Method Selection & Planning

Group 8 Members

Zac Challis

Viktor Atta-Darkua

Dandi Harmanto

Harsh Mohan

Danny Burrows

Part A

Software Engineering Methods

Our chosen software engineering method was agile, after the first assessment we opted to have more frequent meetings so we could adapt to changes in circumstance more easily, a lack of adaptation was one of our groups shortfalls in assessment one as such a focus on this would allow us to work more efficiently and only spend our time working on the most valuable assets of the project.

We organised frequent team meetings, on average twice a week. These meetings served two key purposes:

- Providing a platform to discuss and agree upon key development decisions, most often to review requirements.
- Allowing each team member to communicate their progress on their assigned tasks, and for new tasks to be assigned when necessary.

When creating the initial “road-map”, we chose not to pre-emptively assign any tasks to particular people, instead we assigned tasks throughout the project whilst ensuring all members had a chance to work on all areas of the project to keep the workload more even as opposed to the situation in Assessment 1. Assigning tasks throughout the project was particularly appropriate for a team without large amounts of software development experience as our estimations for the time and effort needed for each task weren’t perfect, therefore this system avoided any particular team member being assigned tasks at the start of the project which transpired to be an unfair portion of the workload. The frequent team meetings in which we discussed task progression, however, allowed us to keep each other accountable for our productivity.

Tools Used

Communication and Collaboration

- For our team meetings we used Discord. Everyone was used to this method from Assessment 1 so keeping it the same made the most sense.
- We used a Github Project board in order to organise and keep track of our tasks throughout the project. Alongside the team meetings, this was crucial in keeping us organised by providing a visual representation of tasks that were completed, in progress and yet to begin.

Website

- We used GitHub pages to develop our website. This meant that our resulting site would be simple to replicate for any team which chose to take over our game for assessment 2.

Architecture

- Draw.io was used in order to create the abstract architecture diagram, as this relatively easy to use tool seemed more appropriate for this simpler diagram
- PlantUML in addition to adobe photoshop was used in order to create the concrete architecture diagram;
    - PlantUML was used at first in order to create representations of classes, categories of classes and the relationships within each category

Implementation

- We chose to use IntelliJ as our IDE. This was due to both the ease of use offered by the tool, along with the fact that we had experience with the tool from assessment 1: it felt like the ideal choice to avoid unnecessary and time-consuming learning curves.
- We utilised the libGDX game development framework during the implementation of our game. Similarly, to our choice of IDE, this was largely influenced by the previous experience of the team and obviously that of the previous group we've inherited the code from.

Alternatives considered

- We considered using other IDEs such as Eclipse for software implementation, however as mentioned previously we chose to use IntelliJ due to team members having previous experience with this IDE

Part B

Team Roles

During our initial team meeting we discussed assigning the following team roles. The structure of team roles was far less formal than in assessment one. We already had a structure of team roles figured out from the first assessment and as such followed on with that configuration with a few tweaks. The major adjustments made from assessment one were that we wanted to ensure every team member had a chance to contribute to every major section of the project. To ensure this we kept roles officially less defined, and focused instead on roles in terms of contribution; with every member taking responsibility for a section of each part of the project, some testing, some documentation, and some feature implementation.

To aid in our hybrid method of group organisation we would constantly review group roles in our weekly meetings to ensure we could adapt to any sudden changes or challenges with members abilities and such. This way our dynamic would evolve throughout the project and eventually would settle on a natural order.

Task Assignments

Task assignment took place throughout the assessment process as detailed in part a of this document. When the time came to assign new tasks, we tried to keep these assignments in line with each person's particular skill set in order to ensure efficiency as such throughout the project members mainly focussed on the particular aspects which they felt most comfortable with, usually due to previous experience.

Part C

Intro

Our first step in planning this assessment was to create a kanban board on GitHub. We used this board to keep track of assignment of tasks and times of completion/deadlines. We found this to be familiar to our approach from assessment 1 and as such was a good fit seeing as early adoption of this tool was found to help with team productivity and decrease amounts of time needed in meetings to keep track of where everyone was in terms of progress with their respective tasks.

These snap-shots were created from a summary of the kanban board by taking a condensed version of the task breakdown table and colour coding the tasks which were due to be done at this point in the project. With the following colour connotations:

- Red – Not started
- Orange – In progress
- Green – Completed