Method & Planning

a)

The software engineering method used by the team to develop the Pirate Game was the scrum agile method. We implemented this method as it prioritises the needs of the customer and bases the product around their needs. The scrum method involves:

- Splitting the work into sprint cycles and completing a goal in the stage of developing the product at the end of each cycle.
- Keeping of the developments made by holding scrum meetings and updating each other on what we've managed to complete.
- Continuing this process until the product is complete.

This method suited the product the team was building as it was based on a brief with specific customer requirements where customer meetings could be held in order to ensure the product matched what the customer had in mind. In addition, it is extremely flexible as the planning process is split into several changes that can easily be changed. The steps taken while following this method are:

- Reading the description and requirements for the game which is known as the product backlog.
- Setting a goal based on the requirements and discussing how to develop the game based on these requirements.
- Scheduling a meeting with the customer in order to ask questions relating to the product in order to better understand how to develop the game based on the requirements given.
- Splitting the work into sprint cycles of two weeks and creating a Backlog list of tasks that need to be completed.
- Creating a Gantt chart to keep track of the tasks to complete during each sprint cycle.
- Holding scrum meetings at the end of each cycle where members of the team would update each other on the progress they made, any issues that arose and discuss what they would try to complete before the next meeting.

The collaboration tools used by the team includes:

Tools	**Justification**	**Alternatives considered**
GitHub	We used GitHub as a file sharing tool and collaboration workspace to upload our code onto. This allows team members to easily access and edit each other's work, as well as maintain	

	source control.	
Discord	We used discord as a tool for remote communication and used it to hold our sprint meetings. It was effective as all members could attend and were able to share their screens.	Slack, WhatsApp.
Typora	Typora is a markdown editor which has functionality for converting to PDF. This allowed us to have a consistent style in our deliverables.	Google Docs, Overleaf. We strongly considered using Overleaf which uses LaTeX markup, but there was a much steeper learning curve compared to Markdown.

b)

Our teams approach to organisation consisted of appointing people to different roles such as:

- The team leader, who organised the scrum meetings and ensured all members attended. They would also make sure that members worked on their tasks in between meetings.
- The Secretary who took notes of what occurred during the scrum meetings and took notes of the customer meetings in order to list the requirements and specifications of the game.
- The Librarian who oversaw version control and organised the documents and added them onto the website.

Developing the code for the game and working on the documentation was split equally between the team. Each deliverable had a main person working on it as well as a shadow to ensure that the main person was working on the deliverable. The shadow also helped proofread and research the topic of the deliverable. This type of approach was taken as this is a large project with many different components and not all group members can simultaneously work on a single part.

c)

The table displays the tasks with the start and finish dates as well as the priority with 5 being the highest priority:

Task	**Start**	**End**	**Priority**
Documentation			

Website	12/11/21	01/02/22	4
Requirements	12/11/21	31/01/22	2
Architecture	12/11/21	01/02/22	2
Method selection and planning	12/11/21	01/02/22	2
Risk assessment and mitigation	12/11/21	31/01/22	2
Implementation	12/11/21	31/01/22	5
Requirements			
Team building and reading assessment brief	12/11/21	15/11/21	3
Write-up requirements	12/11/21	10/12/21	3
Discuss requirements	26/11/21	26/11/21	2
Meeting with Customer	02/12/21	02/12/21	4
Review existing systems/games	03/12/21	03/12/21	3
Design			
Learn how to use libGDX	03/11/21	15/12/21	4
Decide on structure and in-game mechanics	17/12/21	28/12/21	4
Design art style	24/01/22	26/01/22	3
Implementation			
Create game screens	19/01/22	30/01/22	5
Implement movement	19/01/22	30/01/22	5
Implement camera	19/01/22	30/01/22	4
Implement combat	25/01/22	30/01/22	4

Testing and verification			
Test movement and Camera	30/01/22	01/02/22	4
Test combat	30/01/22	01/02/22	4
Make sure Implementation matches documentation	30/01/22	01/02/22	3

The plan evolved during the project due to:

- New features we wanted to implement such as art and sound.
- Harder tasks needed to be split into smaller simpler tasks.
- Reviewing and editing the documentation due to changes made in the implementation

Bibliography

- floppy, “International Scrum Master Foundation,” Scrum.as, 2020. https://www.scrum.as/academy.php?show=0&chapter=1
- “Overview of Scrum Phases | StarAgile,” Staragile.com, 2020. https://staragile.com/blog/scrum-phases
- I. Sommerville, “Chapter 3 Agile software development,” in Software engineering, Munchen u.a.: Pearson, 2018.

Method and planning for Assessment 2

Engineering Methods

We are continuing to use the agile software development method as it fits best how we like to work, which is allowing features and steps in our product to be developed in and released in iterations. This helps us track changes easier which then help us with bug fixes or any other issues that may come about. Along with continuous integration, this helps us keep track and on our progress to see if we are heading in the right direction and if our goals will be met

On the development side of the assessment, we are continuing with the pair programming approach where we have two members out of the six on the team on the programming. We have one of the pair being the main spearhead of the operation by being the one who starts the programming while the other looks over it, checks if it would work, for any errors, if it fits in the grand scheme of the project etc. This helps as there are two different views on the code with the first programmer being a more microscopic view where the follow up person is a more macroscopic view. We found this approach to be quite helpful within the dynamic of our group.

We also continued to use GitHub as our version control for the software as it is just the best option for us as we are familiar with it from the previous assessment. It is most useful for us as the issue feature works well with our development style. For any features yet to be implemented, any bugs or any other issues we could raise an issue and during each iteration of our development we could look to tackle them.

Organisation and Planning Methods

As we had six members of our team we divided ourselves into three groups, being: development, testing and documentation. This gave us a good two members per team so one side of the project is not over crowded while another is starved of resources. Also we thought that equally dividing up responsibilities would help streamline our progress.

As mentioned before we had a pair of our team on the software development side of things using the pair programming approach.

They are then accompanied by another pair of software testers. The testers are responsible for, of course, testing each part of the software. In our requirements we have functional requirements and non-functional requirements. The pair of software testers will then split their responsibilities so that one of them carries out the functional testing while the other carries out the non functional testing.

The remaining two members on the team are responsible for sorting out the documentation for the project. We sort out all the documentation that needs to be submitted and see what we can work on together and what could be completed separately. For example, the change report is something we work on simultaneously but the documents we need to change would be worked on separately.

To aid us in organisation and planning we used discord to keep in touch with each other. It helped in organising any major milestones that were about to come about and how to go about completing them. This helped us keep on top of things and track each other's progress so that each team was not completely in the dark about each other's responsibilities and tasks. Of course we also used Google Drive to be able to store and work on documents collaboratively and simultaneously.

The following table describes the tasks we completed with its priority (1 - highest, 5 - lowest):

Task	**Start Date**	**End Date**	**Priority**
Method and Planning	20/2/2022	29/4/2022	2
Requirements	20/2/2022	27/4/2022	2
Risk Assessment	20/2/2022	30/4/2022	3
Architecture	20/2/2022	2/5/2022	2
Change Report	20/2/2022	2/5/2022	3
Testing Document	20/2/2022	3/5/2022	2
Implement Shop	20/2/2022	28/4/2022	1
Implement Weather	20/2/2022	1/5/2022	3
Implement Powerups	20/2/2022	2/5/2022	2
Implement Savegame	20/2/2022	2/5/2022	2
Finishing Product	20/2/2022	3/5/2022	1