Module: ENG1/ASSESSMENT2

Title: Method Selection and Planning

1. **Charles Stubbs**
2. **Annabelle Partis**
3. **Kieran Ashton**
4. **Yu Li**
5. **George Tassou**
6. **Alex Shore**

Software Engineering Methods and Collaboration Tools:

We used an agile approach to software development. This means we have the freedom to move between the different stages of development rather than having to conform to the strict order of requirements, design, implementation. We thought this would be the best method as we didn't have a clear scope from the beginning of how our game should look like at the end. It meant that during implementation, we could go back and alter our requirements if changes were needed, making the development process much less restricted.

Our team chose LibGDX as our game engine. We chose this engine as after doing some research, we believed it to be the best suited engine for this project. Firstly, the engine uses the Java programming language which we were required to use for our game. Also, LibGDX seemed like an easier framework to learn how to use. It does some of the more complex things from other frameworks, such as Sprite Batching, for you. It's a much higher-level framework than something like LWJGL. We felt that this was important given the relatively tight timeframe given for the project. Furthermore, smaller games made using LibGDX tend to use much less space than games made using other engines such as Unity.

We chose to use IntelliJ as our primary IDE when coding our game. This was primarily because it integrates so easily with LibGDX. However, there were other reasons as to why we picked it. It is very user-friendly, having in-depth code assistance and easy navigation. It also offers a good set of debugging tools.

Our team decided to use GitHub for version control. It easily allows us to share the code for the game amongst ourselves. It also makes it easy to track changes across versions of the code. We used GitBash to help us push and pull any new versions of the software as we needed, since it was explained in the lectures meaning the resources to learn how to use it were readily available.

For the organisation of our deliverable documents, we decided to use a shared drive within Google Drive which allowed us to all collaborate on each of the required documents. Google Drive was perfect for this project as it allows live editing within Google Docs. This made working on tasks together within our meetings possible. The fact that Google Drive makes it so easy to make changes in any part of the documentation process made it perfect for the agile approach we were using

The previous team used tiled to open .tmx files and create the map. Whilst we do have this installed and are prepared to use it if needs be. We aren't planning to due to the map's completeness.

For the pixel art sprites and other graphics, the last developer team used Adobe Photoshop as they were experienced in it's interface. We were not, so instead we chose to use paint.net and Medibang Paint as they are both free, intuitive programs.

We are not going to create any new sounds so we have opted to just use the ones provided by the previous team.

When testing we will use a combination of Tom Grill's GDx Runner, Mockito and Junit in order to create accurate tests that are easy to understand and run. We will also run Scenario tests, to cover tests that were not possible in the time constraints using JUnit.

As for communication within our team, we made use of Zoom to have regular discussions covering all areas of the project. We held one or two meetings per week and discussed what each member had been working on and what we would do before the next meeting. We then used slack to communicate any additional topics that either could not be fully covered, were urgent or could not be covered in the meeting, we also used it to assign roles using this method so they could be easily revisited if needed.

Team organisation:

At the start of the project, we decided to adopt a laissez faire style of management with no specific assigned roles. This was most appropriate because we did not know each other well enough at the time to assign roles relevant to our specific skill sets. Every sprint of the agile development cycle, each member of the team was assigned work to do. At the start of the next meeting we each reported our progress and, if necessary, the work would be reassigned.

During the initial change report and most other documents, we did usually brainstorm as a team as to what the best method would be to complete a document. We then assigned two people to a document to ensure that even if one person was unavailable then someone else could work on a document.

Systematic Plan:

Gantt chart on website provides iteration on the plans