Summary of Changes

Method Selection and Planning

As we continued onto milestone two, we believed our plan remained effective, therefore requiring very few changes. The small alterations that were made are briefly summarised and justified below.

As the project progressed, and we gained more experience with the development tools we chose (Unity), we discovered a fairly steep learning curve. During the first few attempts of using Unity, we realised that the tasks needed to complete our project would have to be divided between multiple team members due to a large amount of knowledge needed for each, thereby creating more solid roles within the team focusing on tasks such as special effects, graphics and game management. Once these tasks had been allocated, we decided that finding and using documentation was the best way to learn and use the tool. We can justify this allocation of responsibilities which differs to our previous plan of having flexible roles, as it relates to our breakdown of the project for development and allows a greater level of competency in each field of Unity.

We also found that Unity and our team management tool, Gitlab, were much less collaborative than we first believed. This led us to allocate the task of researching how to link the two tools, in order to allow all of us to work on the project. A document was then drawn up to describe the most efficient method for the team to follow in order to achieve this using git and the passing round of code responsibility. We justified this change to the plan because the use of Gitlab is vital in our team management method, to allow cooperative work, so this tool needed to be used as effectively as possible.

Our method of allowing the natural uptake of responsibility has worked well, but we did realise some of the disadvantages of taking on this approach. These include not being able to allocate risk ownership, unequal level of knowledge on a task, unequal allocation of work, preventing sharing of tasks and reduction of time spent working efficiently due to task leaders not being allocated immediately. We have altered our method slightly by allocating leaders immediately to tasks where they have experience or interest. However we wanted to make sure that there was still an aspect of flexibility, and that anyone could help on a task if it required more resources and attention. We justified this change to our plan because there was a large workload in this milestone and we needed to work efficiently and track our progress thoroughly, therefore, allocating leaders to tasks would achieve this.

In terms of our method of approach to this project, we altered very little. We did, however, notice that there were more aspects of the Scrum methodology that we needed to apply to our project. For example, we realised the usefulness and effectiveness of tracking our progress throughout the project, which can be implemented in Gitlab through the use of milestones and issue boards. This appeared to motivate the team more, and allowed focus to be put on certain tasks which bared more weight. Also, we found that shorter meetings were very useful and suited the team members individual timetables better. We decided on having two shorter meetings during the week-long sprint, with one longer meeting to go through tasks that needed more attention or help from other team members. To justify this, we considered the amount of work done during meetings, compared to outside of meetings. We worked out that we used group meeting to discuss what tasks have been completed and the next steps that needed to be taken, but during most of the meeting, no tasks were actually completed. Therefore having shorter meetings to catch up on progress allows more time outside of meetings to complete tasks, and any tasks that need to be completed as a group can be done during the longer scheduled meeting.

Risk Assessment

Our risk assessment was mainly still appropriate for our project as we continued into task 2. The project team found the layout very easy to understand and all the columns provided valuable information regarding the risk. The mitigation actions were clear which aided in reducing the number of risks which may have materialised. For this reason, we decided to keep the majority of the risk assessment document the same as in assessment 1.

One of the main issues that did arise within the risk assessment was the lack of ownership in the risk specification table. It was sometimes difficult to establish who was responsible for implementing the mitigation actions. This was done by design to fit with our fairly flat hierarchical system, however, after consideration and discussion by the project team and also based on the feedback we received and reviewed, we decided to include risk ownership in the risk specification table. We would still keep the same hierarchical structure within our team for the most part, but we would also assign people titles which would be referenced in the risk specification table.

The second issue which we felt needed changing in the document was the lack of a clear protocol for monitoring and updating the document. We decided on a method to ensure the risk specification was constantly up to date and correct throughout the project. The risk ownership column aided us when doing this as we could state whose responsibility it was to monitor existing risks and which team member should be consulted when new risks arose before they are added to the risk specification table.

We decided that all risks should be included in the current risk specification table in the same format as the current risks to provide continuity with the table. We felt the lack of monitoring and updating was an issue that needed addressing as the team was not too sure whose responsibility it was to update the table and it was mutually agreed that any updates were done on an ad-hoc basis. After reviewing the current process, we agreed that this could be made more efficient by having a protocol in place, which was more feasible to implement after assigning risk ownership within our table.

Another reason for implementing this change was due to the fact it was highlighted in our feedback that we had no clear way of monitoring risks throughout the project. With a protocol in place for this task written up in the risk assessment document along with ownership in the risk specification table, this provides a much clearer approach to this task which will be important throughout the project lifecycle.

Requirements Specification

As the brief for our game remained consistent, as the client hadn't added or altered any requirements and there was no scope creep, the core requirements for our product remained the same. During a team meeting, we decided as a group that the format of the requirement document was still suitable as we continued through assessment 2, as it had all the information we needed and displayed it in a table which was quick and easy to understand and reference. During different sprints, we did find that there were some issues with individual requirements which needed changing.

One of the issues with our requirements was that there was a certain amount of design pollution within the document. So rather than just constraining the functionality of the product, we were also stating how it was to be designed. This is bad practice as the design of the product may change whereas the core functionality should remain consistent. We updated some of the requirements and the fit criterion to remove the document of this design pollution.

When initially writing the requirements document, we tried to be rather specific so that it was easy to follow and there were no vague requirements. Unfortunately, upon reviewing the document and trying to meet all of the requirements we realised as a team that we may have been too specific and over-constrained certain requirements. We updated all the of these requirements so that they were still specific but not over-constrained and potentially difficult to implement. For example instead of stating how the winning condition should work, simply stating that the program should have a winning condition instead.

Another issue we found which we felt needed updating was the difficulty of testing certain requirements. We discovered this after reviewing the document and also moderating feedback from assessment 1. We found that this issue affected qualitative requirements, so we attempted to alter these so that they were testable, or simply remove the requirements from the document. During this stage, we also realised that some requirements were difficult to test because they were actually multiple requirements in one. We went through our table and split into separate rows all the requirements that may be interpreted as multiple requirements, and then proceeded to further refine each individual requirement.