CS 319 - Object-Oriented Software

Engineering Design Report

Backgammon Game

Group 1-D

Doğukan Altay

Armağan Sarı

Ömer Sakarya

Berkalp Yılmaz

1.Introduction

Backgammon Game is a classic and histroical board game that exist for over 5.000 years. The game itself is played with two players. Each player has its own checker which has 15 identical checkers sets and placed fixed at the beginning of the game. The player's aim is that to move all of their checkers to their area and collect them to the their stacks before opponent does first. In the game players can choose how many rounds they are going to play. The winner decided after the best of the corresponding rounds.

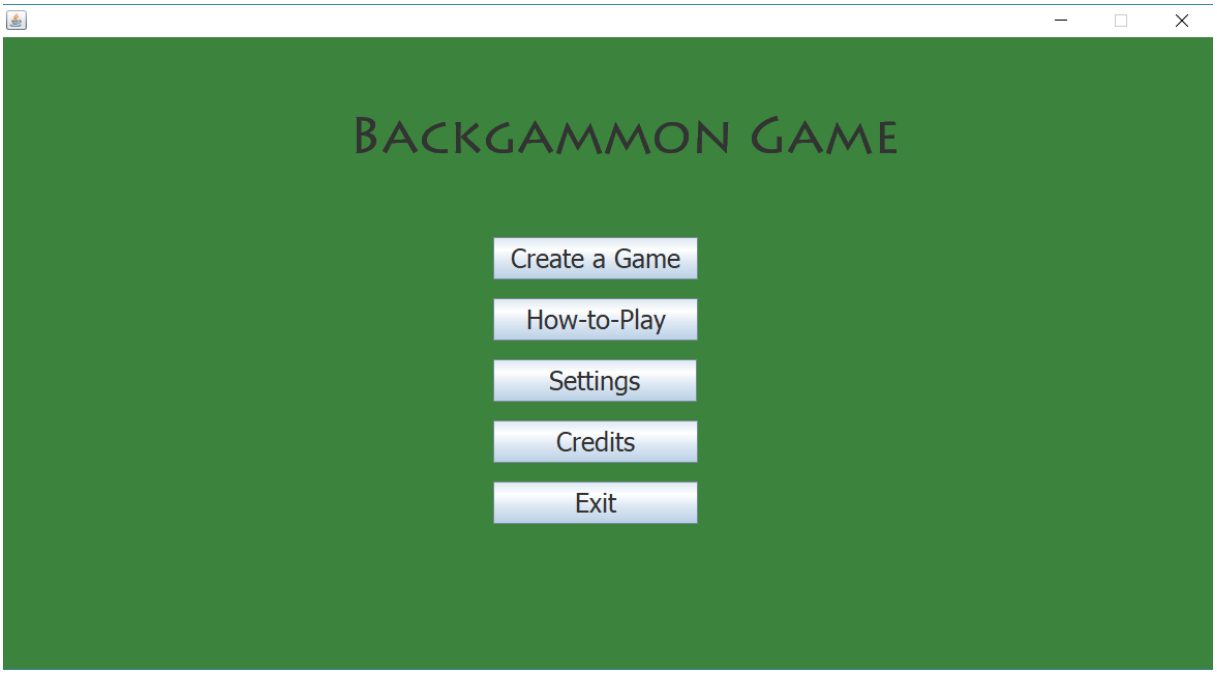
The Backgammon game is a highly strategy and luck dependent game. In order to win the game players need to foresee their opponents strategies and take measures while having fun!

2. Application Setup

To run the Backgammon Game users must have java installed in their computers. After java installation, users does not need any other additional setup procedures to follow. They only need to run the jar file of the Backgammon Game. Since our application is open source, developers who wanted to improve the game can contribute to the project through the github page of the project. Our code has been developed in Netbeans IDE. So future developers highly recommended to use the same IDE because project has been optimized for coding in Netbeans. The github page for the project is mentioned below.

- https://github.com/DogukanAltay/Backgammon

3. User’s Manual

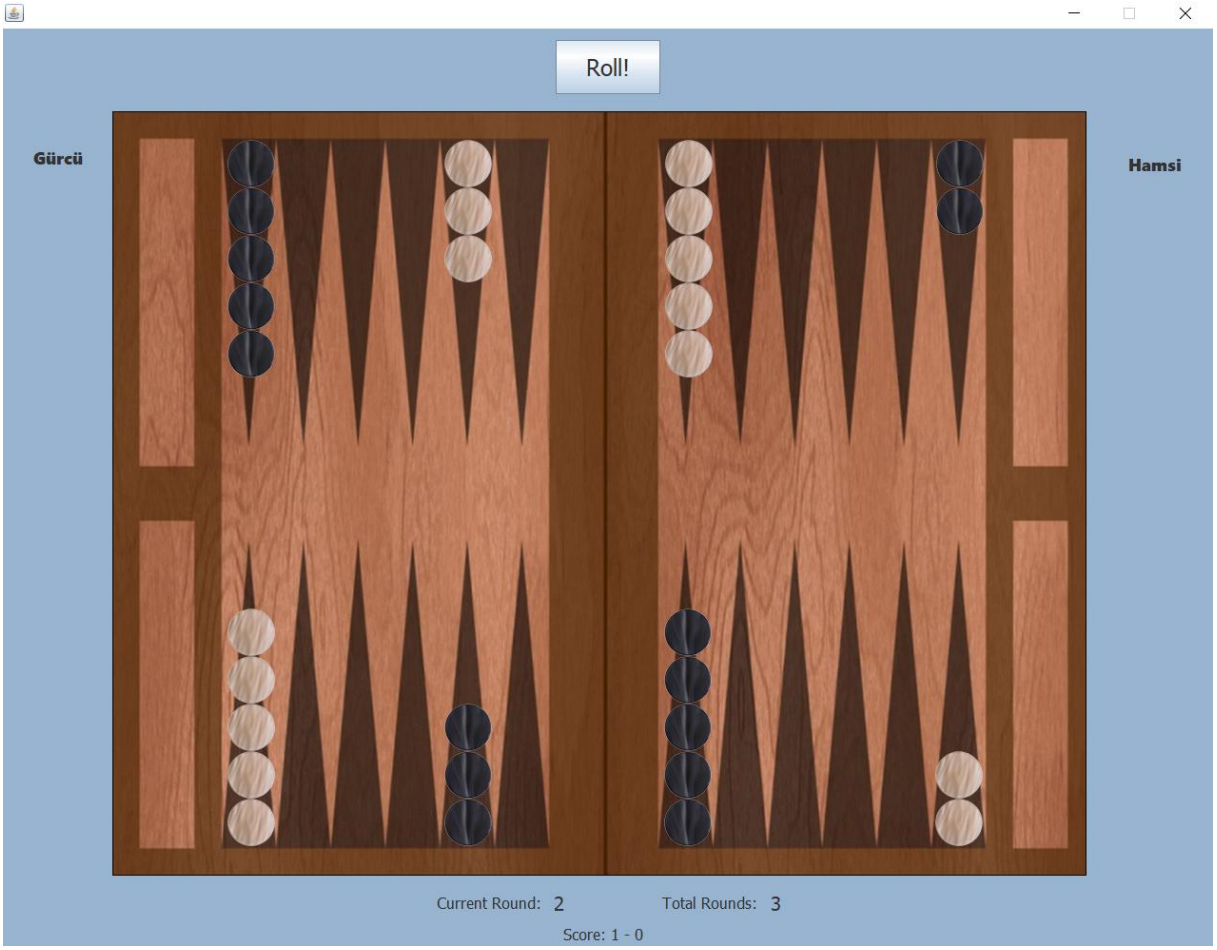


Main Menu:

This page of the application consist of main functions of the game. Users can choose to create a game by pressing the “ Create A Game” button. This button leads the users to the Set Game Menu where they can choose the setup of the game.

Game Setup Menu:

Users can decide the attributes of the game. They can specify their names, can choose the color of their checkers and decide how many rounds they will play by using the corresponding buttons on the menu. After clicking the “ Play” button the system initializes the game with the corresponding attributes.

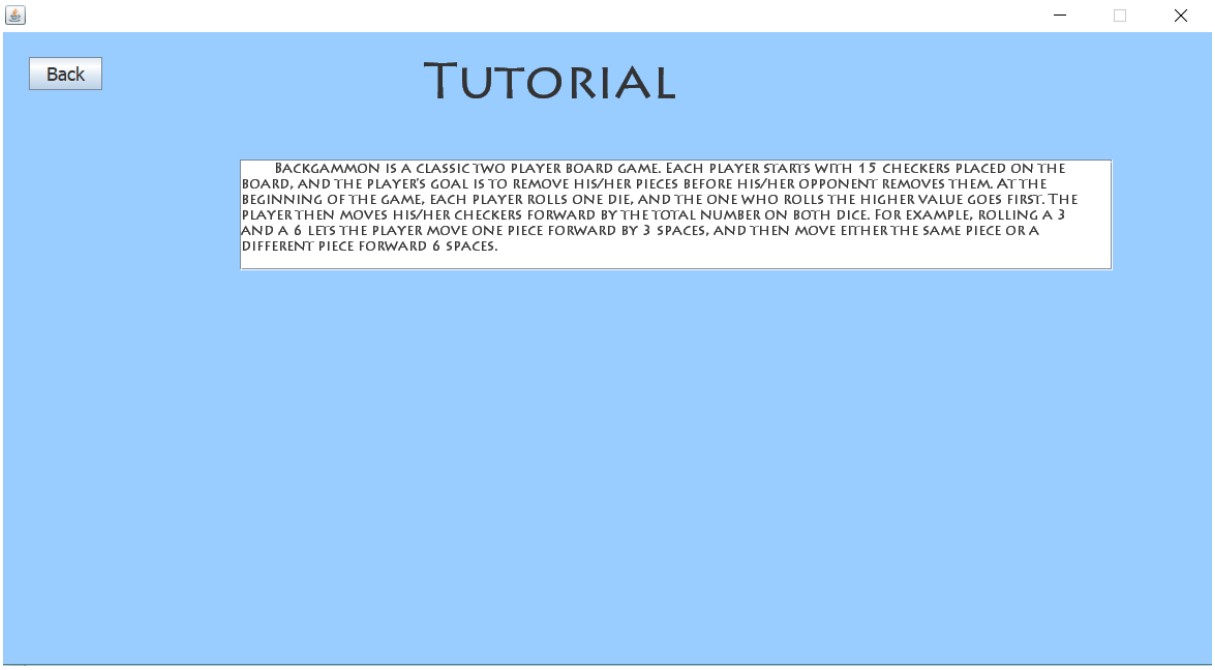


Game Screen:

In this part of the project users have the game board, their name tags each side and a roll dice button on top of the window. They can roll the dice by simply just clicking “ Roll!” button. At the bottom of the window players can see how many round they will play. The “Back to Main Menu” button leads the users to main menu by terminating the active game.

Settings:

User can choose the sound appereances in this menu. They can choose to mute or unmute the sound in this menu and can go back to main menu whenever they want by pressing the “Back” button.



Tutorial:

In this menu, users can learn how to play backgammon game by reading the game's rules and aim of the game. Users can go back to main menu by using the "Back" button in the tutorial page.

Credits:

Users can see the information about the developers of the game. Users can go back to main menu by using the “Back” button.

3. Changes Done During The Implementation Stage:

As we progress through the project, we needed to make considerable changes in the design of the project in order to make implementation easier and doable. In the design stage we thought that a ObjectController class can be used in order to initialize game objects in the GameController class. Since our GamePanel extends the JPanel using a ObjectController class made things harder and inconsistent in the game so we decide to exclude from the design. After this change, GamePanel class has gained the control of the whole game objects in a single hand and made things easier to implement. As a result our GamePanel which extends JPanel has the all game objects which all extends Jlabel inside a container which is JLayeredPane. For our InputManager class, in the desing stage we thought that a single manager class can handle all the user inputs in the all stages of the code. As we progress, we saw that this desing is not possible to implement we decided to go individual input managers in the code which specific to each GUI dependent class.

In the design stage we excluded the ScreenManager class, since our program did not need any specific graphic manager to render the graphical objects. By this decision we simplified the implementation of the code and relieved some burden from the system.

In the GameObjects we thought that Board object is an initializable object in the GameController. Since the board visualy does not have any function except the image of it we decided to remove that class and add as a background object in the GamePanel.

In the Settings menu, at first we thought that players can choose their checker colors in the settings menu. However, this an attribute that players should decide while setting the game, we moved that features to the Set Game Menu. All in all we achived a better, logical implementation and easier implementation.

In the Rule implementation of the project we decided three specific rule classes which extends an abstract rule class. During implementation we decided to add a new rule class in order to make rules of game consistent and a good abstraction of rules.

4. Incomplete Parts

As we planned in our analysis of the project, we decided to put sounds in the game to make game more fun. However, implementation of sound created a huge unexpected work load so we postponed the implementation of sounds in the game for future updates.