Software Engineering Group Project
The End of Project Report

Author: Bhagya Wanni Arachchige [bhw]
Config Ref: EOPReportGP11
Date: 11th May 2022
Version: 1.0
Status: Release

CONTENTS

1. INTRODUCTION

1.1 Purpose of this Document

This document summarises the project as whole , the code, the functionality , the participation of the team members.

1.2 Scope

This supports the standards laid down for CS22120 Group Projects [1].

1.3 Objectives

The objective of this document is to provide a way to reflect on the completed project and assess what parts went well and what improvements could have been made

2. THE END-OF-PROJECT REPORT

2.1 Management summary

In summary, the project achieved to create a board game that consisted of some of the functionality that the client originally outlined along with documentation that depicts the process taken to plan, test and design a functioning program.

From a purely graphically perspective the program compliments the aesthetic of the board game 'Buccaneer' and manages to portray the game successfully as one involving pirates, ships, and islands. The game allows exactly 4 players to play and does randomly assign each player a home port, however, the positions of the ships are not as intended and can appear quite confusing. Treasure and crew cards are correctly distributed at the start of the game but there is currently only a way to view the current player crew cards, treasure held on a player's ship can be viewed if they are able to travel to a port via the trade menu. Ports properly implement trading, storing, and updating any treasure or crew cards deposited onto the port.
A major difficulty of this project that we faced during the project was the fact that was multiple coders working on the same program this led to complications as any code written needed to meet a global standard to work with the rest, we attempt to do this after every session doing a merge of the work done that day. Our solution whilst effective still led to some bits of code being lost in translation and caused some methods, we wrote to not be called entirely. The time constraint combined with this resulted in some functionality of the program being lost even though there were methods that could easily handle it.

In terms of documentation, the standard at which most of the documentation was completed is satisfactory, however, there are certain sections that may not be fully updated or complete for instance the design specification still lacks class and state chart diagrams.

The program can run without any noticeable game-ending syntax errors; however, it contains some bugs that may affect the user experience. The teamwork aspect was especially challenging during implementation week, but the team as a whole handled it very well, helping and guiding each other and also staying past the designated hours to try and improve our program. Taking everything into consideration I believe that the team performed very well, our team's small size meant that we were already presented with a disadvantage but despite this, I feel our team managed to complete a lot of the required functionality of the program.

2.2 Historical account of the project

Our group's approach to the project was to split the workload amongst ourselves from the start so that everyone contributed equally to the project. This was clear from our first ever tutorial where we divided up the tasks and

assigned those that had more experience to the appropriate task, for instance, James was assigned to JavaFX because he had prior experience. On the first week of the project, we also scheduled a weekly meetup to allow each member to debrief the rest about what they had been working on.

2.3 Final state of the project

Firstly I will discuss the parts that don't work as we wanted. Firstly, during testing the program we realised that when the game loads some of the players don't start in the right positions and names aren't passed through correctly, except from player 1, from the start screen to the game screen. Next, when it is a player's turn there is no way to view their own treasure unless they go to a port, the players, however, do store treasure there is just no way to view it. We also found movement to be a little temperamental, but it will move where it is meant to, the issue is that highlighting doesn't work diagonally and can be buggy going straight. Also, after a port or an attack, the player cannot move in any direction.

There are 5 chance ports that don't work as intended, the first two are when a player is blown to the nearest port and their homeport, these simply do not teleport, but they do the second part of each card. The next two are blown to mud bay and cliff creek, when these cards are received, they work but only if there isn't a player in the spot of either two locations. These two locations were also not visually represented on the game board but on the teleporting works. The last chance card that doesn't work is chance card one, this is because we didn't have time to implement it. When a crew card asks whether the player wants treasure or crew cards the treasure visual icon will always say 5 treasures, this was a rushing mistake, but the correct amount of treasure does go to the player.

When clicking on the next turn button you cannot click anywhere you have to click the invisible button within the text, this does work but makes the game not very fluid. When it comes to attacking for the most part it works but sometimes the wrong person will win, and the loser doesn't move. On-screen we have a ship in a compass to represent the direction of the current player's ship, sometimes this faces the wrong way, but you can still see what direction your ship is facing on the game board. When a player transfers their treasure to a homeport the safe zone doesn't work.

2.4 Performance of each team member

James

At the start of the project, James was the most familiar coder in our team with Java FX and therefore he was tasked with experimenting and prototyping with this software which he did quite well. During implementation week, we relied on James's code heavily and used it as the foundation for our final program. Overall, James performed extremely well throughout the project and was a crucial member of the team, explaining his existing code and helping the entire team throughout implementation week.

Ash

Ash was tasked with investigating java persistence, which he and James tested somewhat successfully. Unfortunately, due to our team's small size, java persistence was scrapped. Ash was assigned and completed a section in the design spec and played a crucial role in the implementation of the program as he created crew cards, helped with chance cards and redesigned the GUI of the program. Ash was very much so supported almost every aspect of the implementation of the program and his performance was excellent.

Maciej

Maciej worked alongside Mateusz to complete the testing for the project and acted as the deputy quality assurance manager. Testing for the project involved creating the test spec that Mateusz and Maciej created,

splitting the workload evenly amongst themselves. During implementation week he helped work on the Islands and created the trade screen for the program. Maciej performed well at the tasks he was given.

Dean

Dean and I worked on the UI specification, and he oversaw creating mock-ups of possible screens that acted as a guide when designing the program. Furthermore, as he was the leader at the time, Dean helped me formulate a plan to finish the project. Dean also contributed during implementation week and worked on treasure distribution, and chance cards and created various useful methods used throughout the program. I would say the Dean performed quite well in the project with the tasks he was given.

Mateusz

Mateusz worked alongside Maciej to complete the testing for the project and acted as the quality assurance manager. Testing for the project involved creating the test spec that Mateusz and Maciej created, splitting the workload evenly amongst themselves. During implementation week he helped work on the Islands and created the attack screen for the program. Mateusz performed well at the tasks he was assigned.

Bhagya

The first document I worked on was the UI document and mainly focused on creating the use case document. I was also tasked with completing a section in the group's design specification and then formatting and joining the section that the other members gave me. Heading into the implementation week I took on the role of the group leader and tried my best to make sure that every member had a task to try and balance the workload. I also contributed to the code focusing on implementing popups, describing the game to the use, and functional ports. In my opinion, I performed well throughout the project but there were some aspects related to my leadership role that I felt could have used improvement.

2.5 Critical evaluation of the team and the project

The team performed brilliantly considering our smaller size and therefore added pressure, however, the project was I would say that the project started off slowly which may have cost us valuable time, ultimately affecting our result. Nonetheless, once the pace picked up and the entire group caught up to the same speed the performance was very good.

The actual project description itself had some vague descriptions and we were left to make some informed decisions which may or may not agree with the client's expectations. One of the most important lessons that were made clear to me through this group project is that communication is key in any team-based environment. Communication or lack thereof can make or break a team, especially when relating to a software project where you as an individual are heavily reliant on the rest of the team.

REFERENCES

[1]   Software Engineering Group Projects: General Documentation Standards. C.W. Loftus. SE.QA.03. 2.5 Release

DOCUMENT HISTORY

Version	Issue No.	Date	Changes made to document	Changed by
1.0	N/A	11/05/2022	N/A - original version	BHW