Team Coding Coders Who Code Things - Jacob, Joey, Kenny, Minye, Zach

Meeting Log

Date	Location	Attendees	Description
09/06/2019	SPAHR Auditorium	Kenny. Jacob, Joey, Minye, Zach	Formed team, discussed that we will use C++ and each of our strengths/experiences with coding
09/09/2019	SPAHR Auditorium	Kenny. Jacob, Joey, Minye, Zach	Set up GroupMe, Discord, and GitHub
09/13/2019	SPAHR Auditorium	Kenny. Jacob, Joey, Minye, Zach	Discussed possible classes, functions, and code outline for the project. Scrum meeting
09/13/2019	Eaton Hall Lab	Kenny. Jacob, Joey, Minye, Zach	Began work on the map class and discussed issues with keeping track of ships on the map. Also clarified any GitHub issues
09/15/2019	Eaton Hall Lab	Kenny, Jacob, Joey, Minye, Zach	Continued working on project as a group, worked more on GitHub, finalized division of work
09/16/2019	SPAHR Auditorium	Jacob, Joey, Minye, Zach	Scrum meeting, discussed progress made from the previous meeting and what else is left to complete
09/16/2019	Eaton Hall Lab	Jacob, Zach	Continued working on Map class

09/18/2019	SPAHR Auditorium	Jacob, Joey, Minye	Scrum meeting, discussed progress and current issues. Planned time to complete remaining tasks
9/19/2019	Eaton Hall Lab	Jacob, Joey, Zach, Kenny	Continued working on map class and fixed major bugs
9/20/2019	Eaton Hall Lab	Jacob, Joey, Zach, Kenny, Minye	Began work on documentation and finishing touches on code
9/22/2019	Eaton Hall Lab	Joey and Kenny	Fixed final bug with user input, completed documentation

How Work Was Split

During the first meetings, we discussed as a team what the basic outline of the project would be. From these discussions, we each volunteered for a part that we felt most comfortable with. Once we began coding, we reassigned the work after figuring out specific issues with the project outline. For example, we redesigned the classes and the functions each class would have. Jacob chose to work on the Map class, Zach built the original framework for the project and helped Jacob with the Map class, Kenny volunteered to complete the Executive class and help with the documentation and comments, Minye helped each person with GitHub and rebuilt the framework for the project, and Joey fulfilled a QA role by testing the code, improving the UI, and helped Kenny with the comments and documentation. We all helped with the coding process by answering teammate's questions and helping each other when one person would face a problem.

Jacob: Map class, Executive class

Joey: Tester, comments, documentation

Kenny: Executive class, comments, documentation

Minye: GitHub, rebuilt framework, Executive class, formatting

Zach: Built original framework, Map class, Executive class

Challenges and How They Were Overcome

The main challenges mostly had to do with scheduling, division of work, uncertainty with GitHub, and having to learn how to use automatic documentation. When initially forming the team and beginning work on the project, we immediately noticed that it would be very difficult

for all of us to be present whenever we wanted to work during the week. To overcome this challenge, we decided that we would work on our separate parts individually and meet as a team during the weekend. As for the division of work, each of us were uncertain with each team member's coding skills. To solve this issue, we made an outline for how we wanted to project to look and each member volunteered for what part he felt most comfortable working on. However, when development began, we redesigned the outline to be more streamlined and re-volunteered for the work. Moreover, this was also each member's first introduction to GitHub, except for Minye. During our lab meeting on 09/13/2019, Minye explained how the GitHub process worked while also creating the repository. In the subsequent meetings, Minye helped each team member with GitHub after redesigning the project outline. Lastly, this was also our first time using automatic documentation. We were unsure how this worked and how we needed to format our comments. When using Doxygen to generate the html documentation, we had to reread the manual many times to understand what needed to be done. We had to reformat every comment according to Doxygen's manual. Once this was done, we ran Doxygen and documentation was generated. For the first time using this tool and without previous examples, we believe that this was an accurate representation of what was needed.

Unadded Features

Two unadded features that we thought about were asking the user if they would like to replace his or her ships before beginning the game and being able to save the game and restart it later. For replacing the ships, we thought it would have been a good idea to let the player see where all of his or her ships were before beginning the shooting process. This would allow the player to

see all of the boats and give them the option to replace any ships before starting the game. For saving the game, one idea was to save the current maps and player turn to a text file. When the players would want to continue the game, they could choose an option to continue the game from the menu and upload the text file.

Retrospective

When looking back on the project, we believe that we had a large success. We included all of the features that were needed while also being open to new ideas and challenges that we faced. Once we set up our Discord, we were able to easily communicate any questions or issues with the whole team. This was also a good way for us to brainstorm ideas and possible features. On the other hand, we need to improve coordinating each member's roles and responsibilities. When starting this project, we did not immediately begin work, and instead, procrastinated setting up the GitHub repository and outlining the code. Moreover, this was everyone's first time using GitHub, except for Minye, and this proved to be a large challenge when starting development. We believe that we did our best work for the html documentation, given that this was our first time using this tool. Looking forward to the next project, we will begin sooner and have a more defined way to delegate the work due to more necessary features. We will also have a better understanding of how the html documentation process works. Lastly, we have a clearer idea of everyone's schedules so we can better plan for meetings and continue practicing good communication.