Method Selection and Planning

Team 19

DAVIDE BRESSANI
ALEX DAVIS
HASSAN MAHMOOD
DANIEL PARRY
HARRY SWIFT
BENJAMIN WITHNELL

Method Selection and Planning

A)

- In terms of what we did and the lifecycle process, we followed a spiral process.

Methods

- We implemented agile methods and in particular, we implemented scrum. We wanted to produce our game iteratively and incrementally, so the customer would always have some sort of product that could be constantly delivered with our small team.
- This also allows the following team to react well to the requirement changes in assessment 2, since this decreases the level of technical debt.
- We decided to do this as it allows us to react to the moving environment in a rapid efficient manner, as well as maintaining the requirements set by the customer in our inclusive relationship
- Since we focused on delivering code as the priority (because the game is the priority), and focused less on documentation, so there was low documentation overhead as we utilized SCRUM.
- To accomplish this, we utilised face-to-face interactions to the customer as a way to relay information, with software the top priority for the team as we implement the requirements.
- During an agile scrum sprint, requirements / ideas would be implemented one at a time and then tested for its functionality, with a sprint backlog guiding us of tasks to be completed.
- During scrum reviews, the methodology to see what needed doing regarding our backlog and apply accordingly for the following sprint via our tools which aligned with the characteristics of agile processes.
-

Tools:

Google Drive:

- This collaboration tool allowed us to share our documentation to support the team project and allowed live collaboration.
- Word processor was widely accessible to us and simple to use and convert to a pdf.
- This allowed us to be extremely adaptive, allowing us to make continuous synchronised changes, even at later stages in the project. This aligned with our software engineering methodology, and which is why we chose this instead of using Microsoft office.
- It also allowed us to check what had been done for the day in regards to a scrum, and respond to the changing environment in a fast and efficient way.

PlantUML:

- PlantUML can be used in conjunction with eclipse as well as the web browser due to its build in web server to visualise a gantt chart and the architecture of the program throughout the development process
- It's simple to use, due to it being very similar to pseudo code, and the website is easy to use, with the option to easily export images, as a PNG and SVG.
- It also was possible to use it as a link, loading the image directly from the PlantUML web server

Zoom:

- This collaboration tool allowed synchronous communication between us and the client, to maintain cooperation in meetings and scrum meetings.

GitHub:

- Used to store the latest versions of the game as well as host the website using github pages.
- The ability to make branches is an essential tool to work without affecting the other branches in the repository.
  - This allows us to incrementally work on the tasks as we can push with ease in small releases, and this allows us to straightforwardly do a scrum of scrums and check for overlapping work and integration.

Discord:

- Main means of communication. Used in conjunction with snapchat, for priority communication, to organise meetings and keep everyone in the loop.
- Discord was preferred over Slack, which is a more professional version of Discord, because everyone has had experience with Discord, while not so much with Slack

Eclipse and IntelliJ:

- These development tools were used as the primary IDE for development of the game.
- Both simple to use, and effective for running the game efficiently and quickly. The main developers were also very familiar with the IDEs and were able to teach the rest of the team., aligning with agile characteristics since the method was straightforward.

Jira:

- Used to manage tasks which have been completed and tasks which need to be completed.
- Effective at allocating tasks and taking responsibility for what has been completed regarding the scrum method.
- Helped to maintain a steady flow of work.

LibGDX:

- Is a free and open-source game-development application framework written in the Java programming language, making it compatible with our requirements
- This development tool was essential for the creation of the game and therefore was the game engine we chose to use because it was suggested by the lecturers / customer, which we viewed as an integral team member and took their feedback onboard due to our cooperation.

B)

Team organisation:

Forming stage:

For our teams' approach, we used an array of methods to maximise our thought process efficiently. We first established fairness principles, and outlined and implemented the 3 pillars of teamwork, with Humility, respect and trust all being taken into consideration. We also acknowledged that it is okay to give criticism, albeit in the right way, and that receiving criticism is nothing personal and can often lead to improving oneself, as well as the team as a whole. We then allocated tasks.

Storming phase:

We chose the approach of De Bono's Six Thinking Hats. We did this to provide clarity to our thinking process, leading to efficient discussions and successful organisation. We took the bus factor into consideration, with us ensuring that at least two members could cover any component of the project in case of emergency or if they are stuck. This also promoted our resilience and improved morale. We discussed our strengths and weaknesses, in regards to each task, and redistributed roles with a shadow team member attached, now that we knew everyone's strengths and weaknesses. Often, if a rational argument, it would be controlled and voted upon by the team in an amicable manner, with a facilitator wearing the "blue" hat. This approach allows the project to be peer-reviewed and improved.

Norming phase and performance stage:

We became comfortable with each other and part of the team to frequently review each other's work in Friday's lab sessions. We maintained communication throughout, via our snapchat group chat, Jira board and discord channels, where each channel details a different part of our project. This is also better for visibility purposes as more clarity could be made if needed. In the performance stage, each member were autonomous.

Project planning organisation:

We established a plan to maintain communication between ourselves as well as the client, using communication tools. We also decided to allow the customer to be a proxy for an additional team member, with us holding meetings via zoom or asking questions via email or face-to-face interaction to help decision making. A role structure was put in place, with at least two members per team being able to cover per deliverable. In terms of resource allocation, This allowed us to avoid a low bus factor as well as maintaining a hierarchy on deliverables. We chose re-planning as a reliable approach for any issue that may put us off track, thus still allowing us to meet our contractual obligations. Regarding our upfront scheduling, we allowed enough buffer time and overhead in case of extra time needed due to unforeseen circumstances, which prepared us for a bad scenario since there was a deadline in place. This approach allowed us to budget our time efficiently as well as establish clarity and autonomous decision making in the performance stage.

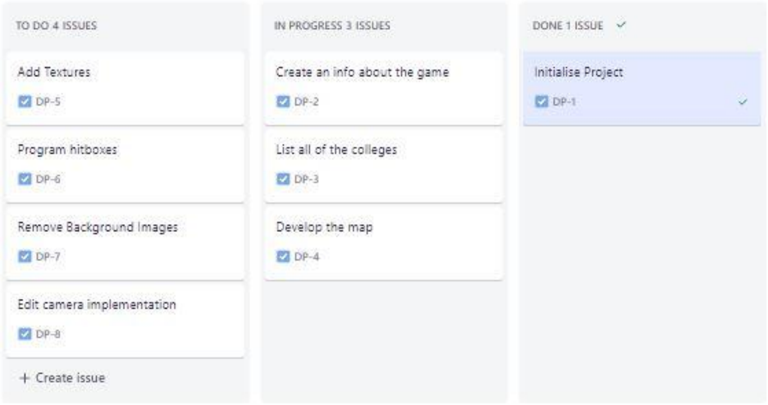
C)

Systematic plan for the project:

Key Tasks	Starting Date	Finishing Date	Priority (1-5)	Dependencies
Initialise Project	10-12-2021	10-12-2021	5	Github setup
Create info about the game	10-12-2021	18-01-2022	2	
Develop the map	10-12-2021	18-01-2022	4	Initialisation is complete
List all of the colleges	10-12-2021	18-01-2022	3	
Add Textures	14-01-2022	21-01-2022	4	
Program hitboxes	14-01-2022	20-01-2022	5	
Edit camera implementation	14-01-2022	20-01-2022	5	
Add WASD controls	21-01-2022	26-01-2022	4	The addition of the ships and map
Fix rotation and movement	21-01-2022	26-01-2022	3	
Create ship constructor	28-01-2022	29-01-2022	3	
Make colleges functional	28-01-2022	29-01-2022	4	
Update ships and colleges to work together	28-01-2022	29-01-2022	5	
College combat	30-01-2022	01-02-2022	4	
Private Classes	30-01-2022	01-02-2022	3	
Points	30-01-2022	01-02-2022	3	
Pause Screen and End Game Screen	30-01-2022	01-02-2022	3	

Link to whole project initial plan

Weekly snapshots are provided on the website, under Weekly updates.



TO DO 4 ISSUES

Add Textures
DP-5

Program hitboxes
DP-6

Remove Background Images
DP-7

Edit camera implementation
DP-8

+ Create issue

IN PROGRESS 3 ISSUES

Create an info about the game
DP-2

List all of the colleges
DP-3

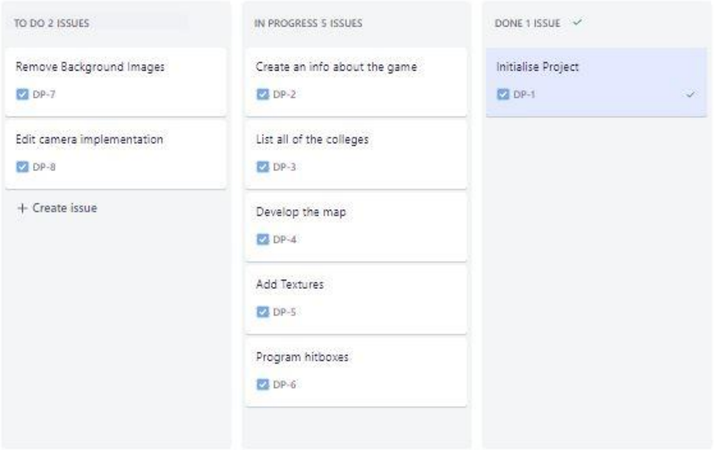
Develop the map
DP-4

DONE 1 ISSUE ✓

Initialise Project
DP-1 ✓

Week 4 Development board

- The initial plan for week 4 was to start the development process
- These plans where however halted temporarily, due to exam revision, as well as the Christmas holiday period



TO DO 2 ISSUES

Remove Background Images
DP-7

Edit camera implementation
DP-8

+ Create issue

IN PROGRESS 5 ISSUES

Create an info about the game
DP-2

List all of the colleges
DP-3

Develop the map
DP-4

Add Textures
DP-5

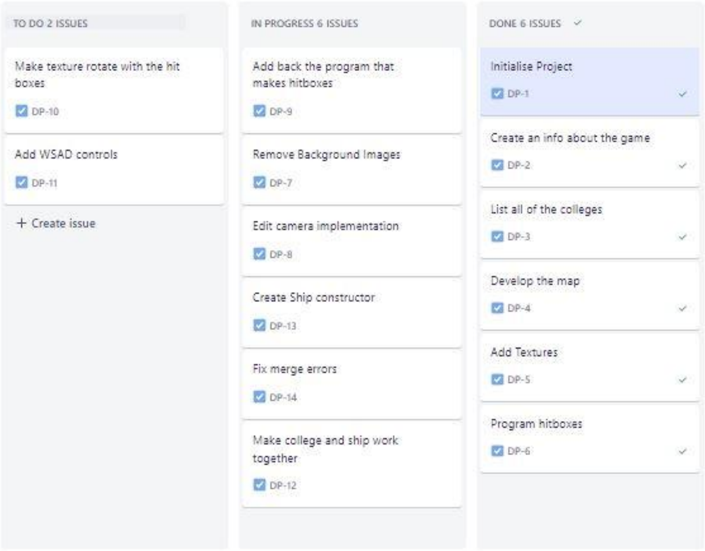
Program hitboxes
DP-6

DONE 1 ISSUE ✓

Initialise Project
DP-1 ✓

Week 6 Development board

- During the week after the exams, the main aim was to get back on track to complete the project
- One thing which made this difficult was members of the group having covid, as well as getting used to new and somewhat different time schedules which made it difficult to schedule meetings



TO DO 2 ISSUES

Make texture rotate with the hit boxes
DP-10

Add WSAD controls
DP-11

+ Create issue

IN PROGRESS 6 ISSUES

Add back the program that makes hitboxes
DP-9

Remove Background Images
DP-7

Edit camera implementation
DP-8

Create Ship constructor
DP-13

Fix merge errors
DP-14

Make college and ship work together
DP-12

DONE 6 ISSUES ✓

Initialise Project
DP-1 ✓

Create an info about the game
DP-2 ✓

List all of the colleges
DP-3 ✓

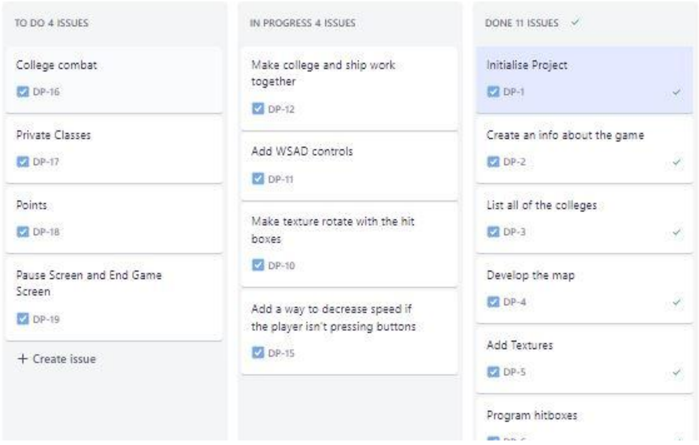
Develop the map
DP-4 ✓

Add Textures
DP-5 ✓

Program hitboxes
DP-6 ✓

Week 7 Development board

- Similar difficulties from week 6, although more progress was made on the development of the application
- The rising pressure to complete the development, also helped to increase the number of tasks completed each week



TO DO 4 ISSUES

College combat
DP-16

Private Classes
DP-17

Points
DP-18

Pause Screen and End Game Screen
DP-19

+ Create issue

IN PROGRESS 4 ISSUES

Make college and ship work together
DP-12

Add WSAD controls
DP-11

Make texture rotate with the hit boxes
DP-10

Add a way to decrease speed if the player isn't pressing buttons
DP-15

DONE 11 ISSUES ✓

Initialise Project
DP-1 ✓

Create an info about the game
DP-2 ✓

List all of the colleges
DP-3 ✓

Develop the map
DP-4 ✓

Add Textures
DP-5 ✓

Program hitboxes

Week 8 Development board

- Final week before the deadline meant that we had time to add finishing touches.
- There were still a few tasks to complete, however all the tasks where allocated and completed