Module:	ENG1/ASSESSMENT1
Title:	Method Selection & Planning
Team name:	ENG1_Boolean_BobCats

Team members

1. Adam Howard
2. Lewis Mcshane
3. Morgan Davis
4. Muhidin Muhidin
5. Roan Gibbons
6. Zijun Zou

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

When creating the game, we used Tiled Map Editor. It seemed to be the best option for creating a 2D level. The editor is easy to learn, has many helpful tools such as different brushes, objects, and the ability to add collisions. This can then be easily implemented into the game using the .tmx file.

For the pixel art sprites and other graphics, we used Adobe Photoshop since it has a very extensive set of tools that makes it really useful. As well as that, the team members who worked on the graphics already had experience with the Photoshop interface.

When creating the basic sound and music for the game, we used a free open source music editor called Bosca Ceoil, alongside Audacity for editing, since the team member who worked on audio had experience using both software. This made it a quick process since they did not need to spend time learning how to use any software.

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. As well as that, we used Messenger in order to schedule each meeting and raise any important advancements or problems. The app is very easy to use and although basic, provides us with everything we would need to be able to get in touch with our teammates.

Team organisation:

At the start of the project, we decided to assign roles for each member of the team. We used Belbin’s nine team roles as a guide. The roles went as follows: Zijun - Teamworker, Muhidin - Plant, Lewis - Monitor evaluator, Adam - Shaper, Morgan - Implementer and Roan - Completer finisher.

We approached the project by splitting the work in equal amounts and assigning a different section to each member. We tried to be fair to every member and where we could, decided who would be given each section based on their strengths. However, in the early stages of the project, most of the time we would be working together. Whether that was when brainstorming ideas for our game or writing up the baseline requirements. We decided that most of the deliverables lent themselves to having each person working on one at a time. However, we also knew that it was important for other members to look over each document to ensure that for each deliverable, everything had been covered and was to a high standard, rather than putting all our trust into the member assigned to that particular deliverable.

Systematic Plan:
Tasks:

T1.1:
Create a website and github domain.
Start date: 21/10/20

T2.1:
Read through the product brief.
Start date: 15/10/20

T2.2:
Arrange customer meeting to ask about key requirements for the game.
Start date: 15/10/20

T2.3:
Write down our design ideas for the game based on the product brief.
Dependencies: Requires T2.1 to be completed
Start date: 26/10/20

T2.4:
Write up the requirements in the Req1 document.
Dependencies: Requires T2.2 and T2.3 to be completed
Start date: 11/11/20

T3.1:
Create an abstract architecture with accompanying UML diagram
Start date: 05/11/20

T3.2:
Create a concrete architecture with accompanying UML diagram
Dependencies: Requires T3.1 to be completed
Start date: 05/11/20

T3.3:
Write up the systematic justification for the architecture choices
Dependencies: Requires T3.1 & T3.2 to be completed
Start date: 16/11/20

T4.1:
Create the project schedule (list of tasks and deliverables) in the Plan1 document.
Start date: 5/11/20

T4.2:
Write up all of the software engineering methods and collaboration tools along with descriptions of why we used them in the Plan1 document.
Start date: 12/11/20

T4.3:
Write up our approach to team organisation in the Plan1 document.
Start date: 12/11/20

T5.1:
Create a list of risks to our project.
Start date: 10/11/20

T5.2:
Create a table in the Risk1 document and use it to contain all of the risks, their likelihood, the severity and how we can mitigate them.
Dependencies: Requires T5.1 to be completed
Start date: 11/11/20

T6.1:
Create a base game to learn the basics of the LibGDX game engine.
Start date: 19/10/20

T6.2:
Implement basic animations, sounds, collisions and movement to the base game.
Start date: 22/10/20

T6.3:
Add basic teleporters for Auber to jump around the map.
Start date: 29/10/20

T6.4:
Implement basic AI pathfinding using LibGDXs built in A* search tools.
Start date: 5/11/2020

T6.5:
Create infiltrators and systems for them to destroy.
Start date: 16/11/2020

T6.6:
Create win and lose conditions as well as the game screens to go with them.
Start date: 18/11/2020

Deliverables:

Website:
Description: The “public face” of our project. Contains links to all of our deliverables as well as our actual game
Related task(s): T1.1

Requirements:
Description: Includes an introduction explaining how our requirements were elicited and negotiated, and why we presented them in the way we did as well as a systematic statement of requirements, split into user, functional and non-functional requirements
Related task(s): T2.1, T2.2, T2.3, T2.4

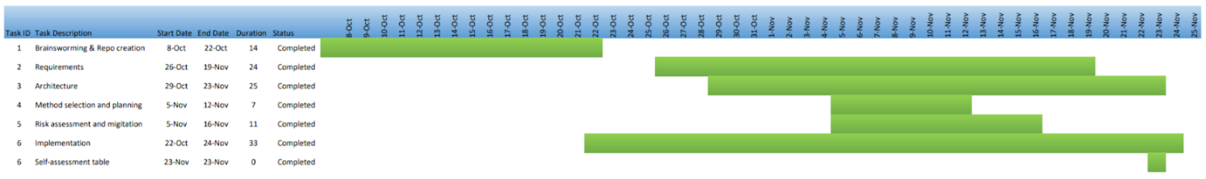
Architecture:

Description: An abstract and concrete representation of the architecture of our team’s software, with a brief description of the languages and tools used to describe the architecture. A systematic justification for the abstract and concrete architecture
Related task(s) T3.1, T3.2, T3.3

Method selection and planning:
Description: Outline and justification of our team’s software engineering methods and an outline of the tools we used during the project as well as a description of why we used them. An outline of our team’s approach to team organisation. Also, a systematic plan for our project, laying out the key tasks with dates, providing weekly snapshots of the plan on our website
Related task(s): T4.1, T4.2, T4.3

Risk assessment and mitigation:
Description: Introduction to our risk format and a systematic tabular presentation of the risks to our project, their likelihood and mitigation
Related tasks: T5.1, T5.2

Implementation:
Description: Provide documented code for a working implementation of our game. State any of the features required that are not fully implemented.
Related tasks: T6.1, T6.2, T6.3, T6.4, T6.5, T6.6



Task ID	Task Description	Start Date	End Date	Duration	Status
1	Brainsworming & Repo creation	8-Oct	22-Oct	14	Completed
2	Requirements	26-Oct	19-Nov	24	Completed
3	Architecture	29-Oct	23-Nov	25	Completed
4	Method selection and planning	5-Nov	12-Nov	7	Completed
5	Risk assessment and migitation	5-Nov	16-Nov	11	Completed
6	Implementation	22-Oct	24-Nov	33	Completed
6	Self-assessment table	23-Nov	23-Nov	0	Completed