Group 14 - End of Project Report

Authors: Thomas Wilkinson [tjw21]
Config Ref: SE.GP14.EoP_Report
Date: 11 May 2022
Version: 1.0
Status: Release

Department of Computer Science
Aberystwyth University
Aberystwyth
Ceredigion
SY23 3DB

CONTENTS

1. MANAGEMENT SUMMARY

In this project we have created a version of buccaneer which adheres to all functional requirements. It is made for exactly 4 players who all have to be given names which are within the set character limit.
All the players cards and treasure are kept track of, this caused some difficulty when dealing with chance cards which had a specific function beyond the player gaining or losing treasure and crew. One of the main issues we had, which was sorted late in the project, was adding the chance cards which are kept by the player in their inventory.
Setting a limit on the amount of treasure the players can have in their ship at any one time was simple to implement and didn't cause any issues.
A lot of work went into the UI with several team members putting time into making sure it clearly displayed the game state accurately.
It took us a while to figure out how the treasure was supposed to be distributed between ports but once we worked that out we managed to implement it. Assigning the ships to ports and dealing crew cards to players were both done without any servious bugs we found. We had an issue with creating and setting the start position of the ships as sometimes 2 ships of each colour would be created at each player port. However, we did manage to solve this.
Collections.shuffle was used to randomise the order of the crew card check while they were being added. After this they were removed as they were sent to the starting points for set amounts of crewcards. As the chance cards are all individual sub classes which all inherit from one superclass they have to be added to the deck one at a time but once they have been they are randomised in the same way but are all kept in the 'deck' at the start of the game so dont need to be dealt out.
Implementing the players reaching islands/bays took some work as we first had to make sure that the game recognised the islands and wouldn't allow players to go through/on them. After this, we had to add the specific features of each island/bay such as getting chance cards by reaching treasure island and gaining treasure if the player reached anchor bay and had the 'Kidd's chart' in their inventory.
Trading at ports was one of the later things we implemented but it works now as is required for the game. Items are stored in ports in 3 seperate array lists for chance cards, crew cards and treasure.
We display legal moves to the player using a greyed out ship on each possible square they can move to.
Implementing the attack rules was a case of calculating the battle total for the ships and comparing. After this it was a case of removing the treasure from the loser and giving the winner the option to take as much as they could. We later also later implemented the rules that players can't be attacked at treasure island or in a port.
Adding the win condition was the last major part of the project and involved checking at the players turn if they had more than 20 value of treasure in their port. If this was the case they were the winner and the game ended, displaying the win screen.

Overall, for the code our project has achieved everything we set out to do and is as complete as we hoped it would be. I am also happy with most of the documentation, while we had to make alterations to them many times based on feedback or changes to tests and design ideas they are thorough and explain all parts of this project and our group's process well. I believe our team has performed well, we have had challenges with communication at times but I believe that all in all we have worked well as a group and almost everyone has done their part in creating a final software that we are all happy with.

2. HISTORICAL ACCOUNT

Week 1 - All group members began reading required documents. The repository and maven project were set up and some tests were drafted. Example game screens created.
Week 2 - Created testing specification document and UI specification. Made a basic UI and began designing the final UI. Began researching JUnit. Added chance cards and port objects to the project.
Week 3 - Continued working on tests and organised the repository, created object loader class and implemented FR1-FR10. Worked on classes based on feedback in tutorial. Added the typical users to UI spec and created a schedule for the deliverables.
Week 4 - Added more tests to the test specification and created a dev branch to store each week's work in order to help with quality assurance and versioning.
Week 5 - Added more prototype tests and worked on some of the java fx. Documented code and prototyped the fight system. Reviewed the test spec and created the document for the design spec. Planned actions for UI specification based on feedback from Chris.
Week 6 - coded tests, worked on the design spec. Worked on the UI specification actions, worked on username input screen and FR14-FR15.
Week 7 - Created diagrams for design specification. Worked on the test specification and design specification. Added dependency description, significant classes, programs in system and worked on back end issues. Developed new board scene and improved screen.
Week 8 - worked on the design specification and fixed the document formatting for all documents to make sure they all matched the template. Added more test code and drafted more pseudo code.

3. FINAL STATE

In this project we got almost everything working; however, when it came to testing in integration week we found a few things that didn't work quite as we wanted them too. The amounts of treasure that were counted by the tests were not what we had expected. Also, the amount of treasure counted on Flat island is not consistent when tested. All of these tests however do work when run individually; however, when they are run all together they do not. Also, the player can change direction to one where they cannot move more than one square as we did not implement it. The restart button was not implemented either but termination is allowed and the game can be restarted by pressing play again.

4. INDIVIDUAL CONTRIBUTIONS

Jay Staddon(jas130)

As the deputy project leader he helped me in organising the group and distributing the work. He also stepped up and kept things running smoothly on the few occasions I was ill or otherwise unable to attend meetings.

Early on in the project he did a lot of work on the UI specification and created the Presentation meant to show the different game states. When we began working on the actual code for the project, Jay created the system for checking the player names that users entered and helped with some of the chance card tests as well as bug testing the game. He also wrote out the set of instructions we included. Also, he worked on sections of the design spec and did a few quality of life changes to the project such as displaying to the user what is happening in the code (with chance cards and attacking). Jay agrees with what is said in this statement.

Hancheng Zuo(haz15)

Eric did a lot of the work on the UI. He coded the UI interfaces using Javafx and added some of the touch effects for buttons. He also added some of the image assets which he resized in photoshop. He coded all the FXML files and implemented the associated control functionality in the controller classes. He worked on the CSS, participated in the board testing and recorded testing results with screenshots. Eric agrees with what is said in this statement.

Ali Bajwa(alb113)

In the early stages of this project Ali did little work, despite attempts by many members of the group to give him work to do and offer him help if he needed it. However, I understand he was having family issues outside of the course and this should be taken into account. Towards the end of the project, he created the UML case diagram and component diagram for the design spec and helped with bug testing in integration week.

Thomas Wilkinson(tjw21)

As project leader I attempted to keep the group on track throughout by giving members specific tasks to do and helping with them when I could. I created schedules for the group at several points in order to make sure we got things, such as document reviews, done on time with enough time left over to make any required changes.

I worked on the UI specification as well as the design specification. I created most of the documents for the project and, after feedback from Chris on the UI spec, made sure they adhered to the template. In the code I helped with some of the chance cards and debugged some of the issues we had with them as well as fixing a lot of the warnings in the project in later stages and bug fixing some of the css. I also helped with bug testing in integration week and checking that all the required features had been added correctly.

Sebastian Sogan(ses35)
Sebastian set up the original java fx project with maven so everyone could develop the code on all platforms. Throughout the project he mainly worked on the backend functionality of the code, as well as contributing to code for the user interface/tests in places. This included designing the data models and their base classes/interfaces to enable their integration into the project.

In the backend he additionally wrote the object loader for handling the correct loading of classes which covered functional requirement 1-7. The game loop for tracking game state (e.g.. ship turns, win conditions, trades etc..). The map square class I produced was utilised as an interface between the frontend and backend to manage correct updating of board objects. The save data class was used for storing the essential board data as compressed data so it can be restored by the object loader. He implemented chance cards with their base case.

For the user interface he made contributions in BoardController for updating ship position and direction. Additionally, he implemented the inventory loading system for ships and islands using a custom image view class. He designed this class to be passed a commodity and automatically initialise a JavaFX image view with the correct image. Sebastian agrees with what is said in this statement.

Kacper Prywata(kap48)
As QA deputy he helped take care of the structure of the repository, added git ignore files and moved files to the correct folders. Added files to the dev folder for each week's progress. Checked the testing documents for the QA standards.

He worked on the test spec, wrote most of the back end tests and implemented unit tests. He helped to choose UI tests, worked on the test report and bug fixing in the late stages of the project. Kacper agrees with what is said in this statement.

Jac Lingard(jal74)
As quality assurance manager, Jac mainly worked on the documentation aspect of the project. They were responsible for going over any documents and checking they were up to the standards given in the QA documents on blackboard. They also had the responsibility to make the minutes files for all the formal meetings every week.

Jac also maintained the repository and made sure things were where they were supposed to be in order to comply with the QA documents. They also had to create a file (config_refs) that would keep track of all the documents in the repository and where to find them as well as other information.

They worked on the test spec, design spec, test report and maintenance manual. Jac also needed to continuously go over all the documents in order to make sure they were structured properly and weren’t missing anything from the general documentation standard. He was also responsible for working on the dev files and making sure the work was put there before being released.

Hassan Farooq(haf22)
In this project Hassan's main role was testing. Early on we had some discussion on how we would implement some of the FR's and what the best ways will be. He drafted some tests and thought about how we would test the system.

He helped in creating the testing specification, testing report, test code and parts of the maintenance manual, and also code cleanup for tests, He also tweaked/fixed some chance cards to make them work correctly. He helped choose which tests were going to be coded and which were UI tests, for the main chance cards he did most of the UI testing with help from Kacper and Eric. He took screenshots for the chance card tests to show as proof for passing. Hassan agrees with what I've said in this statement

Dylan Murphy(dym27)
Dylan mainly dealt with the production of the front end. He produced a mock GUI using JavaFX based on the design done by Jay. He implemented a programmatic approach to creating the grid and allowed for it to be occupied by icons that represent players. He also colour coded the players, which meant that it was far easier to see which player was which, and also let the players more easily see whose turn it is.

During the later stages of production, he improved on the overall look of the game by adding graphical islands and added a "combat icon" to make it easier to see when players will engage in combat. As well as maintaining the game board, he also helped production of the game by adding in dev tools in the form of automatic name entrance when debugging the game; this generally aided production as it meant that testing features of the game no longer required for you to enter 4 different names.

5. CRITICAL EVALUATION

I believe overall we performed very well as a team. At first it took some time to get used to working together but after a short time we began working to our strengths in the team and finding what each member was best at. Later in the project when everyone had their own role I feel that communication wasn't as good as it could have been because everyone was so focused on what they were doing individually but we managed to make sure everything was done on time to a standard everyone was happy with.

The project could have been improved if we were given a clearer set of rules for how the game was to work in the end as I feel there was a lot of ambiguity between the requirements we were given and the rules of the game as set out in the original.

Some of the most important lessons we learnt about software development were about how important planning and time management are in a group project. Also, having good leadership to make sure everyone has a good idea of what stage the project is at and what everyone needs to do.

6. DOCUMENT CHANGE HISTORY

Version	Issue No.	Date	Changes made to document	Changed by
0.1	N/A	05/05/2022	New Document	TJW21
0.2	N/A	07/05/2022	Added the management summary	TJW21
0.3	N/A	08/05/2022	Worked on some of the individual contributions	TJW21
0.4	N/A	10/05/2022	Added the historical account	TJW21
0.5	N/A	10/05/2022	Added more individual contributions	TJW21
0.6	N/A	10/05/2022	Added final state section	TJW21
0.7	N/A	11/05/2022	Added critical evaluation	TJW21
1.0	N/A	11/05/2022	Reviewed all sections, worked on structure	TJW21