Cs319 Term Project
Final Report

Section - 03
Group - 3E

Coronapoly

Group Members
1. Osman Buğra Aydın - 21704100
2. Ertuğrul Aktaş - 21802801
3. Muhammed Doğancan Yılmazoğlu - 21801804
4. Oğuzhan Angın - 21501910
5. Mehmet Alperen Yalçın - 21502273

Table of Contents

1. Introduction

In terms of the main functions of the game, the game is near completion with minor bug issues that will be finished by the time this report is uploaded. The UI, game object classes, mechanics like buying, selling, bots, bankruptcy, quarantine are working fine. The only part that completely lacks in the game is the save file system. This feature was planned to be available in the final version but in the implementation stage it was decided that it would take too much time that would eventually jeopardize the completion of fundamental mechanics of the game.

Another feature of the game that is not fully working is the settings window. The users can not change the size of the window and the volume of the game music as it was planned.

Lastly, there exists minor changes in the object class diagram of the game in terms of properties and attributes that will be reflected in the final game presentation. It should be noted that the changes made in the class diagram are not major, that no new classes are added or removed but their small parts are modified to make the whole system work.

2. Lessons Learnt

The Coronapoly project has been a great experience for our team members in terms of both academic gains and team-wise skills. This section will talk about the lessons learnt over those two categories.

All of the computer science students have completed a java project at the end of their first year as part of CS102 class that could be compared with our current project in terms of the depth of the project. The academic lessons learnt through CS319 is massive compared to the project done in the CS102. We have learnt to properly design a project rather than code things until it made sense. The planning aspect of CS319 has taught us the importance of dividing big chunks of work into smaller parts that eventually fastens the process of implementation. The reports we have prepared were time consuming but while doing the reports we could picture the same project

without even doing any coding. Especially the uml diagrams we learnt and prepared, helped us to understand exactly what we needed to implement rather than learning on the process of coding. After we were done with our reports, we were aware of what we needed to do in order to complete our project. For us, learning the efficiency of planning ahead was the greatest lesson we could learn from this class. In terms of the tools, we have learnt many new concepts. Although all of us heard of git, intellij, javaFX, we have become good at using them while doing our project. We have also used tools such as Asana to do planning which most of us never used before.

In terms of team-wise skills, it was a remarkable experience. Although CS319 has a considerable workload compared to other classes, due to the large number of group members we have accomplished each and everyone of them on time because we were constantly communicating to get better. At the start of the group formation, we have decided that it would be better to have a project leader to lead and push us in times of need and we were right about it. We have decided that Osman would be a good project leader due to his social skills that would motivate all of us. Osman divided us into sub-groups that worked on small things like drawing different UML diagrams or coding different classes. This way we were reporting back to Osman and he was connecting the sub groups' work into the whole project. We have learnt that not everyone needs to be in contact with each other while doing something as some tasks could be done with one person. Tasking 5 people with one paragraph of work would only deduce efficiency of the group. Additionally, we have learnt that doing regular group meetings on zoom with our cameras open would strengthen our team skills as we both talked about the future tasks and other stuff. As we get more and more involved in the project as a team, we started to get more efficient in our work.

3. Work Allocation

As explained above, the team was divided into subgroups that provided tasks to be completed simultaneously. Here is a list of the things that the group members worked on. It must be noted that all of the work is reviewed by everyone and changes made by the reviewers are not listed below. Lastly, the parts in quote are directly taken from the report subsection titles.

- After the formation of groups, meetings about the content of the game were made.
- It was decided that the game would have a theme of coronavirus.
- When it was time to write analysis report the work was divided as:
  - Doğancan worked on "Introduction", "Overview", "Use Case Model"
  - Ertuğrul worked on "Functional Requirements", "Activity Diagram" "State Diagrams"
  - Alperen worked on "Non-functional Requirements", "Object-Class Model", "Sequence Diagrams"
  - Oğuzhan worked on "Sequence Diagrams", "Object-Class Model", "Screen Mockups"
  - Osman worked on "Sequence Diagrams", "Object-Class Model", "Screen Mockups", "Conclusion"
- When it was time to write design report the work was divided as:
  - Doğancan worked on "Introduction", "High Level Software Architecture"
  - Ertuğrul worked on "High Level Software Architecture"
  - Osman, Oğuzhan and Alperen worked on "Low-Level Design"
- The presentation was prepared by everyone working on their specific parts as before simultaneously using Google Slides.
- After taking feedback for the reports, each individual changed their respective sections according to the given feedback from the TA's.
- The implementation was divided into classes. However, the work done was done through constant communication through Discord.
- The final report was written likewise through common work on Discord.

4. User-Manual

Welcome to Coronapoly!

This game is a bit different than your classic Monopoly experience. Players up to 5 race each other to become the monopoly in the Covid-19 Pandemic environment! In order to win, the players must also pay attention to their infection levels and avoid being infected to not fall behind!

The game could be played as single player against bots or as multiplayer against real players. Each player starts with 1500TL and players must roll dice and must choose for an action to end their turn. Their action could either be buying the property which they landed on, paying rent for that property, managing their property or skipping their turn. The goal of the game just like the classic monopoly, is to be the last standing player in the game economy.

In the main menu, the single player option starts the game with other non-human players. If a user wants to play multiplayer, then the game will lead the user to another screen where all of the player information must be filled. After each of the players check `Ready` the game starts!

The game board consists of neighborhood properties, transportation properties, tax slots, quarantine and card slots. The first player starts with rolling the dice and then choosing what to do on the game bar at the bottom of the screen. On the right side of the screen, player information could be seen with their money, properties and health conditions.

Users could mortgage their property in return of some instant cash. However, mortgaged players can not gain rent income from their properties. Players could also trade their properties by pressing the trade button. A new screen will pop-up and will require the users to fill in the spots such as the proposed money and the desired property from a specific player. After the information is filled, the players will need to type in their passwords to complete the transaction.

Users could buy houses on their properties and hospitals if they have the required number of houses on their properties. If the players are eligible for building, then the rent price of the properties increase that creates a solid advantage against other players.

When there is only one player left who has not gone bankrupt yet, the remaining player wins the game!

5. Build Instructions

In order to run our project, an advanced IDE such as Intellij is strongly recommended.

Here are the steps to run Coronapoly:

- Open Intellij

- Create new project using `Get from VCS`

- Clone Coronapoly to the IDE

- After cloning go to https://gluonhq.com/products/javafx/

- Download the latest version of JavaFx

- After downloading, open the cloned project

- Go to File > Project Structure

- Select Libraries and add new project library

- Choose the lib file of the downloaded and extracted javafx library

- Then go to the Project files section

- Choose srs > Controller > GameUI

- Build the project

- Run the project using the run button above or by scrolling down to the main function and using the run button next to the function

- If any error comes up, try rebuilding the project