



COMP30050

Software Engineering Project

Final Report

“A creative variant on an old reliable”

/* THIS IS A TWO MAN PROJECT BETWEEN ALAN AND NIAL. */

Doctor No's

/*Alan Doyle (team leader), 14401758*/

/*Niall McGuinness, 15335796*/

Joel Jacob, 13319401

Daniel Graham, 15319536

Table of Contents:

i) Introduction	2
ii) Approach	3
iii) Implementation	5
iv) Work Breakdown [<i>Gantt Chart</i>]	6
v) Individual Reports	7
vi) Group Collaboration.....	9
vii) Technical Challenges.....	11
viii) Testing and Evaluation	13
ix) Final Design	14
x) Conclusion	16

Introduction

“Like many businessmen of genius he learned that free competition was wasteful, monopoly efficient. And so he simply set about achieving that efficient monopoly.” - *Mario Puzo*

For our 3rd year Software Engineering Project, we were assigned the task of creating a virtual rendition of the classic Monopoly game. We were then split into groups of four, and given a time period of roughly 5 weeks to complete our assigned project, in which we could design and create an entire Monopoly game, to our own design specifications. We were then advised to marry our game with a theme, one from famous pop culture references, and place a template of that reference on the game. The game was to be designed in the Java programming language, and we were given the necessary resources through online sources, the course teaching assistants, and information provided through the lecture slides each week. The task, while it seemed to be arduous, will be of major significance, as currently there are no major Monopoly games online for a consumer base that has been loyal Monopoly players for generations.

This is our final report for the project, detailing our successes and failures encountered, as well as reasoning behind the decisions made in the pursuit of our goal. Coupled with smaller reports from each team member, this report details our journey from idea to implementation, and from challenge to code.

Approach

Our initial approach to the project was simple, we research what monopoly is, we plan what we want to do with the project and how we are going to do it as a team, how we are going to work as a team and also setting the deadlines for each sprint of the game to help ensure that the final outcome is as desired.

From the previous years, we learnt many things from the previous Software Engineering modules as to what good Software engineering practices are, and how they could be implemented and integrated into projects, and as such, we endeavoured to draw on this knowledge as much as we could in the completion of our project.

We drew on this experience and implementation to bare the ground for our own version of Monopoly. Concurrent with the information received in the lectures, we began our research and development in order to get our project working as quickly and as efficiently as possible.

To achieve this, we knew we had to work as collaboratively as possible, and keep communication frequent and pervasive. We knew it would be a challenge completing this project, to our own specifications, as well as to that provided by our lecturer, Tony Veale, as we knew we had to prioritise the time to get our code up and running by the due date.

Based on experience, Alan had the idea of using Agile methodologies, in having weekly meetings to keep up to the date with the work each team member was doing, as well as the overall progress of the team. We knew we needed the time and space to work together, so we vacated to a free room often, to make use of whiteboards, and map out our path for completing this project.

We coupled the weekly sprints, with collaboration on Trello and Github, to make our lives and respective jobs easier. We understood the only way we would ever get the work completed in time, was to keep an open mind as to our approach to the project, and be ready to share our thoughts and progress as often as possible, simply for the nature of a team project.

Committing to our sprint deadlines, coupled with any other outstanding and ongoing work, would prove to be a challenge in itself, but one we were all ready for, and the confidence in overcoming this challenge, was evident to see.

As the project went on, issues began to arise in the form of a lack of communication, as well as false promises for code completion. This began to slowly halt overall progress, leading to us revamping our initial spec to fit in with these circumstances. Along with this certain individuals for personal and various other reasons only let myself and Alan know that the certain areas that these individuals were working on weren't complete and that they would like to leave the group last minute, which led to myself and Niall arranging a meeting with Tony to discuss how we could reapproach the current issues that arose within the group and how we were going to deal with it as a brand new team in such a short time frame.

After the meeting with Tony, we sat down together and developed a whole new approach to this project; we decided to park outstanding work in order to prioritise getting semblance to a Monopoly game up and running. We achieved this by setting meetings every day, and in the event either was unable to make it, we arranged Skype or Facebook chats to ensure we were both on the same page in cleaning up our project as best we could, so as to develop a working program based off the code we had previously implemented.

Implementation

The initial aim of the project was to use this vast database to create randomised content which we could invariably use in order to create a unique feel for the game. However, this proved too much, and we were unable to integrate the database into our project, instead being forced to skin our game in a chosen static theme. This proved to be a significant setback, owing to the fact we were looking forward to the procedural content generation, as well as it being prefixed as a key part to the project.

Our GUI consisted of a number of JFrames, JPanels and JButtons, as these proved to be the most effective way of portraying the game to the player. We combined these with functionality to control how the overall collective information was presented to the user, labelling a generic currency to allow for freedom in design. The underlying logic comprised a number of classes, linked together using object-oriented paradigms and principles. This was done on the basis of past experience, to ensure a more robust code layer.

We also used the NOC list given by Tony, which contained a wide variety of known popular movies, tv-shows and books, and listed attributes such as characters, whether real or fictional and the locations in which the movie, tv show or book was based on. This meant that all the groups from COMP30050 had a fantastic opportunity to design their Monopoly game, from the board to the Chance Cards, as part of a theme from the pop culture reference. We were given access to the NOC list repository on Github, where we could access the files, and we used hash tables to control the way the data interacted with the board.

Each class comprises specific and relevant logic to control how the game is monitored, including what features are present; this was done by making use of the various interfaces available to us - including creating our own interface and enumerations for usage in handling various game states, as well as tile typing. The idea was to use these game states to control some of the logic as well, enhancing our program, but it is an unfinished feature.

Individual Reports

“As the team leader, I knew I had a responsibility to ensure the project was firstly to be completed to specification, as well as being completed on time. This was to ensure that we had a project we were all satisfied with, one that worked to our team spec, as well as that provided by Tony.

Our first step was to decide on the path we would take in order to achieve this specification, and we attained this through open and collaborative communication. I knew it was the only way we could get it working, and from my experience of the module last year, I knew it was a method that would work.

Through much deliberation and suggestion from the team, we settled on an idea of how to make our game work and stand out, and we built from this foundation. I feel our weekly meetings proved invaluable as we could chart our progress, and build upon completed work in preparation for the next meeting.

I worked on the Board class, as well as the classes that handled each individual tile. Drawing inspiration from autonomous research, as well as my own knowledge of the game, I began drawing up my plan for completing this task. Using the various interfaces given to us through our lectures, I began implementing the Property class, building upon what I had from the lectures. Using this, I formulated the rest of the class, creating variation where required, such as the Railroad and Utility class. I endeavoured to update the rest of the team with my progress, which proved to be slow at times. Eventually, I completed the tile implementations, including methods I knew would have to be completed in the future; this was to ensure I covered every possibility the game could throw at us.

As well as this work, I made sure to keep informed as to what work the rest of the team was doing, advising them where I could, and listening to their ideas, paying dividends on our way to completing the project as best we could.

Taking charge of this project was my first experience in doing such, but I enjoyed the challenge. I found it tough at times, and the project unfortunately took a major hit towards the latter stages of the sprints, but I feel our best effort was made given we were down, but most certainly not out.” - *Alan*

“When we began the project, as stated before, we decided on fundamental elements of the projects that people would work on based on the level of expertise we all had in the subject. I created the group file, which was available on Google Documents and ready to be edited by the team members. I also began working on the implementation of the NOC list that was compiled by Tony to introduce a theme of a famous pop-culture reference that would encompass the entire Monopoly game, down to the properties, characters, and other features of the game. Unfortunately, due to prior engagements and a slight error of my time schedule, I did have a delay when compiling the code necessary for the program, and as such, I was unable to complete it.” - Joe/

Niall:

I was assigned by the team leader Alan to get the core mechanics of the game developed and some of the basic classes for the game. Early on in the project we decided that I would create basic player, property classes etc. and as the timeline of the project progressed with people doing their parts of the project such as the development of the NOCList class, I would then integrate their classes to the other ones that were developed by the other members and then create a running file for the project.

The game would run off the Board class in the game with some of the methods being the most important aspects of the game, such as the card, property actions, starting the game off, getting the number of players, their info etc. Also initialising the bots and UI for the game.

As time went on and the lack of communication from some of the members the timeline of the project fell closer to the submission date. Myself and Alan both worked on trying to get the whole game to a state where we could then work together on the project and produce a game that we both imagined at the start of the module. However as mentioned in other parts of the project this couldn't happen. So my new job was to get a NOCList working and also to work with Alan in developing all the features that we could and to try and figure out what we had to scrap in order to get a functioning game. Through constantly meeting and chatting with Alan we worked on these classes such as Properties, Players, PlayerStart etc. and of course a NOCList.

Group Collaboration

As a group, I feel the collaboration between the individual members was initially quite strong. As with any project, there were members more communicative than others, as some people like to work in silence. Although work was completed and our weekly sprints adhered to, it did pose a bit of a challenge, especially when communicating through the various mediums we assigned ourselves to. This was notable in the Facebook chat, for example, as it was hard to find a time when everyone was present and online, so progress updates weren't always as quick as we would have preferred, but we made do.

The usage of GitHub and Trello paid dividends as well, as it allowed us to keep track of what was currently being worked on and by whom, not to mention what work was left to complete, as well as any potential ideas that could help our game stand out. These mediums oft made up for the occasional miscommunication or lack of, and it helped us stay on track and meet our deadlines, hard or soft.

As team leader, Alan took it upon himself to talk to the individual members of the team, communicating updates should a lack of online presence arise. This kept the focus, coupled with constant ideas and suggestions on what to do next, and as such, allowed the team to overcome the challenges they were faced with in the collaborative aspect of the group. I felt that the delegation that Alan produced was exemplary, and it gave us the motivation we needed to progress along with our sprints.

Towards the final few weeks of the project, there was a noticeable difference in open collaboration and communication from some members of the team. A basic game was developed but a lot of vital work that was expected from other members of the group wasn't completed or uploaded to the Github repository in time and as it drifted off, it made for a lot of guesswork. This proved an issue as the workload shifted onto other members, where certain class implementations would have to be done by another person on the team, thus increasing strain and stress for all the group.

The group was working steadily and getting along fine, until those last few weeks. Individuals of the group were sat down and were given a piece of work to do and after a couple of weeks of being told that the work is being done and more, Niall and Alan were extremely disappointed to be let down by two of these members for various reasons.

Although Alan and Niall did what they could in the limited timeframe to complete the Panopoly game. We feel that, considering the outcome of events that happened and the functionality that we had to leave out, since we had to cut corners to accommodate the time limits, we are overall happy with the project.

Technical Challenges

Below covers the technical challenges encountered by members of the team.

“When attempting the assignments preceding the initial project implementation, I found understanding interfaces to be a bit of an issue. Having never used them before, I knew they would be paramount to the success of our project, so I endeavoured to research interfaces in an attempt to upskill myself, allowing for a stronger project. After careful study and some trial and error, I finally got to grips with interfaces, so much so that I decided to create an interface for usage in the various tile classes.” - *Alan*

“Whilst designing the classes, I found myself encountering instances in which I had to implement classes using methods that were not yet implemented in any class, even those done by other members of the team. I found this to be a bit challenging as I had no reference to formulate code for the methods so there was a lot of chopping and changing of code required to overcome this. It also helped that I made a list of the methods which satisfied the condition of having not been implemented, allowing me to delegate them where required.” - *Alan*

“During the process of collaboration, I found the implementation of Github as our shared workspace a little complicated as I was unfamiliar with the process of commits and just the general layout of the portal. Therefore I decided to brush up on my Github knowledge by following their online resources, familiarising myself with github with my frequent visits to the website, and using tutorial videos when I got stuck. I found it very helpful and gave me the understanding about Github that I needed” - *Joel*

“While designing the implementation of the NOC list, I realised that, as shown by Tony in his example, using hash tables would be the logical method to control the randomisation of the data used in our game. I therefore decided to conduct more research into the functionality of hash tables, along with my pre-existing knowledge on the topic itself, to understand how to introduce them into the game” - *Joel*

“One of my main challenges when it came to the project was completing the individual tasks I was assigned at the required time. Due to my other commitments to modules in my current year, and especially since some of them involved graded assessments in those modules, it was a challenge to balance my timetable to get everything done. It was even worse during the final stages of the project, as I had to start my study sessions for my end-of-semester exams. However, I decided to take a full week to dedicate myself to all the tasks given to me in the project and I found out that this gave me the time I needed to complete my existing deadlines, ensuring that we had overall success in our Monopoly game” - *Daniel*

“The main challenges faced during the course of the problem was the overall teamwork and communication. One of the challenges was creating a NOCList for the project. I was given this challenge very late in the project and I had to dedicate a lot of time understanding what the NOCList was and how we could implement this into our game to make it more dynamic and enjoyable as it gave users a unique feel to the game having new tiles in every new game played.” - Niall

“Having previously overcome various challenges in our journey to completing this project, the two of us were hit with the major problem of refactoring our entire codebase to fit in with a new spec and approach, which resulted from a meeting with Tony, our lecturer. This was given by the fact that two of our team dropped out last minute, so the both of were left wanting; this proved a major challenge as we had an extremely limited timeframe in which to develop the barebones of a Monopoly game, as basic as it could get. We knew it would be very ambitious to get anywhere near our initial spec, but having the foundations for the project allowed us to jumpstart and somehow pull a project consisting of a very basic, but working Monopoly game out of the bag” - *Alan and Niall*

Testing and Evaluation

As a group, we decided that it would be best to leave the majority of the testing implementation to after we had finished the core game mechanics and logic of our project except for basic things that we could test such as the Dice class and roll. Upon completion of this condition, we would use the techniques picked up over the course of the module, as well as those picked up through autonomous research, to facilitate our testing.

Using JUnit, we would begin to test each class in turn, writing enough test cases so as to cover ourselves should an issue arise with our classes. This was done in an effort to maximise our testing, concurrent with designing clean code for use in our project.

This delay in initial testing allowed us to formulate the foundation for our project, with due focus on the completion of the initial spec. However, this focus would invariably shift as two of the team pulled out due to various reasons, leaving myself and Niall in the position of refactoring the entire codebase in order to fit in with our new spec and approach. This resulted in our test classes becoming obsolete, and we didn't have enough time to redo working test classes, and as such, we decided to leave out the test classes altogether and trust in our program to work as expected.

Overall, we were relatively happy with how our project turned out. Myself and Niall pooled our resources together and kept up constant communication, as was the only way to get the code up and running as best we could so as to maximise our grade for the module.

Given our unfortunate circumstances and how approached the work, as well as our mindset in doing so, it was our opinion that the project was coded to the best of our ability, and are open to keeping communication alive post project, in order to further improve the project itself.

Final Design

For the initial final design of the project, we decided to bring to life a simple, easy-to-play and therefore easy-to-configure Monopoly game, that involved human players. We were to choose an iconic pop-culture that composed from Tony's prepared list and model our board, characters and other elements of the game around this chosen world.

The idea for our final design seemed simple in theory, but proved to be tricky and challenging to implement, as unbeknownst to us, it would involve a lot of back and forth code manipulation. This put a small strain on the overall project but we were determined to see our idea through. As a group, we decided on a relatively simple GUI consisting of three panels; one for the relevant player information, one for relevant tile information, and one panel for the actual game board. It was through these panels that we would control most of the internal activity of our project.

Despite two of the team members dropping out, leaving Alan and Niall to pick up the pieces, the base code for the GUI was almost fully completed; however, several changes had to be made to fit in with our respect to . Our initial idea for the GUI was simple. We have three JPanels spanning to the right of the JFrame, the three panels being PlayerInfoPanel which would show the players info taken from the player class and display things such as the players name, balance, number of properties owned, the number of utilities owned and the current position they were in. This information would have also been highlighted also to indicate whos turn it was.

The other panels would've been the CommandPanel which would have contained a panel consisting of six buttons which would perform that specific action such as roll, buy property, mortgage property, build, Auction and Trade property. The InfoPanel which would've displayed information about the property such as the name, price, was it buyable not, was it owned etc. Taking sample images found online, we would attempt to include the images of individual dice rolls, based on the values a player would roll; a nice twist, simplifying our job.

We also looked into creating save states for the game, allowing players to come and go, should they see fit. This was to ensure no player would feel pressured into finishing a

game once they started. Our initial aim was to create a comfortable environment for the player, allowing them to play at their leisure, whilst also diving deep into a game that would challenge and entice them into coming back for more.

Having implemented a number of interfaces, functions etc., during our weekly assignments, we took it upon ourselves to include as many of them as possible, so as to create cleaner code.

However, we had to abandon much of our initial plan due to unforeseen circumstances, which would result in losing connection to the various interfaces initially implemented. While the code remained, the lack of teamwork and communication towards the tail end of the project ensured that we would struggle to fully implement everything into a feasible, working solution to our project. We knew that we would need to get at least a basic game up and running, so at least we could submit something worth evaluating, and so we as a group made the decision to sacrifice a lot of the functionality to ensure we had a working project at the very least.

Conclusion

Unfortunately, things didn't go our way. We started off well, having a clear idea of what was required as well as what we had to do to achieve our goal. Communication and collaboration was strong, allowing smooth and continued progress with an overall end goal, structure and features being set out for the game.

However, as time went on, it became clear that obstacles would prevent us from achieving our desired state of project.

On one hand, a lack of communication between team members, despite best efforts from people in the group to try and push these members to get their tasks done and to communicate with the team if any issues or problems came up with their tasks so that we can pull together as a team and figure it out.

This proved to be a major pitfall as the lack of progress updates meant that members had to cover, leading to an imbalance in work done. This put undue strain on the pre existing copious workload, negatively impacting the overall project as a whole. It meant that key features that we wanted for our game and even basics one could not be completed on time, thus forcing us to change our game plan and deadline schedule by unnecessarily cutting corners.

On the other, a lack of semblance of work done proved another pitfall, as the cover required increased yet again. That being said, the majority of classes required for a basic game were indeed coded and linked up. This allowed for some breathing room, as we knew by the latter stages of the sprints that we would not have enough time to implement additional functionality so we shifted focus to the basics.

No doubt the overall game plan suffered, but given what we had to work with, myself and Alan believe a reasonable project was done in creating semblance to a game of Monopoly. This was primarily down to the determination of certain members in finishing out the work that had been done.