Project 1 Retrospective Write-up for “Big SegFault Energy”

Team Members:

Chance Penner

Haonan Hu

Markus Becerra

Sarah Scott

Thomas Gardner

Git Link:

<https://github.com/MarkusBecerra/BattleShip>

Date finished: September 20, 2019

Group Meeting log:

**Meeting 1:**

09/06/2019 @ 11:47AM to 11:50AM

Location: Eaton 2

All in attendance.

Agenda:

\*Chose C++ as language for project.

\*Set next meeting date.

**Meeting 2:**

09/07/2019 @ 2:40PM to 3:30PM

Location: Alcove 1326 in LEEP2

All in attendance.

Agenda:

\*Brainstorming

\*Ran through simulation

\*Started Class Creation (pseudocode)

\*Set up potential next meeting.

**Meeting 3:**

09/09/2019 @11:40AM to 11:50AM

Agenda:

\*Discussed changes we made since last meeting

\*Planned to meet in lab in two days

**Meeting 4:**

09/11/2019 @ 9AM to 10:50AM

Location: Eaton Hall in a Lab Room

All in attendance

Agenda:

\*Created Board and Player classes

**Meeting 5:**

09/13/2019 11:30AM to 11:50AM

Location: Eaton Lawr2

All in attendance.

Agenda:

\*Discussed when to meet next and what to focus on

**Meeting 6:**

09/14/2019 @ 2:30PM to 4:00PM

Location: LEEP2 1322

All in attendance.

Agenda:

\*Outlined classes and their functions

\*Added class functions

\*Mapped out the flow of the program, step by step

**Meeting 7:**

09/16/2019 @ 4:00 PM to 6:00PM

Location: LEEP2 1322

Haonan and Chance

Agenda:

\*try and catch implementation

\*Get the board operating without ships

**Meeting 8:**

09/18/2019 @ 9:00AM to 10:50AM

Location: Eaton 1005C

All in attendance

Agenda:

\*Everyone works on Ship set up together

\*Now ships can be correctly placed on the board

\*Some methods left in Ship class need be defined(\*\*\*\*\*Important\*\*\*\*\*)

**Meeting 9:**

09/20/2019 @ 10:00AM to 11:00AM

Location: Eaton 1005A

Chance, Thomas, Sarah, Markus

Agenda:

\*Worked on input verification

\*Worked on ship class keeping track of destroyed ships

**Meeting 10:**

09/20/2019 @ 12:00PM to 2:00PM

Location: LEEP2

Chance, Thomas

Agenda:

\*Worked on damage counter and game over

\*Boat verification input

**Meeting 11:**

09/20/2019 @ 4:10PM to 5:30PM

Location: LEEP2

Chance, Thomas

Agenda:

\*Completed game over functionality

\*Add input verification for coordinates

Work Distribution:

Our project is formed by 4 classes: Ship class, Board class, Player class and Executive class, we split work by class, Chance is mainly in charge of Board class, he created default board layout and helped other people with their methods implementation as well. Thomas is doing Executive Class, which prints the user interface, main menu and battleship game rule. Haonan implemented Player Class, it controls player’s movement like shoot or get shot by other player, he also did documentation for the group. Sarah helped doing players shooting function and make sure board is being updated. Markus did most of Ship related methods and put the ship tracker on board.

Challenges:

First of all, since this is our first group project, everyone is excited about it and they shared a lot of brilliant ideas for setting up the game, and here I mean a lot! We just have too many thoughts about this game, which increased difficulty for everyone staying on the same page. For example, we nearly spent most of time discussing what features we should add into the game in one of our meeting, which was super fun, because everyone got involved. But at the end of meeting, we barely had any thoughts in common and didn’t have a specific plan for game setting up.

Secondly, due to the different schedule, we cannot meet as frequent as we can. Our common schedule would be Saturday for about 1 to 2 hours, which could be relatively short to work together. Instead, we decided to meet as small group, it helps a lot. During a small group meeting, people work on problems that they have on their coding. Thanks to it, we solved a lot of critical problems in our program and kept the game functional.

Collaboration between team members is definitely one of the challenges. We are all new to the GitHub platform, we made a lot mistakes when we were using it. We have to constantly keep each other up to date on what we were working with. Every time someone is trying to push their work to master branch, he/she will text other group members in GroupMe and avoid merge conflict. It is a little annoying at the first time, because you have to stop your work and pull others’ work first, but we all get used to it.

Unfinished Features:

We had a lot features that we wanted to add into our games at the second meeting, here is some of them:

1. We wanted to add sound effect to ship actions
2. Player can control their ships with WASD
3. Timer for game time lapse
4. Player name customization

Would have done different:

After our first project, we realized that we still have a lot to improve. Communication between team members must be more frequent and detailed. We need some well-planned ideas, not just saying, it can save a lot of time during the meeting. Also, we need to focus on basic functionality of the program first, once we achieved, then we can move on to next stage, which is making the program looks fancy. Another approach that would be different is the division of work, for this program, we initially have four classes, but we have 5 people in the group, splitting work and make everyone is doing same amount of job is really hard. One of example for this imbalance is Board class, it has most methods and most complicated implementation, no one in our group wanted to do it alone I believe, but once everyone else picked their job and Board class became left over. And then we had to work on this class together, which became a chaos situation. On next project, we need to spend more time on splitting tasks, class-based work distribution should not be accepted. It could cause imbalanced work and conflict.

Work Cited:

https://stackoverflow.com/questions/903221/press-enter-to-continue

https://stackoverflow.com/questions/2616906/how-do-i-output-coloured-text-to-a-linux-terminal

https://stackoverflow.com/questions/5029840/convert-char-to-int-in-c-and-c