited to the flexible workflows that the team members required. The process of having weekly meetings and allocated tasks allowed continued development over the holiday period despite the family obligations and time zone differences that team members had to face, making it a very suitable and robust development process for our team. The team organization of fluid roles was left unchanged except for the addition of subgroup leaders (for the front end and the back end) so that quality control could be scrutinised and the allocation of tasks could occur in a more organised way.

Changes to Risk Assessment and Mitigation

The changes to the risk assessment document were mainly in reassigning risk ratings for risks that had affected us during the development of the project.

However, there were some particular additions to technical risks which originated multiple times across the team from working from the project repository:
- Repository syncing errors
- Game design (architecture) interfering with certain requirements

The team felt that some risks and their risk ratings were previously underrated (i.e. their impact was more than expected) and changes were made so that each risk was appropriately rated. Stakeholder and Data risks were not changed as the team felt the initial rating of each risk in those categories were appropriate.

Risks affected :
- (Team) Team member being ill
- (Team) Team members not coming to meetings
- (Technical) Error in time estimation for programming
- (Technical) Bugs in software tools

Notable changes were made to the risk assessment document for the technical risks section. Specifically, the team decided to modify the risk rating for "Error in time estimation for programming" from medium to medium/high, and the risk rating for "Bug in software tools" from low/medium to medium. This was done as a result of certain aspects of the programming taking longer to complete than expected, creating a ripple effect that hindered tasks further down the development process.

Additionally, mitigation techniques were changed according to the team's requirements, the key example being where weekly meeting reports were added to the communication clients so that team members could be kept updated on the development process and prevent risks that stemmed from misinformation.

The team felt that prevention of risks was generally was not up to the level that was expected from the team, so risks which had a priority on preventing them had more techniques assigned to them or there was a shift or priority on to mitigating them.

The team felt the model of measuring risk ratings was notably effective in allowing development of the project to be a smooth process. This was because defining mitigation techniques for each risk proved to be very useful in assigning resources to mitigate them thus helping maintain the development process.

Risks and mitigation techniques that were only appropriate for assessment stage 1 were removed from the document.