Outline to Software Engineering methods and collaboration tools

We will use an agile approach to software development. Rather than have set stages for requirements, design, implementation etc we will allow ourselves to jump between the stages. There will be set times to instantiate each section, but if during implementation our requirements change then we will be able to go back and change these and continue from where we left off. This means our development process will be less limited which is essential as requirements and design can develop during later stages of development

For collaboration we are making use of the communication application Discord. On this application we have created our own server and divided it into different channels. For example we have a channel dedicated to work related to the project and another one for work from other modules. This helps us stay focused and organized so long as we stick by the different channels and don't allow their content to overlap. As well as enabling communication, both by text and voice, Discord allows us to share files with each other. This is especially convenient for sending, for example, diagrams of ideas for the program or pdfs of the project brief. There is a feature which allows certain messages to be pinned for easy access, this was especially helpful for important links such as the one to our github depository. We prefer using Discord for voice chat in comparison to alternatives such as zoom. Not only does it save our messages but it is generally smoother to use and less intrusive (discord doesn't display an overlap of the voice call on the screen at all times like zoom).

For organizing our documentation we have made use of Google Drive and have created a team drive which we can all collaborate on. This is especially useful as Google Drive allows live collaboration on services such as Google Docs. Although the documentation will all be uploaded to the website, it is easier to store it on Google Drive during early development when lots of changes are being made as we can make these much easier and don't have to reupload new versions like we would to github. The simplicity of making changes enabled by using Google Drive makes it perfect for following an agile approach to our software development.

We used the UMLet software to develop the UML for our project. Specifically for the architecture of the project. We used this because it is free, open-source and simple and easy to use. This made it perfect for our project as using a more advanced commercial UML software wouldn't be justified for a project of our scale. UMLet allowed us to make multiple object-oriented class diagrams, one abstract and one concrete, and sequence diagrams. This helped us design and understand the structure of the project which is vital to have when progressing to the implementation stage. We can clearly see the entities of our project in the class diagrams and the flow of the game in the sequence diagram.

Approach to team organisation

We haven’t selected an outright team leader, rather in different scenarios one will naturally emerge. When approaching the project we have divided up the work load into even amounts and, where possible, done this based on each member's strengths. In some scenarios we have worked on the same task together, for example in gathering and writing the requirements. However this isn’t always beneficial and in other circumstances it makes more sense for people to choose something to work on and do it themselves. Requirements engineering lent itself to collaboration as there were several types of requirements which could be split up and working on the same document wasn’t a problem. However in other circumstances, like write ups, more than one person on the same document would just cause confusion and people would get in each other's way. For this reason we decide based on the task what the best way to tackle it is.

In the circumstances where we each work on our own task, we encourage ourselves to check each other’s work and allow any suggestions and improvements to be discussed and possibly implemented. This makes up for a potential lack of collaboration which could come about from each working our own individual tasks.