Method Selection and Planning

Engineering method and tools

We decided on using agile methods to develop our program which meant having Scrums and Sprints. Scrums were used to assess what we had done the past week and what plans and goals were for the following week, also this time was used to figure out when each pair was going to meet up in our pairs to do our coding sprints.

The key tools used for aiding development and communication were the following. For software development we used Visual Studio Code. This was very useful for pairs during their sprints as visual studio allows extensions such as liveshare, which means that one person has their visual studio code open and they can invite their team member to join the coding session and both people in the pairs could code at the same time without clashing with each other. Visual studio also links in very well with github as there are tabs on the side of the IDE that helps users with version control and notifies you when changes are being made to different classes using various symbols next to their names. There are also shortcuts from the github desktop application which allows you to open the code of the branch you are currently on in visual studio without having to select the application. Visual studio was very good during our development however sometimes when we tried running our code there would be a random error that occurs that wouldn't be there when re-running the code and was just an error caused by visual studio. Alternatives to visual studio code would be other IDEs such as Eclipse and IntelliJ, however we decided on visual studio due to the flexibility of having many useful extensions that aided in our collaboration and ability to work as a team. Although at the end of the project we had trouble creating an executable JAR file in visual studio so had to briefly go to eclipse in order to create it.

For managing our code as a team we were fairly set on Github as this is very popular and is used by many big companies. This was difficult to use at first and so caused some confusion among the team however as the project went on we found that it became a very powerful tool and helped massively with the development of our project. As stated previously github works very well with visual studio code and helped with learning as we could open our code directly from github to visual studio seamlessly. Working on branches was also very useful as it stopped there from clashing if it were the case that pairs would end up working and changing the same class and pushing it to their branch. When we were discussing alternatives from github we didn't have many ideas off the top of our head, but some alternatives are in fact there such as bitbucket. But in the end we settled on github as it is the most widely known to the team and using something that is popular and well known is often the safest course of action.

For communication we decided to use slack as it is a classic workplace communication system that is free and whenever the app is not visibly open if a message is sent to you or one of the workspaces that you are in it pops up at the bottom of your screen notifying you that you have a message this is quite a useful feature especially during coronavirus when

there is no face to face meetings which makes sure you don't miss a message when one is sent to you. During coding sprints and scrums whenever another member of the team was working on something like graphics for the game slack provided an easy and quick way of sharing these resources. When sending coding ideas to each other code appears in a special formatted box indicating that the message sent is code and is formatted to make the recipient able to read the code in a way which makes it much easier to read. In slack you can set up things called channels, this is a useful tool that allowed the team to split up communication for different parts of the program so you could have a channel dedicated to graphic design and a channel dedicated to something else as well as a general channel for communication between the whole group about the project or just general chat, it also linked well with our task management system. Alternatives to slack would be applications like Microsoft teams or Discord, we decided against teams as we were not sure whether we could use it for free, we also decided against discord as although as a team we were probably more familiar with the program it is often seen as less professional and may have caused the team to be less professional by association. Our team meetings were conducted on Google meet as everybody knew how to use it and were comfortable with it.

For task management we used Trello, this is often seen as the classic business task management system that goes along with slack as you can link your trello and slack account. The way we used trello was by having three sections: 'To do', 'Doing' and 'Done'. To these sections we added cards (these are the tasks) and for each card in the 'To do' section we labeled it with the colours of the team working on it so that there was a clear distinction of who had what tasks. When a task was being worked on it was moved into the 'Doing' section and when completed was put into the 'Done' section after this in our scrums we would review the code that had been marked 'Done' to see if any changes needed to be done to the code if we came up with any new ideas that would perhaps need to change the code of the class in the 'Done' section. Alternatives to trello were also hard to think of as trello is so popular, however there are alternatives like Asana. But in the end we decided with trello as it works with slack and as mentioned previously it is safer to go with something that is popular that is tried and tested.

Outline of Teams Organization

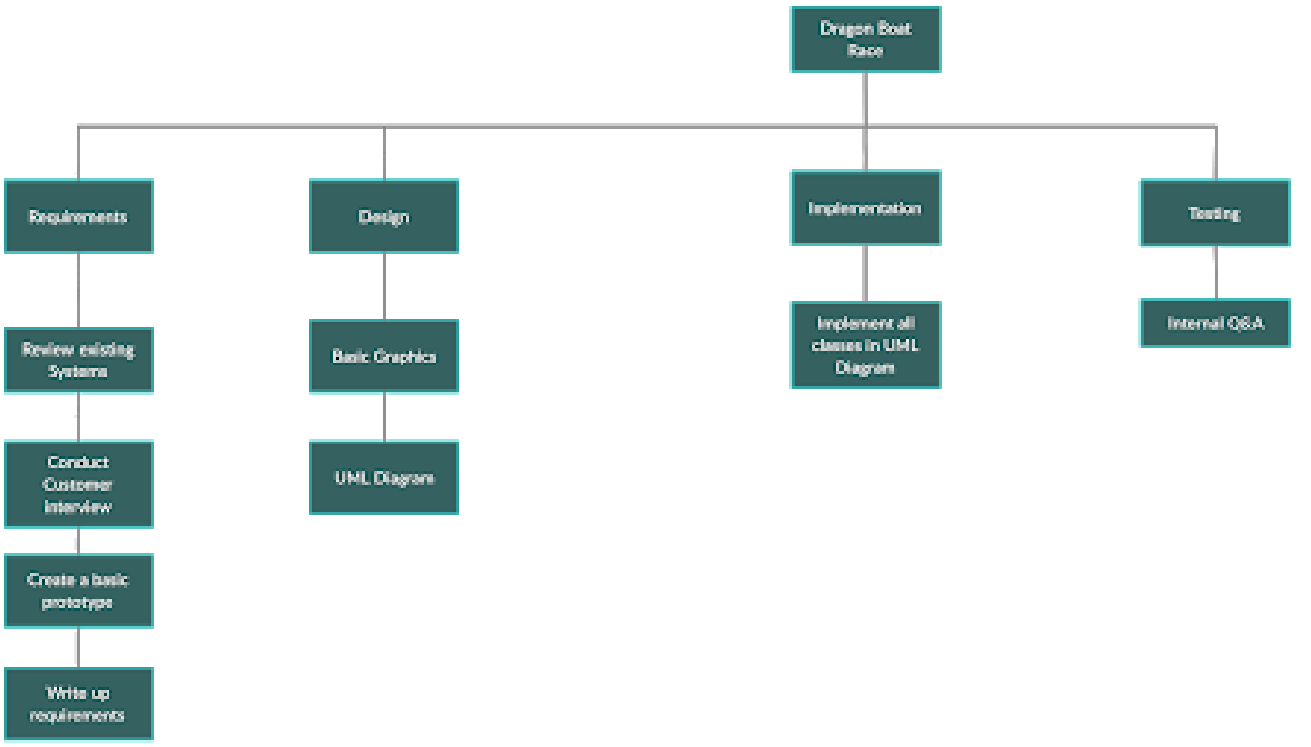
We followed a waterfall approach using agile development methods, we decided to do this because as a team we thought that as we develop the game off the base requirements given to use by the customer we could adapt to any changes and new requirements, ideas and on how to implement ideas with relative ease. Also due to coronavirus and ever changing conditions we split up into groups of 2 trying to balance out Java coding abilities so that each team had a fair amount of knowledge and experience to help aid in the development process. As before the project we did not know each other we didn't know how other individuals worked so using agile methods was wise as it allowed us to adapt as a team as the project moved forward and allowed us to redistribute tasks among teams if necessary.

To keep on top of our progress every week we decided to have a team scrum to discuss what we had managed to complete in our pairs and as a whole team by doing this it gave us an idea on how well the team worked together and how quickly each team were completing their tasks. Afterwards we would discuss what we thought would be good objectives and goals to reach by next week's scrum, then we would go away and in our pairs we would schedule coding sprints were each team would get together and work towards and hopefully achieve their goals for the next weeks scrum and at the end of the coding session reflect on any ideas that they have and decide whether they had any new ideas or thoughts about what has been coded and perhaps whether they thought there could be anything they could add for when the next coding sprint was.

Working in pairs was a good idea because as we were strangers to each other before the project it would provide some for of moral incentive to turning up and pulling your weight as if you were not pulling your weight you wouldn't just be making yourself look bad you may also make your partner and team look bad to other people in the team and to onlookers looking at the team from an outside perspective. This moral incentive is typically enough to make one feel bad for not pulling their weight causing that person to work harder later down the line which may in fact help the team in some way shape or for, be it becoming very motivated to prove yourself to the team by hyper focusing and helping with the project a lot later on.

Plan for the Project

The start date for the project was the 8th October and ending on the 25th November, this gives our team 48 days to complete the project. First thing to do for the project was to create a simple breakdown diagram of the project.



After creating the simple breakdown diagram a Gantt diagram for the aforementioned breakdown diagram as seen below.

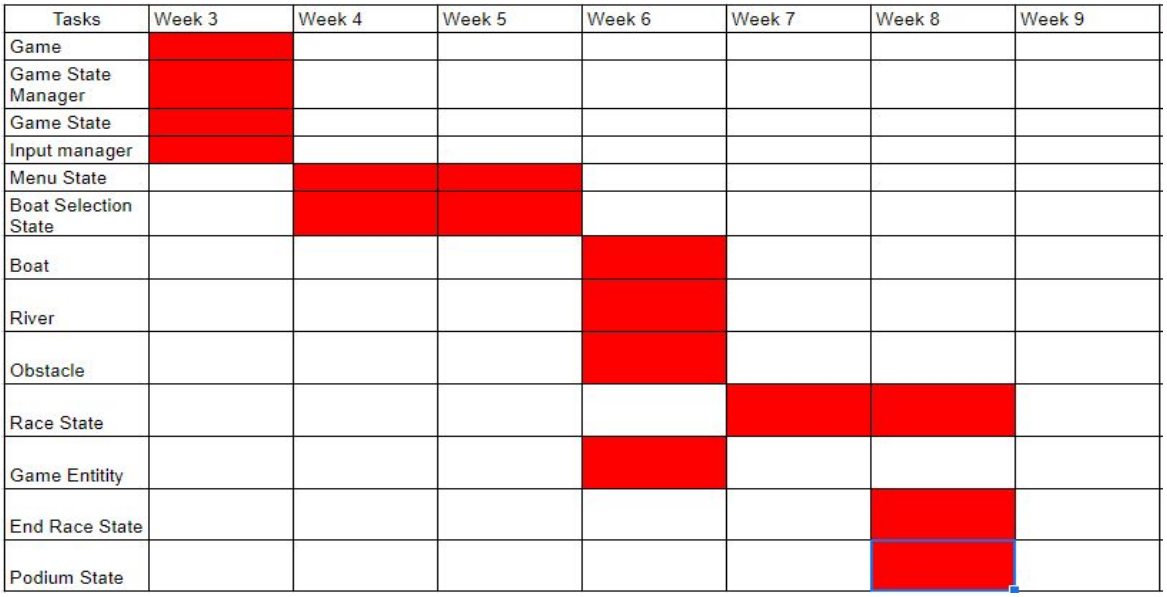
Tasks	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9
Review existing Systems									
Create intial Requirements									
Conduct Customer Interview									
Design Graphics									
UML Diagram									
Implementation									
Testing									
Release									

Once all the initial plans were put in place the next thing to do was to carry out our plan. So we looked at existing systems and came up with some initial requirements. Afterwards we scheduled a customer meeting which enabled us to get a formal list of requirements from the customer meaning we could then improve our existing requirements and UML diagram.

Then decided on our pairs as a team and decided to turn our UML diagram into a list of tasks on trello and for each class we would assign teams.

The way we assigned the tasks was so that each team had a super class and all/most of its subclasses so that all the subclasses that depended on that super class could all be developed by the same team. However since all the classes depended on the initial Game class that was developed in the meeting before the tasks were divided up so that we had something to start from and not just creating it down the line which would inevitably cause errors and would be very complex.

Class Gantt Diagram



In terms of replanning there was a point in the project when we realised there were some classes that were not done in time for when we needed them so some pairs would group up into a team of 4 so that they could all focus on that class and get it done so the the team could move on and progress further in the program.

In terms of scheduling for when the project would be finished it the initial deadline was Thursday 12th of November, however due to unforeseen circumstances with some of our team members we had to change this and delay the product release to Thursday 19th of November.

Link to the trello board for weekly snapshots: https://trello.com/b/Xx6LzbRD/eng1-project

The critical path of the program would go as follows:
Game → GameStateManager → GameState → MenuState → BoatSelectionState → RaceState → EndRaceState → PodiumState