

Sea of Geese

Method Selection and Planning

Software Development Methods

During our initial discussion about the SEPR project, we decided that we should choose a specific development methodology to work with as it would make sure that we know exactly how we are going to develop the game. There are several different development methodologies that are designed to cater for different development styles and group sizes, so we had to find the method that would work best for us.

During our research, we concluded that the best option for us would be to use the Agile method as it fits our situation perfectly. It is best used when the development is more customer-centered which, as we are producing this for customers and following their requirements, is perfect. It also “encourages teamwork, self-organization and accountability”[1] which is incredibly beneficial when working in a small group.

We also looked at several different models such as the waterfall method, but we found that it was too inflexible for us to be able to work with as it would require us to extensively plan our software before we started development [2]. As we are on a fairly short time frame, we thought that Agile methods would give us the most flexibility when working with changing requirements and any potential problems.

We have decided to adopt the Scrum implementation of Agile [3]. There are three main roles in this methodology. The first is the Product Owner. They set the project requirements and communicate the priorities to the team. Secondly, there is the Scrum Master. They are not a project leader in the traditional sense, but instead make sure that the Scrum methodology can be followed and that distractions are dealt with. Finally, we have the Development Team. This is a self-organising group that decides how to split up the tasks that have been given to them.

The actual methodology proceeds as follows. Firstly, there is the “Backlog” which is a list of all requirements for the project. It is changed throughout the project duration.

Work is organised around several short “sprints”. At the beginning of the sprint, the team meets to plan what they hope to achieve during the sprint. During the sprint, the development team meets regularly to track the progress that has been made toward the sprint goal. At the end of the sprint, the Backlog is updated accordingly, and the team reviews what went well, and what didn’t. After this, the next sprint begins.

During our project, between team meetings, we shall use two main methods of communication - Facebook Messenger, and Discord. Facebook Messenger is useful to quickly get in contact with other team members if a meeting needs to be quickly arranged or something needs to be quickly communicated. For more formal communications when team members have access to a laptop or desktop computer, Discord is more ideal. It allows for us to organise our communications into channels corresponding with the different assessment tasks.

We shall be programming this project in the Java programming language. To aid the development of a game, we have decided to use the “libGDX” game development framework [4]. This will allow us to conveniently handle the rendering of images, sound effects, and music.

As with any project of this nature, a version control system is important. Git was chosen by the team as it is the system that most members of the team were familiar with. A service to host our repository was needed. Github was chosen due to some members having used it before. Our university email addresses entitle us to free private repositories which is also an advantage.

While Github provides a good place for hosting our code repository, for documentation we have decided to use Google Drive and Google Docs. This allows for us to have a shared storage space where we can collaborate in real time on editing our documentation, as well as being a storage space for all sorts of miscellaneous files that don't belong in the Git repository.

For the actual coding, we initially used Eclipse as an IDE, as we had all used it in the previous year and were familiar with it. However, it soon became apparent that the configuration on the university computers would cause problems when we tried to use libGDX with it. Therefore, we had to look for another IDE. We decided on the IntelliJ IDE as it was present on the university computers, and it worked well with libGDX.

Team Organisation

For Assessment 1, we adopted a paired approach - each pair would be assigned tasks, and the pair would then help each other to achieve the relevant goals. The pairs initially assigned to each task were as follows:

- Requirements: Jordan Cameron, Spencer Atkins-Letten
- Architecture: Jiahao Wang, David Norman
- Risk assessment and mitigation: Ben Hassell, Tia Briggs

In addition, the following roles were assigned:

- Website maintenance: David Norman
- Meeting records: Jordan Cameron

Team members were assigned to method selection and planning once they had either finished or made substantial progress with their assigned areas.

For Assessment 2, we assigned the following team roles:

- Meeting chair: Benjamin Hassell
- Development: Jiahao Wang, Spencer Atkins-Letten
- Documentation: David Norman
- Game design: Tia Briggs
- Graphical design: Jordan Cameron

These roles were flexible, and members helped out in different areas as required. For example, more members helped with documentation once the main game development was completed.

For Assessment 3, all members initially worked on game development in two groups of three:

- Minigame: Spencer Atkins-Letten, Jordan Cameron, David Norman
- Updating main game: Tia Birggs, Benjamin Hassell, Jiahao Wang

The initial Scrum sprints were to complete the implementation of those two aspects. After both were completed, successive sprints involved individual members being assigned to development tasks. In the final sprints, individual members were assigned to finish various aspects of documentation.

Plan for Assessment 4

We have judged our methodology for Assessment 3 to be successful as it allowed us to achieve milestones at a far quicker rate than we did at Assessment 2. Following this, we intend to adopt a similar methodology for Assessment 4.

Our plan for Assessment 4 is as follows:

1. Create presentation for our Assessment 3 submission and present in practical session.
2. Meet as a team to decide which product to select.
3. Meet as a team to consider new set of requirements and divide development responsibilities among two groups.
4. Weekly Scrum sprints to finish development of product.
5. Shorter (two-day) sprints to write documentation and relevant material. (First half of 5 is concurrent with second half of 4)
6. Upload submission.

This [Gantt Chart](#) [5] provides an initial abstract overview of assessments 1 to 4, I've included shading with darker meaning a more intense workload, the tasks are split up into work packages for each of the four assessments, it keeps abstract so we are more focussed on getting each work package done within it's time frame rather than focussing on the smaller parts within it however for assessment 2 at least, the individual tasks are covered in detail by the critical path diagram below.

We created a [Critical Path Diagram](#) [6], this shows us how tasks depend on each other within the second assessment, showing us how to progress, though there are only 11 weeks total as shown in the Gantt chart, we can see that the total for the critical path is 24 weeks, this is because several of the tasks can begin and progress at the same time, one week tasks tend to be setting up URLs and such.

Bibliography

- [1] Cprime. *What is Agile? What is Scrum?*, [Online] Available: <https://www.cprime.com/resources/what-is-agile-what-is-scrum/> [Accessed: November 6th, 2018]
- [2] TPX Manifesto. *Agile vs Waterfall: Comparing project management methodologies*, [Online] Available: <https://manifesto.co.uk/agile-vs-waterfall-comparing-project-management-methodologies/> [Accessed: November 6th, 2018]
- [3] Scrum Alliance. *Learn about Scrum*, [Online] Available: <https://www.scrumalliance.org/learn-about-scrum> [Accessed: November 6th, 2018]
- [4] libgdx. *Goals and features*, [Online] Available: <https://libgdx.badlogicgames.com/features.html> [Accessed: November 6th, 2018]
- [5] Gantt chart: https://davidjnorman.github.io/SEPR/assessment_1/GanttChart.pdf
- [6] Critical path diagram: https://davidjnorman.github.io/SEPR/assessment_1/CriticalPath.pdf