METHOD SELECTION AND PLANNING

Quackthulu: Aaron Price, Alec Coates, Charlie Curedale-Rayner, Eleanor Griffin-Smith

4.a

Our team uses Zoom to hold our video conference meetings because it is easy to use and allows us to schedule the recurring meetings that we use for our scrums. It also has a screen-sharing feature that makes it very easy to show each other exactly what we are talking about when talking about a specific part of the program or outlining a certain issue - this allows us to communicate much more like if we were meeting in person and could physically see each other’s screens.

We use Facebook Messenger to share messages as a group for giving updates or requesting help that can’t wait until the next meeting. This allows us to save valuable development time by not having anyone waiting for a group meeting to continue their tasks. There are lots of digital messaging services such as WhatsApp or Discord, but we chose Messenger because it had all the features we needed and it was something that all members of the group already had access to, so we didn’t have to wait for one of use to make a new account.

For our basic documentation and modelling we have a shared Google Drive because it is easy to access and the documents kept in it do not necessitate any serious version control. The documents in the Google Drive also support Google Docs, which allows team members to work on the same document simultaneously, which is good for collaborative changes and reviews. We also used Diagrams.net (formerly Draw.io) to model our program because it has native support for Google Drive integration. We considered using other sites such as DropBox or Microsoft OneDrive, but none of them had collaboration features as good as Google Drive, plus all of our team members already had education Google accounts with unlimited Google Drive storage space, so we had more space and didn’t have to waste time setting up new accounts.

We decided to use Github for the program’s actual code because we wanted to have full version control and we also wanted group members to be able to make changes to the code individually and still be able to build/run the code they have changed, then any changes they have completed can be pushed to the whole group. This ensures that no member gets stuck at any point in time because somebody else has a half-finished feature that is stopping the rest of the program from working correctly. Github also has a convenient desktop GUI application that made it much easier to use and quicker to pick up for project members who had never used Git previously. We also used Github’s Pages feature to host our project website, which we can use to deliver project information to the client. We considered using GitLat instead of Github, but Github had a better desktop client, which was easier for the new Git users to understand and the client has better integration with Github than GitLab because it is their own service.

4.b

For our project, our group used the Scrum agile engineering method. We chose this method because we are a small team of 5 developers meaning that we can all easily communicate together as a team and effectively utilise every member's skills and capabilities. Also, having short iteration cycles for our development helps allow us to model, design and program the game within the short time frame of under 2 months that was given to us by the client. Our team holds bi-weekly meetings to discuss the development updates we have made, provide support for any team members who are struggling with any of their tasks and plan what needs to be done for the next meeting. Each week we have a sprint to complete the next logical section(s) of the project that needs completing, with a meeting mid-week to ensure that everybody is on track and to give new tasks to anyone who may have completed all of their assigned tasks earlier than initially projected. All of our meetings are held using internet video conferencing because the current COVID-19 pandemic limits us from meeting together in person due to health risks.

4.c

	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9
Objectives	Come up with team name	Research requirement	Write up requirements	Code	Code	Code	Code	Do self assessment
	Everyone reminds themselves of Java	Choose an IDE	Research architecture					Finalise and submit zip
	Create google drive	Research risk assessment	Write up risk assessment					
	Create GitHub	Create website	Start coding					

Task dependencies

Requirements -> architecture -> implementation

Implementation cannot be started without architecture and architecture cannot be started without requirements.

Task priorities

Our priority was to make a plan for the project and organise the team so we could work efficiently. Our second priority is to define the requirements as we need this to do most of the other part of the project. Our next priority is architecture so we can start implementation.

In order to keep to the time frame – during every weekly engineering practical we decided what each team member would do for the next week. These plans and weekly progress grids are available on the website