Change Report

Cohort 1, Group 5
Team Name:
JAzZ MoLeS
Group Members:
Sophia Taylor, Lucy Wood, Mitchell Gilbert
Jamie Creed, Archie Adams, Zayed Iqbal

Summary:

Having taken over this project from group 9, we needed a concrete plan to make, keep track of, and review changes to the assessment 1 deliverables, documentation, and code we inherited. We designated each team member two or three sections each, with no sole person covering an entire section themselves.

Any changes made were communicated to the team, using Slack, in the respective channels. More specifically, code updates were made visible to everyone through ‘Git-Commit’ messages.

Every Thursday, during our group meetings, we would review any changes we had made and assess as a group what changes still needed to be made, and we would update the gantt chart accordingly.

Requirements:

In addition to inheriting the work from team 9, we were also assigned 2 new deliverables, as seen below:

- A leaderboard with the name and score of the top 10 people who have completed the game successfully.
- Achieving streaks. For example if the student goes for a walk every day during the week, they get additional points (on top of those they would get anyway for doing this activity).
    - These streaks will be visible in the game over screen as hidden achievements, for example "Jogger" for the walking one! Other ones could include "Bookworm" if they go to the (optional) library at least 4 times that week, "Duck duck go" if they feed the ducks at least 6 times, etc. Your game should include at least 3 streaks/achievements.

Due to these changes in deliverables, 2 new user requirements and two new functional requirements were added:

- UR_leaderboard_system
    - The game must include a leaderboard which a user can check
- UR_achievement_streaks
    - The game must include hidden achievements the user can earn
- FR_score_tracker
    - The game must keep track of the top 10 highest scores
- FR_streak_tracker
    - The game should keep track of activity streaks

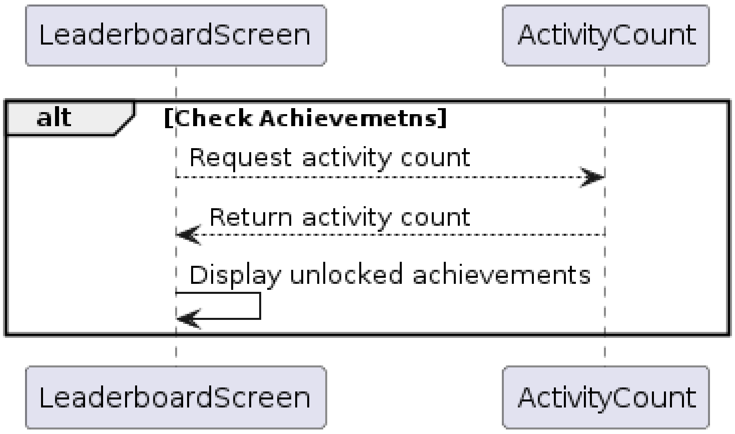
Some user requirements were also changed from 'Shall' priority to 'Must' priority. We felt that 'shall' didn't accurately represent the importance of these requirements.

- UR_map_movement
- UR_time_limit
- UR_activity_places
- UR_score_system
- UR_energy_system
- UR_energy_consumption
- UR_load_time

Similarly, UR_load_time had its priority reduced from 'Should' to 'May'. This is because load times were not an immediate concern for us, and we feel if it is an issue for users, it can be improved at a later date.

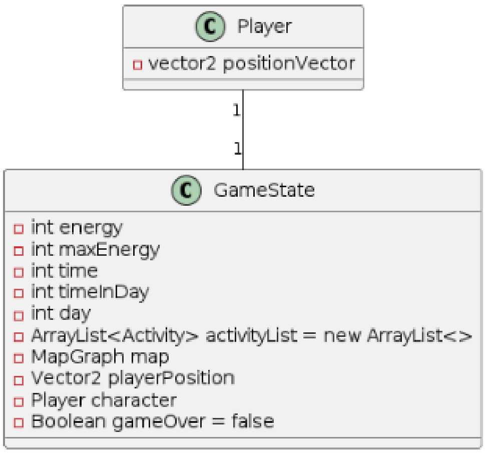
Architecture:

With these new requirements, the sequence diagrams and class diagrams had to be updated.
One of these new requirements was the addition of ‘hidden achievements’. This meant that the user had to be able to complete certain goals to unlock these achievements. Implementation of this required an activity tracker to be implemented. This tracker would count how many of each activity are completed and in turn could be queried when checking for certain achievements.
Below is a sequence diagram for how this interaction would take place:



LeaderboardScreen is only referenced as this is the screen where the achievements will be displayed.

In addition to this, much of the game's general architecture was reworked to make it easier for us to use, and for any future developers to expand on the game and potentially make new maps or rework existing features. One of these changes was with how the player's position is tracked and calculated. Originally, the row and column was stored in an object called TilePosition and used getters and setters to increment the values. This was very inefficient as 3 different operations had to take place (*get, increment, set*) to simply add 1 to a value. We overhauled this, and stored the player’s position in a Vector2 - which is an inbuilt object. It allowed us to simply just add a movement vector to the position vector, which just took one operation. It also meant that when seeing if a tile is in the map, there is no need to loop through every tile as we can just do *.contains(playerPosition)* which makes the code much more readable. Below is the updated section of of the UML diagram



Method selection and planning:

Selection and Justification of Software engineering methodology and development tools

- Since we chose to continue and use the same agile development methodology, there was no need to change that part, but simply to add onto the reasoning as to why we feel it is the most effective strategy for our group.
- In terms of the tools we utilised, we chose to use slightly different applications, so these were changed accordingly.
- We wanted to keep this document concise and easy to read, so some parts of the documentation have been cut from this new document.
- We also used git so the documentation for reasoning about git was not needed to be changed since out reasoning was very similar
- We also used github pages, but our reasoning was slightly different so that was changed accordingly
- We did not consider eclipse as an option for our development, it was between vs code and intelliJ so the last section was edited to reflect this

Approach to team organisation

- In our current group, we have six, not seven members
- Decided to add in our own approach to do with initially giving roles to people, then changing that as we needed something more flexible.
- We didn't necessarily have designated time for meetings per week due to peoples schedules, so as a result of this we organised when the meetings should be between us-this has been changed and communicated through the documentation.
- We chose different methods in terms of identifying and allocating tasks to our team members, so this section was edited slightly to represent that, as the sentiments of the last group were very similar to ours in terms of the reasoning of choosing this methodology
- We went through the work individually and did a big read through before submission, so the workflow is different
- We didnt hold a retrospective as that was not the method we had decided to go with, however the online meetings we held were mentioned earlier on in the thing

Systematic plan for the project

- For the first part of the systematic plan for the project, there wasn't any need to change it, as that is to do with the work in relation to the first assessment, which is unrelated to the second half. So what would be best is to simply continue the documentation for the second assessment in the project

Referencing

- The references were also adjusted accordingly

Risk assessment and mitigation:

We chose to edit part of the risk management process description, as we assigned risk owners randomly based on to whom the risk was relevant, as opposed to purely random selection.

For the risk register, we chose to combine the avoidance and mitigation columns, as they often contained the same strategies or had only one column filled, making two columns redundant.

We chose to split the risk about team members being unavailable into two risks, since the severities and mitigation strategies for someone being unavailable short-term and long-term are quite different. We also added risk R12.