Method selection and planning - Plan1

Dragonite
Team 21

Omar Omar
Rhianna Edwards
Okan Deniz
Craig Smith
Omar Galvao Da Silva
Joel Wallis

AGILE Methods, development and collaboration tools

Agile Methods

Followed modularity

- Group members worked on their own assigned section of the code and project that were later merged together. This allows for more flexibility in the development of the project and makes sections easier to review.
- Weekly meetings to decide on changes that ne
- ed to be made for a set deadline but also develops a to-do list to focus the progress for the week down to a few points.

Review based iterations

- Once a task has been completed by the team working on it, instead of checking it off the group as a whole will be asked to review the work and make sure that it fulfils the criteria but also provide feedback on the task. Afterwards, the team working on it will implement the changes outlined by the group and it will be put for review again.

Involvement of the client in design

- We plan to try and get a demo in the hands of the client so that we may be able to get feedback from them and try to create an experience that better fits their vision. As a team we know that this is unlikely to happen due to time constraints but still wish to achieve this as it would be beneficial to the project.

Development and Collaboration tools

Github

- We are going to use Github as our shared file space and version managing software to carry out this project. Firstly, it is the software that the majority of the team is familiar with so it will take less time than finding a different software and having to learn how to use it. Secondly, due to its branching program feature we can work on the code in a modular fashion then combine these modules together in the master branch which fits the AGILE methodology we will use in this project.

IntelliJ

- When choosing the IDE we knew it would have to have support for the version management software we were going to use. IntelliJ has support for Github as we needed however, some of our members had difficulty getting this to work so another piece of software was used to aid this.
  This IDE comes with inbuilt Gradle support and as such made it the choice for us to program the project in. More details on this can be found in the Considering Alternatives documentation.

SourceTree

- To solve the problem that some of our members were having with IntelliJ's Github integration we chose SourceTree. This will make it easier for some of our members to start programming and updating the project with files.

Zoom

- This software will be used to have formal meetings with the group and work together on the project. It is a professional style software and such should lead to less distractions than alternatives.
  However, it does not have any permanence once the meeting has finished and as such we have chosen to use an alternative alongside Zoom.

Discord

- Even though discord is a less formal software, we have decided to use it to host text conversations, video and image links as well as to better understand our fellow team members. The permanent text chat allows for ideas to be retained and reviewed as well as resource sharing between team members. We don't plan on using it for meetings and to continue using Zoom for meetings

PiskelApp

- As per the requirement interviews, the client will prefer custom made graphics. So we have chosen to use PiskelApp for producing graphics. It is new user friendly unlike alternatives and as such will speed up the project. Furthermore, it is free to use.

Hiero

- This tool was used to generate the font files required for the game. The font files needed to be in a ".fnt" format and this was one of the only free up to date tools we were able to find.

MAGIX Music Maker

- This tool was used to make the music for the game. It has a variety of musical instruments which was helpful.

Considered Alternatives

Discord Vs Zoom

We have decided to use both softwares to best work on the project.

Zoom had to be used once a week for meetings with the University

- We decided to host all meetings on Zoom to be consistent and have continuity between meetings

Why we chose to also use discord has previously been stated.

VSCode Vs IntelliJ

- IntelliJ has inbuilt Gradle support which made it faster to begin programming the game compared to VSCode which some of our members tried to get working however, it was taking too long and so thought it is best to work with IntelliJ

PiskelApp Vs Photoshop

- When deciding on software to use for the graphics of the game we considered two main programs, PiskelApp and Photoshop. Photoshop is the more professional of the two however, we found it was more difficult for new users to pick up, which would have slowed development, and the limited time we could use for free made it unreliable.

  On the other hand, PiskelApp was new user friendly but also always accessible to be free.

Hiero Vs Gdx Freetype

- Gdx Freetype had trouble with font scaling and required more time for its implementation so Hiero was used.

MAGIX Music Maker Vs Pro Tools

- While Pro Tools is a powerful tool. Prior Knowledge with MAGIX made it the chosen candidate.

Team Organisation

We took a Laissez-faire strategy to leadership; this allowed members to work on separate parts of the project in their own time and to best work to their own strengths. This will give us more flexibility while working on the project as some tasks become complete more people will be able to work on the tasks taking more time or are more complex. Smaller tasks won't become sidelined and left to the last minute while larger tasks won't be given to too few people and overwhelm them. Furthermore, for the larger tasks it will create team members more familiar with the task who would be able to guide members joining that task later on direction and help understand it faster.
The team does not have any members who explicitly want to be a leader and as such we decided against forcing someone into the position. This could have caused unnecessary stress on that member so we decided to split the responsibilities across the team. On individual tasks, members who have been working on them the longest will become de facto leaders as they have the most familiarity with the task and be able to guide the task to completion as other members move to that task.

We find that this approach is appropriate for the project as it aids in a modular development of the game but also speeds up development which is helpful with the amount of time given to produce the product. Having the increased flexibility will also benefit the project as if members find they are stronger in different aspects of the project than initially thought then they can move to different tasks without much difficulty. This ensures members can play to their strengths and keep the project going smoothly but also within the time restraints.

There are risks involved with not appointing a leader for the whole project. Firstly, this leads to more discussion about setting deadlines and reviewing work. More time will be spent on these sections as they are a group effort and won't have someone to definitively call and end to discussions or be decisive but this will be balanced with a calmer environment to work on the project in. There is also the risk of work that doesn't have shared access gets forgotten or not worked on as much due to little oversight; members may stick to their own tasks and have little interest in the tasks of others.

Systematic plan

Key tasks

The key tasks are listed below, this outlines everything that we will be doing in order to cover all parts of the project. It breaks down the major tasks into smaller ones. This allowed the team to easily keep track of our progress as we colour coded the key tasks once we started/completed it.

Website
- Submit URL
- Link all assessment documents to the website
- Ensure website is well structured

Requirements
- Write an introduction explaining how requirements were elicited and negotiated, why they were presented as they are.
- Evidence research into requirement specification and presentation
- SSON, how requirements are set why used a certain layout
- Table of requirements with environmental assumptions, possible risks and alternatives

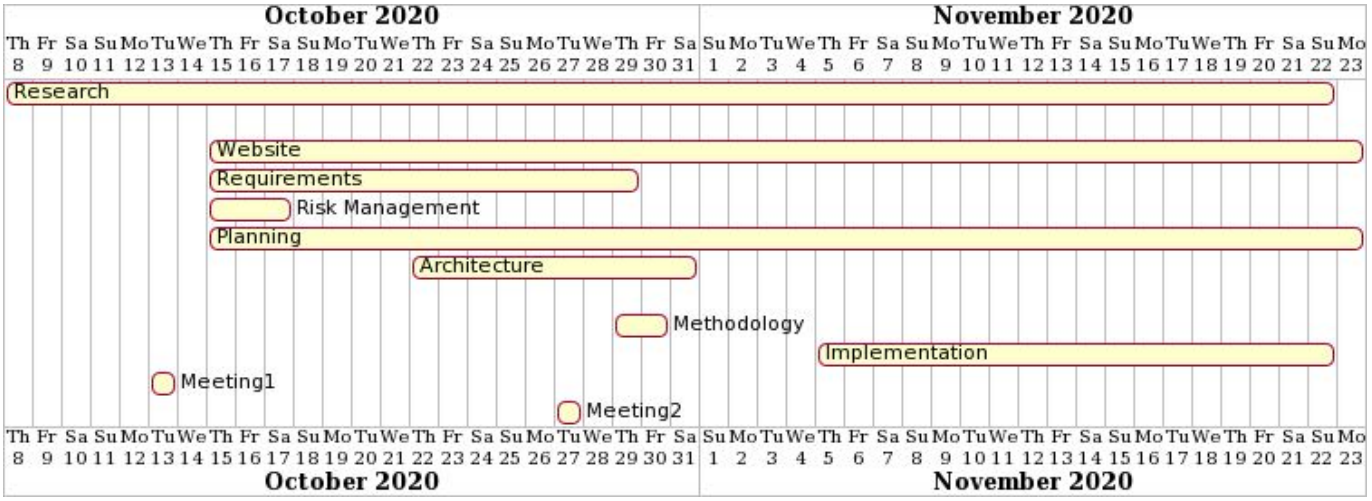
Architecture
- UML, descriptions of structures , a statement to describe the languages used to describe architecture and tools used to create architecture.
- Represent relationships between entities
- Justifications of the structures – how abstract -> concrete and how concrete architecture fits the requirements, consistent naming of constructs to provide traceability.
- Outline methods(agile), identify development/collab tools used and alternatives tools considered. All with justifications

Methodology
- Team's approach to team organisation – why is it appropriate for the team and the project.
- Systematic plan – gantt chart, critical path and task dependencies.
- Weekly screenshot of the plan on the website and discussion of how the plan has evolved.

Risk management
- Introduce and justify risk format and level of detail
- Table of risks, likelihood, impact and mitigation

Implementation
- Zip file / link to github of the documented code with an executable JAR file.
- State features that are not fully implemented

Gantt chart



The critical path is: requirements, architecture, implementation then website.
The architecture is dependent on the requirements. Then the implementation is dependent on the requirements, risk management and methodology. Finally the completion of the website depends on all the other tasks being completed as everything has to be uploaded into the website.

Plan evolvement

We set milestones for when we need to complete certain tasks for the critical path so that all the tasks were completed before the deadline.
When certain implementation tasks took significantly shorter or longer than expected, the plan was adjusted accordingly.
For example, an implementation taking way shorter than expected meant we could introduce a new implementation that was not originally planned to be included in the code. An example of this is the implementation of music into the game.
Whereas when certain implementation tasks took longer than expected, focus was either shifted towards their implementation meaning more time and effort were spent on it or the level of detail originally planned to be put into it was reduced.