Bilkent University

Department of Computer Engineering

CS 319 Project

Project short-name: Settlers of Anatolia

Final Report

Group members: Enes Merdane, Irmak Demir, Mehmet Alper Genç, İrem Kırmacı, Göksu Turan

Instructor: Uğur Doğrusöz
Teaching Assistant(s): Hasan Balcı

Final Report

Decemberr 22, 2019

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Object Oriented Software Engineering course CS319.

Introduction

In the current status, the main functional requirements of Settlers of Anatolia which are playing the game, setting preferences for the game, learning how to play are already implemented. Besides, our new mode Crazy Mode can be played both as a local multiplayer game and with bots. As suggested in our reports, bot players exist in our current status of implementation. Even if our bot players do not make the most logical decisions in every situation they can play like a normal player. In the settings section, player can change sound settings and save this change.

Design Changes

2.1 Changes in Presentation Layer

Presentation Layer is implemented by adhering to our low level design which is mentioned in the Design Stage. However, some minor changes are required during implementation. TradeRequestText class is added to manage text views in trade window. We changed the name of SettingsPreferences to SettingsController. ScreenViewer class is removed and instead all methods of ScreenViewer is implemented in MainApp.To initiate game we need a class extending Application.We already had this class, MainApp.Therefore, all screen transition is implemented in that class.GameOptionsScreen is added addition to design state.

2.2 Changes in Business Layer

In the Business Layer, we implemented the layer by adhering to our low level design which is mentioned in the Design stage. However, some minor changes which are removing, adding and improving classes were occurred during the implementation stage. Abstract Trade class is removed due to efficiency reasons. We changed the name of GameObject class to LocatableObject. Since we thought that it is more efficient, rendering of locatable objects is removed from LocatableObjects and implemented in the Presentation Layer. Similar to this change, Thief class is removed and related methods of Thief class is implemented in the Hexagon class.

BotPlayer class is implemented as we indicate at the Design stage. However, due to implementation of BotPlayer addition of some new methods and classes is required. Also, currently existing methods and classes are modified to integrate bot into our project. In methods of GameController, we check whether the player is a bot or not to disabled user ability of play. In the PlayerList class, three of the players are turned to bot player. Also, to show both multiplayer and bot mode, we changed the constructor of GameController.

For the trade options, the Trade abstract class was removed, and now DomesticTrade and TradeWithBank classes work differently. These two classes hold the trade requests of a player, either with another player or with a bot, respectively.

As we indicate in the Design Stage, classes are related to CrazyMode are implemented by using Abstract Factory Design pattern and the implementation is completed as consistent with the design report.

2.3 Changes in Data Layer

Since we focused on the main features of the game, we did not have enough time to implement this layer.Therefore, currently our continue game option does not work.

Lessons Learnt

During this project, we have learned the benefits of analyzing and designing a software project before coding it. We have found out that doing so results in a more structured implementation, a more organized and errorless code, and overall a better piece of software. We have also learned the benefits of working together. Dividing the code subsystems and assigning people to these subsystems, allowed us to implement more code in a smaller amount of time.

User Guide

Main Menu

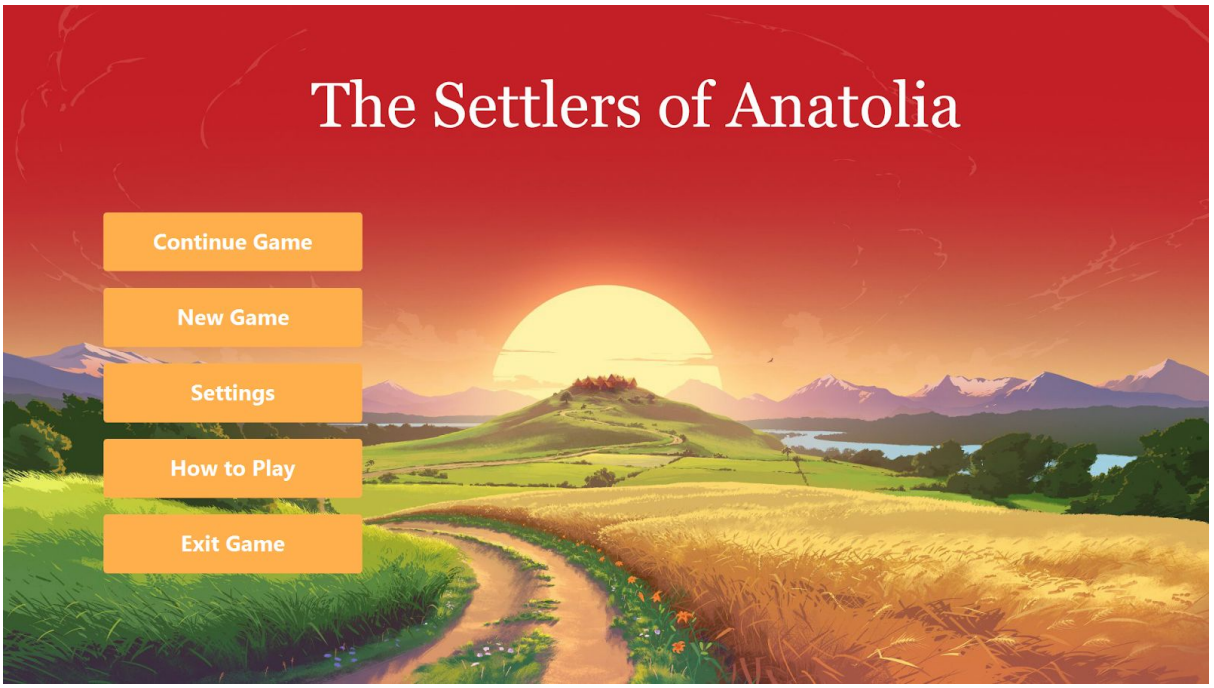

Figure 1 : Main Menu Screen [1]

This is the Main Menu that appears when one runs the game. There are 5 options: Continue Game, New Game, Settings, How to Play and Exit Game. New Game takes the user to the Game Options screen so that they can select the properties of their game. Settings opens the Settings screen for the player to adjust the settings of the game to their preferences. How to Play opens the How to Play screen that contains useful information about the game. Finally, exit game terminates the app.

Game Options Screen

Figure 2 : Game Options Screen [1]

This is the Game Options screen that allows the user to select their username, color, and the type of the game. There are 2 types: Play with Friends, and Play with Bots. If the user chooses to play with their friends, the game will be played with 4 real players. Otherwise, the game will be played with bots.

Game Screen

Figure 3 : Game Screen [2][3][4][5][6]

This is the screen in which the game is played. In the middle, it contains the map. The circles on the corners of the hexagons represent vertices, on which the players can build settlements, or cities. The edges of the hexagons represent edges, on which the players can build roads. The inside of the hexagons represent what resource they produce, the number that the dice must roll in order to produce resources from them, and whether or not that hexagon has the robber inside it. If a hexagon has a robber inside, it can not produce resources until the robber is removed. Around the map, there are 4 player icons, that also show the color of that player, as well as their scores (on the bottom left of the icon), and the number of knight cards they have used (around the icon, signified with a knight picture). Under the map, is the number of resources that the current player has. These numbers are marked X if the current player is a bot.

To the left of the map, there is a roll dice button that allows the players to roll the dice. The sum of this dice will either be used to move the robber and steal resources, or produce resources. To the right of the map, is the development cards frame, the resource costs, and the user options. The development frame is used to play the development cards that the user has. Next to the name of each development card, are two numbers, one in parentheses and the other not in parentheses. Those in parentheses represent the total number of that card the player has, while those not in parentheses represent the number of cards that the player can use. The resource costs represent the number of resources the player must spend in order to build a settlement, city, or a road, or to buy a development card. Finally, with the user options, the user can decide what to do. If the user selects “Build settlement”, when they click on a vertex, a settlement will be built on the vertex if they have enough resources, and can build a settlement on that vertex. A similar case follows for building cities and roads, except the user must click on an edge instead of a vertex when

building a road. Trade with Bank and Trade With Players options open a menu with which the user can trade with the bank, or trade with other players. Finally, they can use the “Buy Development Card” button to buy a development card if they have enough resources.

Figure 4 : Game Screen [2][3][4][5][6]

This is an example of how the Game Screen looks like during gameplay. Above the Roll Dice button is the result of the dice roll. On the map, the vertices that have settlements and the edges that have roads are colored with the color of the player that has a settlement or road on them. If the vertex contains a city instead of a settlement, there is also a gold stroke around the vertex. In the user options menu, the current choice of building is denoted with a red rectangle behind it.

Figure 5 : Game Over Screen [2][3][4][5][6][7]

This is the screen that pops up when a player wins the game by reaching 10 victory points.The white filter behind the “Game Over” menu forbids the player from clicking anything on the Game Screen. The only action the user can perform at this point is to go back to the Main Menu.

Trade Screens

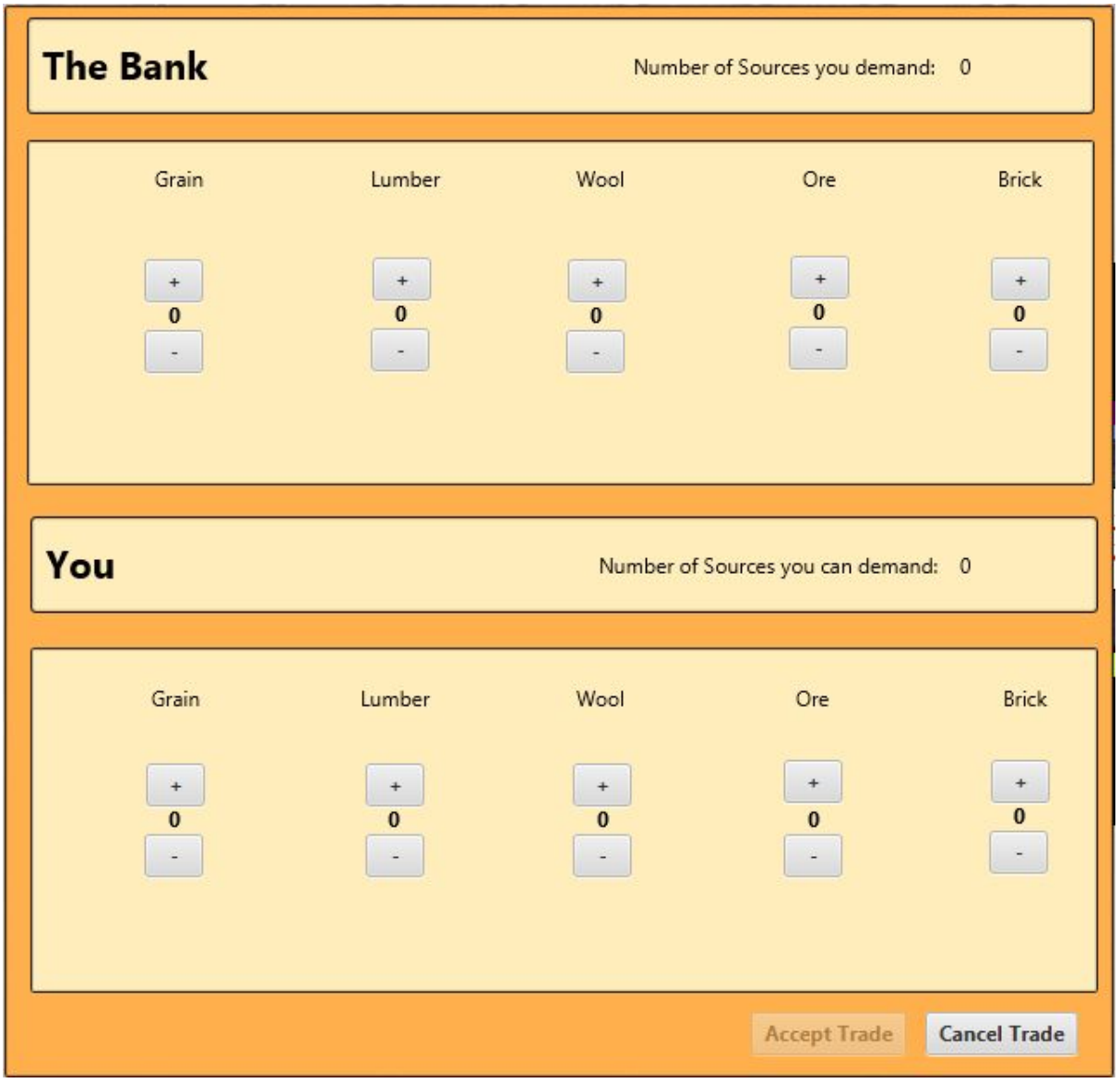

Figure 6 : Trade with Bank Screen

This is the Trade with Bank screen. Using this screen, the player can trade resources with the bank. The player must give 4 of a resource to get 1 of a resource from the bank. If the trade is invalid, meaning it does not satisfy the condition, “Accept Trade” button will not be enabled. The Cancel Trade button closes the Trade with Bank screen.

Figure 7 : Trade with Player Screen

This is the Trade with Players screen. On the left is a list that the player can see all trade requests made by other players. Clicking on one such request will display the information of that request on the top right frame. While a trade request is being displayed, the “Accept Trade” button on that request will allow the current player to finish the trade. This button will be disabled if the user does not have enough resources to complete this trade. On the frame at the bottom, the user can create a trade request to send to other players. The - and + buttons are used to manipulate the values. When the user decides on the trade, they can click the “Create Trade Request” button to finish the trade. Finally, the “Close Screen” button disables the Trade with Players screen.

How to Play Screen

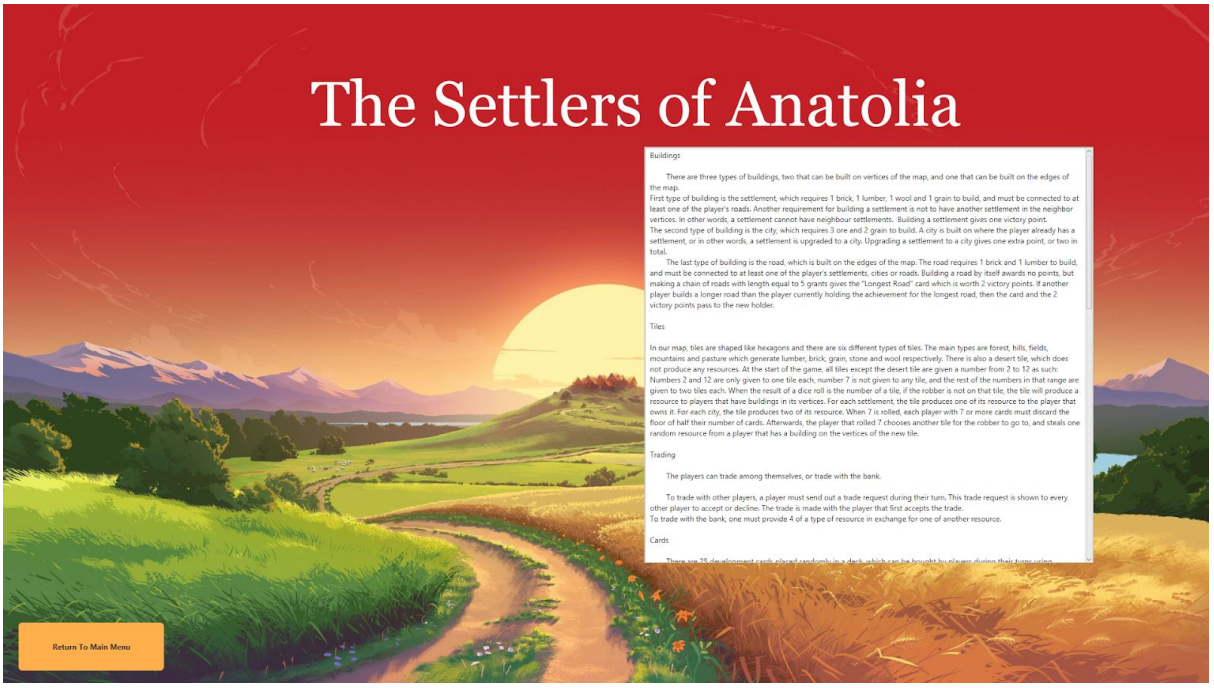

Figure 8 : How To Play Screen [1]

In the How to Play screen, the user can read useful information about how to play the game.

Build Instructions

System Requirements

The Settlers of Anatolia game is able to operate in every computer whose resolution can be changed to 1920 x 1080. To be able to play the game in this specified computers, JDK 8 version of the Java Development Kit must be installed, since newer versions do not contain JavaFX..

User's Install Guide

In order to play the game, as mentioned earlier JDK 8 version of Java must be installed and can be installed from https://java.com/en/download/ User should download the game file named *SettlersOfAnatolia-1.0.jar* by entering https://github.com/Enesmerdane/CS319-1B-CA. After, downloading the jar file, by writing the following command to the command line:

```
java -jar SettlersOfAnatolia-1.0.jar
```

References

[1] https://wallhere.com/en/wallpaper/1591901
[2] https://wallpapercave.com/wood-wallpaper-1080p
[3] http://www.baytekent.com/about/facebook-default-no-profile-pic1/
[4] https://www.artstation.com/artwork/Z60a8
[5] https://www.artstation.com/artwork/ybo4O
[6] http://historywallcharts.eu/view/a-medieval-road