Software Engineering Group Project
Final Report

Author: Wgf, Jaj48, Lie4, jor15, adl12

Config Ref: SE.G17.FR

Date: 2017_04_05

Version: 1.0

Status: Release

Department of Computer Science

Aberystwyth University

Aberystwyth Ceredigion

SY23 3DB

Contents

1 Introduction 3
1.1 Purpose of this Document 3
1.2 Scope 3
1.3 Objectives 3
2 Management Report 3
3 Historical account of the project 4
Week #1 – 2/2/17 4
Week #3 – 16/2/17 4
Week #4 – 23/2/17 4
Week #5 – 02/03/17 5
Week #6 – 9/3/17 5
Week #7 – 16/03/17 5
Week #8 – 23/03/17 5
Week #9 – 30/03/17 6
Implementation week 6
4 Final state of the project 6
5 Performance of each team member 7
6 Critical Evaluation of the team and project 10
7 Document History 11

1 Introduction

1.1 Purpose of this Document

The purpose of this Document is to show self evaluation as a group on the project.

1.2 Scope

This document specifies the design of the system. It will show the final state of the project, a critical evaluation of the team and project, a personal feedback for each member and a historical account of the project

1.3 Objectives

The objective of this document is to aid the implementation of the system. It will provide a summary for the project as a whole.

2 Management Report

The project achieved making a game that is in a functional playable state. The game can be fully played by 4 players. There is 3 known bugs in the game chance card 20 & 22 don't work and there is a bug when moving after and attack that doesn't let you choose a direction but this only happens in very few circumstances and we have only ran into this bug twice. Chance cards 20 & 22 can be disabled in the csv file by removing them making the game playable with no crashes but these 2 chance cards would not be useable. Fixes for these 2 chance cards are known and have been documented so with a small amount time tweaking the code the game could be fully working.

All documents are in a good state lining up with QA standards. The actual look of the actual look of the program has some deviances from the UI specification where we thought during development something different would look better. All other documents are accurate and are to a good standard.

One part of the project we struggled with was only having 4 people confident enough to program in Java. This resulted in almost all of the program being written by 3 individuals. We overcame this by assigning the others to do documentation, testing, module testing, image creation and other roles. Due to this good delegation of tasks this meant that the 3 individuals were able to just program and not have to worry about all of the other parts of the project meaning that we got the project finished on time. The team performed well as a group and all got on with each other. We had some problems with communication, with people being unreachable at times, but overall the group worked well together and completed tasks. There were some team member worked harder than others however everyone provided to the project. The group as a whole performed well and had a good variety of strengths through the group.

3 Historical account of the project

Week #1 – 2/2/17

In our first group meeting everyone introduced themselves, assessed strengths and weaknesses and got a rough idea of the section people wanted to be in. Photos were then taken of each team member. We set up a platform for communicating as a group. A face book group chat was established to maintain communication between the groups. We set up a blog system for each member in order to monitor each member progress through the development of the project. We then assigned some members the role of reading the SEQA documents, and the other members to start planning a rough architecture proposal, and start building a basic prototype of the game.

Week #2 – 9/2/17

In week 2 we started writing a test document, in preparation for testing after the project. This would later help us with the development of the project, as well as helping us find issues that we may not have discovered during normal development. We also set up our initial GitHub repository, as we did not yet have a SVN repo, and all of the team preferred GIT to SVN. A formal UI Document was also started on. We discussed how we delegate different parts of the program to different people. We decided to split people into GUI and System Architecture teams, so that those who were more comfortable with Graphics and JavaFX could be the GUI and graphics team, while those more comfortable with Data structures and program flow could work on the backend. We chose how we could make both sides of the program work with each other, using an event driven system. We took more notes on the quality assurance notes and the system Architecture team went into more detain on each areas that needed worked on, including the storage of objects within the game, and how different objects interact with each other.

Week #3 – 16/2/17

This week we assigned roles for each person in the team. we wanted roles to be well defined, but also for people to also experience a wide range of roles within the team. As a result, we had flexibility in the roles, allowing people to use their particular skills to the best effect. The test team continued with the test document, with users collaborating with one another in order to split the workload. The GUI Team started a use case document so that we knew as a group what we needed as a GUI, and the requirements that that GUI most had. The GUI team made a presentation outlining the use case document, in order to represent the use case in a visual manner to the rest of the team. The system Architecture team created a basic project for the game, and started creating basic classes and methods, whilst planning out how to do the more complex methods. We had a meeting later that week to discuss what has been achieved so far, and what areas needed more work. We also created a way of managing each person’s time within the group, so that the group leader can effectively manage each individual member’s time accurately.

Week #4 – 23/2/17

This week the tests were standardised, so that all the tests could fit into one document. The UI Document was finalised, and we had a review meeting. The System architecture Team developed the flow of the program even more, implementing states for the players, so that the game knew what it was doing at which stage of the program. The GUI Team developed another prototype of the GUI, but this time with the game implementing collision detection, and better movement, following the specifications. Github was also organised according to the QA document, with all the minutes and supporting documents, as well as the pre existing source code uploaded correctly and In the correct location.

Week #5 – 02/03/17

This week the design document was started.

The system architecture team presented a power point on how the game flows through the game logic, and how the classes within the game interact with each other to the rest of the team. A GUI team presented a demonstration of the GUI working with some of the game logic. The JUnits tests were started on, and an automated tested system was implemented, using a Continuous Integration system. This also checks that all code that are done compiles, and notifies the committer if any code that is pushed does not compile. Some people were tasked to implement methods that had been planned out but not yet implemented. The GUI team completed movement. The system architecture team continued with design of the functionalities.

Week #6 – 9/3/17

In week 6 improvements were made to the User Interface Document. There was also significant changes made to the testing documents adding more information ie error messages. The presentations were updated and improved to show the rest of the team how the interactions between the GUI and the back end would work and how they will come out like. More JUnits were completed and a lot of to-dos were done involving the code.

Week #7 – 16/03/17

On week 7 we had a meeting to discuss as a group some changes and improvements to the documents. The UI Document was improved to the meetings standard. We also improved the Tests document fixing a few small issues. The power points were improved for the relivent teams to the meeting standard. The testers also completed some of the JUnit tests which could test some of the code we had completed to the testing document. More Design document work was done before having a review meeting mid week. The programming team worked on ports, movement and move highlighting, islands, crew cards and some of the GUI dialogue boxes to display information.

Week #8 – 23/03/17

On week 8 we completed more JUnit tests and the people who didn't know how to do JUnits had to learn how to make and run them. We set a big list of JUnit tests to everyone as we was falling behind in that section. The backend team fixed a few bugs relevant to movement and crew cards.

Week #9 – 30/03/17

The Design document was completed and with all relevant parts attached. We assigned and finished the 3 diagrams for the document that was completed that week, which was added to the design document. These 3 diagrams were the sequence, the component and the object diagram. The testing team completed more of the JUnit tests whilst the graphics team focused on rendering most of the images and gifs for the game. We discovered and fixed a lot of bugs as well. The backend and GUI team started on trading.

Week #10 – 06/04/17

During week 10 we did a few more Junit tests referring to the testing document. We also arranged to have conference meetings during the holidays. The backend coding team worked on getting the trading implemented and get a main menu working.

Implementation week

The testing team completed all of the Junit tests, as well as testing the program as it was being completed. Issues were made by the testing team, and assigned to the appropriate member who was working on that area of the program. They also tested each new feature as it was completed. The JavaDoc was also completed, in order to increase the maintainability of the code. The quality assurance leader ran through the code and checked it to the quality assurance standards. During Implementation week the GUI team created images for a 3d game board and ships. They also implemented new and improved main menus and help menu to improve usability and give it a more defined professional look. The system architecture team implemented chance cards and got them all working. Both the system architecture team and the GUI team fixed bugs and issues that was picked up by the testing team. The island port and player gui was added to give additional information to give the user everything they needed with a click of the button.

4 Final state of the project

When the game starts the menu is loaded in which a user can type the names of the four players in the text boxes. If he or she does that and press “Start” button the game will check if the names are correctly and if they are then this window will close another window is loaded. If the names are incorrect then a notification window will pop up informing the user what is wrong. There is also a “helpin’ hand” button which loads up another window with the game rules and concepts explained.

The first Player is assigned randomly to one of the four Ports and other Players are then assigned to the next Ports clockwise. The movement works perfectly. In each turn Player is allowed to “move & turn” or just “turn”. After each movement Player’s new position is checked if it’s next to an Island, in the Port or in the Bay. The Player is not allowed to move inside of the Islands or outside the Game Board.

When the Player passes another Player a window will pop up and ask whether he or she wants to attack. Attacking can be also achieved by simply landing on another Player's ship. Attacking works well, it compares the attack strength of both Players and the winner gets the Treasure or Crew Cards. The losing Player is allowed then to move and turn in any direction up to maximum move points. One bug that we weren't able to fix, which is happening under - what seems to be - random circumstances, is when the losing player is allowed to move away, he or she cannot rotate and a message pops up that a move is invalid. The bug is hard to reproduce therefore we didn't know how to fix it.

When the Player arrives at the Port the Game checks if this is the owner of the said Port. If yes, the Treasure in the Player's ship is automatically transferred to the Port. The value of Treasure in the port makes up the Score of the owning Player. If the Player isn't the owner, then a trading window pops up and lets the Player trade Treasure and Crew Cards. There is no problem with that functionality.

The Islands functionalities are fully implemented and working. You can take the Treasure and Crew Cards from Flat Island, take a Chance Card from Treasure Island. Treasure Island also stores all the remaining Treasure. The Pirate Island is only for storing the deck of Crew Cards.

The Chance Cards are implemented but there are two that aren't working. Chance Card number 20 is giving an error when it tries to remove Crew Cards from Player's hand. The reason is a null pointer. Chance Card number 22 is not working because of the wrong instruction when Player has less than seven cards on hand (i. e. it should say continue; instead of break;).

5 Performance of each team member

Aaron

Aaron was a key part in the architecture, design and programming of our system. He and Jakub did the architectural design of the system and did most of the programming of the back end of the program. He had the role of deputy project manager and performed this very well he helped other in the team when they needed and was very good at helping when it came to problems with intilliJ. He wrote a good majority of the back end code of the game and was a critical member in getting the game to a working playable state. He performed all tasks asked of him on time and to a high standard and was present to all group meetings.

An improvement Aaron could make would be to comment his code with more useful comments so others working on the same code can better understand his thinking. Overall Aaron was very hard working and a very valuable member of the team, the project would not have been completed without him.

Aaron agreed with all that was said here.

Jakub

Jakub was a hard worker and provided a lot to the team. He helped Aaron with the architecture design and programmed a lot of the chance cards and. He attended most of the group meetings apart from when he had legitimate reasons. He completed all tasks that were assigned to him to a good standard.

Jakub could improve by being more vocal, we wouldn't hear about problems he has with work until we asked him being more vocal would allow him to work better in the team and be able to get more out of his abilities. Overall Jakub was a solid member of the team completing all tasks asked of him and provided some good ideas to the design of the program.

Jakub agreed with all that was said here.

Will

Will was the Quality Assurance manager in the team. He performed very well as the QA manager made sure all documents, code and java doc we submitted had the correct QA standards. Will performed all of our tests during the implementation week and found many bugs and is a large part of why the program runs with a small amount of bugs. He also wrote all of the Java doc for every method in the program.

Will could improve his value to the team by spending more time learning Junit tests and programming as while he was great value in testing and the documentation he didn't provide anything to the program. Overall Will was a very useful member of the team and was a very good QA manager making sure our documents and code were up to a good standard.

Will agreed with all that was said here.

Josh

Josh was a big part in how the UI looks, he wrote a large amount of the UI document and helped design how the UI would look and how information would be displayed to the user. He also provided toward the design document and produced our sequence diagram. Josh was a helpful member of team for doing odd jobs that needed doing throughout the programming and design stages, he was always happy to do a lot of the small tasks for example the help menu. During the implementation week he wrote Junit tests, helped make the help menu, and did the JavaFX for the aesthetics of the start screen.

Josh could improve by having better focus on the task he is working on at that moment. He would tend to get distracted by others work and want to help them instead of doing his own. Whilst the help he gave others was good and useful it would often mean the tasks he was doing would take longer than they needed. Overall Josh was a valued member of the team providing some useful ideas.

Josh agreed with everything that was said here.

Dean

Dean provided towards the testing document and the maintenance document. He was present to most group meetings before the implementation week and provided good ideas during it. He created a fair amount of the system tests.

His attendance during the implementation week was poor and for the first half of week had not managed to get intelliJ working and did not ask for help. He did work on the maintenance document during this week but provided nothing toward the code or Junit tests. When he did attended he spent a lot of time on his phone and we as a group would sometimes forget he was a part of the team. Overall Dean when he was present and working he was a useful member of the team.

Dean agreed with everything that was said here.

Liam

Liam was a large part of how the final game looks over all. He made most of the images for the game which were used to make the game look good. He provided some tests to the testing document. In the week before the Easter break he was awarded a yellow card because he hadn't done the Junit tests he was assigned to do for a few weeks. However he earned this off after the Easter break when he had completed the Junit tests and had created the main game board image. During the implementation week he continued to write Junit tests and create images to make the game look aesthetically pleasing.

Liam could improve by practicing coding more to as he didn't provide anything to the coding of the final program. Overall Liam was a useful member of the group coming up with ideas on how things should look and creating images to make them look good but did not provide much towards the technical side of the assignment.

Liam agreed with everything that was said here.

Adam – Written by Aaron (deputy project manager)

Adam was the main group leader for this project. He did an excellent job of managing the team efficiently, and always took into consideration the needs and availability of the team. He took into consideration the skills of each member of the team, and assigned each member tasks that they in particular would excel at. In addition, Adam took a key role with the GUI side of the development, and worked well collaborating with the back-end team to ensure that both systems are compatible with each other, as well as communicating with the other member of the GUI Team. He put a lot of work into the GUI, and produced a functional GUI as a result.

An Improvement Adam could make is trying to make his code reusable, which is especially useful when working with GUI tools, as some of his code is recurring boilerplate code.

Adam agrees with everything that Aaron said here.

6 Critical Evaluation of the team and project

Critical evaluation of the team and the project

How did the team perform as a whole, and how could that have been improved?

The group as a whole did great and there were many good ideas that was put forwards, which most got used. We planned most of the architecture and the GUI at the beginning with presentations where we got the groups feedback which played a real critical roll in the later stages. Our group also used event system driven. This was done by having the backend and GUI working on the code, once completed they would hand it to the testers to test and if failed it would go back to the relevant section to be checked again. This caused little room for error or mistakes. We also set rolls based on each team member's strengths and abilities at an early stage. This was good as we from the start had formed our own little groups and started pitching ideas from there sections.

Our group could have been improved by slightly better communication. We had one planned group meeting a week where we could have benefited from 2. We could have also met up more in our groups and done some group coding rather than coding by ourselves. We could have also improved by starting the project a little earlier and doing more in the early stage.

How could the project that was set for you have been improved

The project provided to us could have been a bit better if it was based on a newer board game mainly because it would have given us as a group more of a chance to stand out from the other the groups because a lot of the group projects will be extremely similar because of the limited options we were given. The project could have been so much better if as a group, we had been given 2 weeks for implementation instead of one so that one week could have been used for last minute bug fixes and time to improve the quality of the documents and code and then the last week could have been used for adding small things to the game for example better graphics and more sound effects.

What were the most important lessons learned about software projects and about working in teams?

Understanding each team members skills would probably be the most important lesson learnt on this group project. This allows for the group to set smaller groups and assign roles based on member's strengths to get the best result. Time management is also important. You may need to give a team member more time for certain tasks to get a better result and trusting your team members to deliver on time when it is needed rather than waiting around stopping the whole group from proceeding.

7 Document History

Version	CCF	Changes	Changed by
0.1	N/A	Initial creation	wgf
0.2	N/A	Added content	Jaj48, Lie4, jor15, adl12
1.0	N/A	Final Checks	wgf