Poor Yorick Write-Up

Max Goad, Jace Bayless, Apurva Rai, Tri Pham, Meet Kapadia

As soon as our team was formed and we knew we had to make Battleship as our project, each one of us had a lot of ideas running through our brains; however, once we started talking more and outlining, it became more and more clear how we were going to approach this project. Max in the past has had experience using Java and JavaFX, and he explained that because of this he had a clear direction we can head and how it would be easy to do this in Java and make it look pretty as well. Meet, Tri, and Apurva never have coded in Java, so Jace and Max took the lead in the project coding wise and teaching the others on how Java works. All this happened on the day our team was formed, which was on 6th September. Here are the rest of our meeting logs Meet has been keeping track of since then:

#1 9/6/19 In-class Eaton 2 (everybody Present)

Decided to work in Java. Max and Jace the most experienced will take the lead.

#2        9/9/19 in-class Eaton 2 (everybody Present)

Tomorrow meet up and start the project. Meet-up on Wednesday as well. Max will set up the GitHub by then. All the other details will be discussed then.

#3        9/10/19 Eaton 1005C (everybody Present)

Got everything set up Github wise and decided to meet up over the weekend to get significant of the coding done.

#4        9/13/19 in-class Eaton 2 (everybody Present)

Meeting this Sunday to get significant of the code/project done.

#5 9/15/19 Eaton 1005C (everybody Present)

Got started on the coding, got a bulk of it done. Needs to be cleaned up. Meet did the placement code and assumed the role of the Scrum Master, Jace created the board, Tri made the ships, Apurva and Max did the GUI and modified the code to fit the GUI.

#6 9/18/19 Eaton in-class (everybody Present)

Decided to meet up Friday/Saturday/Sunday to clean and wrap everything up. Meet got started on the documentation and Jace and Tri got started on combining different classes

#7 9/20/19 Alcove in the Spahr Library (everybody Present)

Combined some of our classes, further worked on the GUI, and finished the backend of the project

#8 9/21/19 Alcove in the Spahr Library (everybody Present)

Combined some of our classes and further worked on the GUI.

#9 9/22/19 Alcove in the Spahr Library (everybody Present)

Cleaned up everything and finalized our project. Did the documentation as well.

Max was in-charge of making the GUI. No one else knew exactly how to do this, so Max volunteered to take the lead. Max did bulk of the GUI because no one else knew exactly how to do it and needed to be taught. Others were able to contribute with the best of their abilities, but this was definitely Max’s domain. Max contributed to every single file in the GUI, and at one point once we were done with our specific parts and only had one file left, we just sat around one laptop and coded that together. This code was done on Max’s laptop but we all ended up being around him pitching ideas and coming up with solutions together. Not only that, he took the lead on how exactly the overall design went as well. He also modified some of the code that others made so it would fit and work with the GUI interface. Max’s laptop is the only one that compiled some of the Javadoc stuff.

Apurva was basically Max’s right-hand-man. Max needed help with the GUI as he couldn’t do it all alone, that’s where Apurva stepped in. Apurva had to go over a bit of a learning curve to learn how to do the GUI, but with Max’s help, he was able to contribute with that as well. The some of the GUI was done by Apurva and Max’s teamwork. The biggest issue Apurva had was that his personal computer/laptop was having issues in making/compiling the code. This made it to where every code or file that he worked on, he had to push it to the github and had another person test it for him. Because of this, Apurva was mostly the idea person who helped others come up with solutions.

Jace being really experienced in Java, made the board class for the program. This class handles creating the Battleship board itself and takes care the actions that the ships take. He also created the “README” documentation. Jace also helped Tri and Meet a lot in learning Java and showing them how it works. Furthermore, Jace was in-charge of the backend stuff, and majority of that code was done by him and Tri. Jace also played a huge role in getting the game to work on the terminal which gave us a back up plan just in case our GUI didn’t come through.

Meet being the least experienced in Java, did the ship placement code for the game and was in-charge of everything a Scrum Master would be in-charge of. He tracked and set up all the meetings and did majority of the documentation as well. The works cited, the write-up, and the code documentation was done by Meet. He also ironed out several aspects in the GUI, mainly the Players Option portion of the screen, along with the ship placement. Meet worked on the backend a little bit and the front end (GUI) as well. Meet ended up having the same issues as Apurva when it came to his laptop, but he did all the documentation, helped make the code work in the terminal, and helped solve some of the issues others were having with their code.

Tri was in-charge of making the Ships class. This class involved the most amount of code when it came to the core of the game, and Tri wanted to and volunteered to finish this class and was able to finish it with ease. Tri, Jace, and Meet also helped in combing all their code so they are compatible with each other and worked in the terminal. Tri also worked on the GUI when the backend stuff was finished. Tri was able to make the win screen and the transition screen in the GUI and helped make the game work in the terminal.

The biggest challenge we had to face was that 60% of our team had not ever used Java before. Knowing this, those three people made the effort on their own time to read up and learn a little bit of Java; furthermore, Max and Jace helped out a lot as well when it came to learning the basic code. The second biggest challenge we faced was the GUI. Jace, Tri, and Meet already got the code to work within the terminal when all their files were combined, but that code obviously didn’t transfer over fully to the GUI. Because of this, the code had to be remodified and the GUI itself ended up being way more complex and intuitive than expected. The only person that had any knowledge and experience with the GUI was Max, so the others had to go over a major learning curve, but we were able to do that and were able to contribute to the GUI. The GUI took the most amount of time and several times we thought to ourselves, “lets just do this in the terminal,” but we kept going and kept grinding on the GUI and got it to work.

There are a lot of crazy features we wanted to implement but we couldn’t due to lack of time. Some of them are themes, sound effects, and visual effects. During the early stages of the program, we thought to put in cool themes that the players can choose from. Some of those included a Star Wars theme, a dark theme, light theme, etc. Unfortunately, the GUI took too much time and we quickly abandoned that idea because we had other priorities. The same thing happened with sound and visual effects, but we still tried our best to make a GUI that looked decent. After a while we abandoned all the “cute” ideas and just wanted to make the game work in the GUI and make it not look bad and still be functional.

One thing we would’ve done differently is allocated time a bit better. All five of us are busy most of the times, thus making time to meet and work on the project was very limited. Even with the limited time, we were able to cooperate really well and were able to make the game decently; however, a bit more time and we could’ve made it even better. A part of the reason for this is the fact that how little experience most had with Java or Javafx, but after the project, we all were glad we went through this process because now all of us are familiar with this language and can put it in our resume. We all agreed that doing the project in Java was the best thing for us at the end of the day because it forced us to learn a language we didn’t know and features we didn’t know, so it made us better coders.

Overall, we had a lot of fun with this project, even though it took a lot of time and effort to get it the way we wanted it to be; however, we all did our part and as a team were able to finish this project. The program can run on both Mac and Windows, but when we tried to compile it in the lab stations, it did not because the lab computers do not have JavaFX.