Project Review Report

Team Management and Structure

Our team's structure changed considerably throughout the project. Initially, our structure was much more rigid. In the first assessment, we assigned concrete roles based on what team members felt most comfortable doing [1]. Once the project started, we found that we were most productive when our structure was more dynamic, as all members of the team had similar abilities in all sections of the project and a more collaborative approach made us much more efficient. We also found that using this approach kept all team members informed on as many aspects of the project as possible, rather than relying too heavily on specialists. However, we did not change the Scrum master throughout the project, as we felt he had a solid grasp of the Taiga platform and we feel our team communicated very effectively, which generally reduced the role of the Scrum master.

Our understanding of software engineering developed quite substantially throughout the assessments and this greatly influenced the dynamic structure of our team. For example, we initially believed implementation would require a much more substantial proportion of our resources than it did in practice. Fortunately, we adapted very quickly to this change and this was, in part, due to the importance of the planning process, which is something we began to put an increasing amount of effort into. Having a class diagram as a reference for implementation or utilising Gantt charts to enforce concrete task deadlines and quality checks made us work more efficiently and raised the standard of the work we were producing.

Similarly, we allocated a fairly large proportion of time to learning the technologies and platforms used by the groups whose projects we were choosing to adopt. In reality, the other team's projects were built on very similar platforms to ours, this was largely due to the programming language constraint applied in the specification which wasn't present in previous year's assessments [2]. This factor contributed considerably to the inflated importance of the planning and organisation processes and the deallocation of some resources assigned to implementation. This constant fluctuation in task priorities was mostly facilitated by our dynamic approach to team management, as we could rotate team members throughout the production line whenever our internal requirements deemed it necessary.

Development Methods and Tools

Our team employed the Scrum framework for Agile development throughout the project. We found using Scrum met our expectations and abiding by the three pillars of Scrum [3] made us much more adaptive and retrospective in our work. For each assessment, when we received the requirements we were able to plan our work schedules flexibly, sticking to and deviating from our Gantt chart where necessary. Throughout the assessments, we adapted our sprint lengths according to the tasks at hand and aimed for weekly scrum meetings with all team members present. During holidays, we found weekly briefings to be unrealistic due to differing schedules and so we started to make more substantial task allocations before the end of each term, allowing us to rely on communication apps and more concise updates.

We continued our consistent use of the Taiga project management platform, it provided all the functionality we required alongside a very accessible UI. All team members had access to their currently allocated tasks and could post completed work for review, which is a feature that encouraged best practice quality assurance for every document we submitted. Taiga proved less useful for the implementation sections of the assessment and we didn’t believe the method of peer review via Taiga was as applicable to the code we were producing. Instead, we utilised the flexible nature of Scrum to organise semi-regular code reviews to maintain code consistency and quality.

We used Facebook Messenger very regularly for arranging meetings, updating on individual progress and reminders for important project events such as presentations. Using a more informal communication medium alongside Taiga and Google Drive (for document sharing) was a combination that we found suited our group and the nature of the assessments really well.

In terms of development tools, our selection has changed considerably since the first assessment. Once we had decided our game would be 2D, we had to make a switch to the LibGDX platform and began to learn about the structures and objects that make up that particular game engine. Once we had established the art style of our game, we used Piskel to develop many of our game assets, as well as using Tiled to develop the various game maps. Due to the similarities with the games we worked on, these tools were adequate throughout all of the assessments which was very useful, the details of these similarities are further explored in our assessment three change report [4].

References

[1] Shaun of the Devs, ‘Method Selection and Planning’ [Online], April 2019, Available: https://lloydbanner.github.io/SEPR-Team-7/Plan1.pdf

[2] University of York Department of Computer Science, ‘SEPR Assessment Structure’ [Online], April 2019, Available: https://lloydbanner.github.io/SEPR-Team-7/SEPR%20Software%20Engineering%20Project%20-%20Team%20Assessments%201.pdf

[3] Scrum Guide, ‘The Scrum Guide’ [Online], April 2019, Available: https://www.scrumguides.org/scrum-guide.html

[4] Shaun of the Devs, ‘Change Report: Changes to Methods and Planning’ [Online], April 2019, Available: https://lloydbanner.github.io/SEPR-Team-7/Change3.pdf