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

Meeting Log

Date	Location	Attendees	Description
09/27/2019	SPAHR Auditorium	Kenny. Jacob, Joey, Minye, Zach	Began discussing possible features for the second project and scheduling
9/30/2019	SPAHR Auditorium	Kenny. Jacob, Joey, Minye, Zach	Continued discussing possible features (Salvo mode and game history), discussed scheduling conflicts with Fall Break
10/2/2019	SPAHR Auditorium	Kenny, Joey, Minye, Zach	Discussed how to implement new features and began outlining work and schedule
10/7/2019	SPAHR Auditorium	Zach, Minye, Joey	Discussed what work was done with rewriting the code. Ship placement is still buggy and will work on that first
10/9/2019	SPAHR Auditorium	Kenny. Jacob, Joey, Minye, Zach	Continued discussing what work is left. Will implement game history during lab time later this week
10/11/2019	Eaton Lab	Kenny. Jacob, Joey, Minye, Zach	Discussed what work each team member should complete during break.

10/17/2019	Eaton Lab	Zach and Minye	Worked on executive class and fixed some bugs
10/18/2019	SPAHR Auditorium	Kenny. Jacob, Joey, Minye, Zach	Discussed what each person accomplished over break and what else needs to be finished
10/18/2019	Eaton Lab	Kenny. Jacob, Joey, Minye, Zach	Continued working ship placement with AI, bug fixing, documentation

Custom Features

Salvo Mode: Players load 5 shots at a time and all are fired at once

Game History: Prints winner of each game, the mode played, and number of ships

How Work Was Split

Since this was a continuation of Project 1, we decided to keep our original division of work. As for the added features and AI, we each volunteered for which ever feature we were most interested. Minye decided to work on the AI, Zach worked on implementing the two classes and Salvo mode, Kenny improved the UI and tested the code, Jacob fixed the AI ship placement issues, and Joey designed the game history and completed the 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: AI ship placement and bug fixing

Joey: Game history, documentation, and bug fixing

Kenny: UI and bug testing

Minye: Easy/Medium/Hard AI, bug fixing, and GitHub

Zach: Player Class and Executive Class

Challenges and How They Were Overcome

The main challenge included having to fix many issues with the inherited code base and its lack of documentation. When we initially inherited the project, we noticed that the code was not modular. The majority of the code was written in the main.cpp file with no comments. This lead

the code to be extremely difficult to troubleshoot and to add features. Moreover, there was an abundance of unused variables and functions. For example, there was a function to check the input's type that was repeated in every class but was only used in one class, one time. These unused functions and variables along with the lack of comments made us very confused as to how the project worked. Furthermore, there was no documentation, save for their meeting log. There was no generated HTML documentation nor retrospective write up. Overall, the code was very unstable and would crash at any moment. It seemed like the group did the bare minimum. To solve these issues, we had to almost completely rewrite the whole code base from scratch. We were able to fix a majority of their issues using our Project 1 code. However, we did face some issues when trying to implement our algorithms with their code. As for the game history and Salvo mode, these two features were very simple to implement and improved gameplay. Similarly, when initially implementing the AI, we faced no large issues, and any problems we faced were easily fixed

Unadded Features

Some other features that could be added are asking the players if he or she wanted to replace his or her ships before starting the game and being able to save and resume the game later. With replacing the ships, it would allow the user to see where the ships are placed before starting the game and give the option to replace ships. For the other idea, the user could save the current maps to a file. This file could then be uploaded later and the game could be resumed.

Retrospective

For Project 2, we believe that this was a good test of our problem solving and teamwork skills. We faced many issues when trying to fix the inherited code base and this allowed us to practice our communication and problem solving skills. Some things that went well were the AI and other feature implementations. Minye was able to easily implement the AI modes with Jacob's help for randomly placing the ships. Zach and Joey were also easily able to implement Salvo mode and the game history. Lastly, Kenny found many bugs when working on the project. On the other hand, we did not improve with our procrastination and we were not expecting to have to do as much rewriting that we did. When we first saw the code, we knew that it would need to be rewritten, but we did not know how much until we started coding. This, along with Fall Break, caused us to start the bulk of the project in the last week. However, since we had experience with GitHub and the HTML documentation, we did not have to spend as much time on those requirements as we did for Project 1. Looking ahead for Project 3, we will try to start the project sooner. We also have an even better idea of each person's strengths. This will be helpful when figuring out what to do as our project. We would also like to implement a Gantt chart for the next project. Using this chart will help us visualize each task that needs to be done while also showing us a basic timeline for the project.