COMP132: Advanced Programming
Programming Project Report

UNO Card Game

Tahir Hayta 83412

Spring/2024

Part 1

General Demo Information:

Users Information:

- Nickname: fatma, Password: f

Total wins: 15, Total losses: 2, Total games: 17, Total score: 2300, Win/Lose Ratio: 7,50, Average Score: 135,29

Existing game: fatma TTT 4

- Nickname: Haşmetli tahir, Password: t

Total wins: 5, Total losses: 2, Total games: 7, Total score: 400, Win/Lose Ratio: 2,50, Average Score: 57,14

Existing game: Haşmetli tahir TTT 7

- Nickname: tt, Password: tt

Total wins: 11, Total losses: 28, Total games: 39, Total score: 793, Win/Lose Ratio: 0,39, Average Score: 20,33

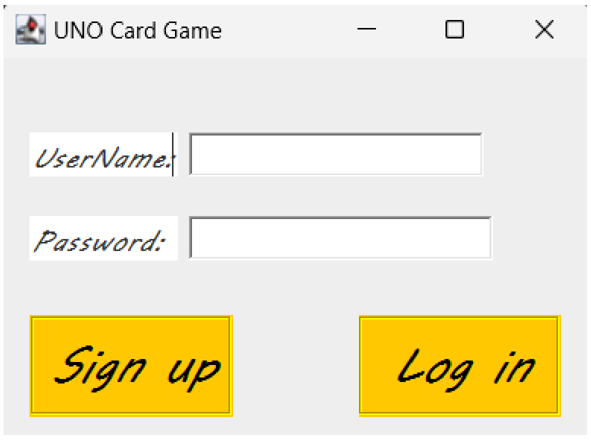
Existing Game: tt TTT 3

- Nickname: Ege Garibansoy, Password: e

Total wins: 1, Total losses: 999, Total games: 1000, Total score: 10, Win/Lose Ratio: 0,00, Average Score: 0,01

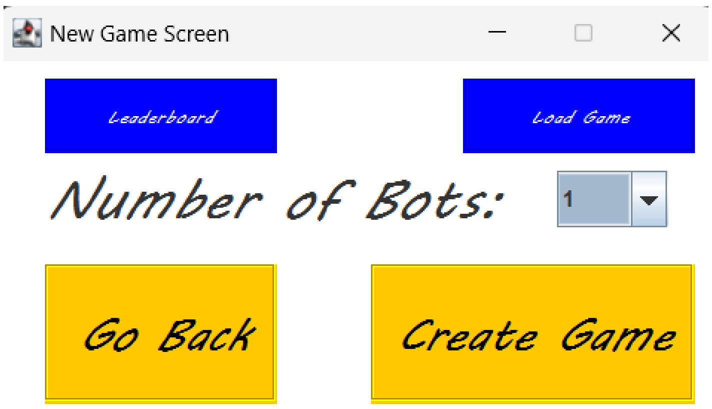
Existing Games: Ege Garibansoy TTT 1, Ege Garibansoy TTT 2

Application usage information:



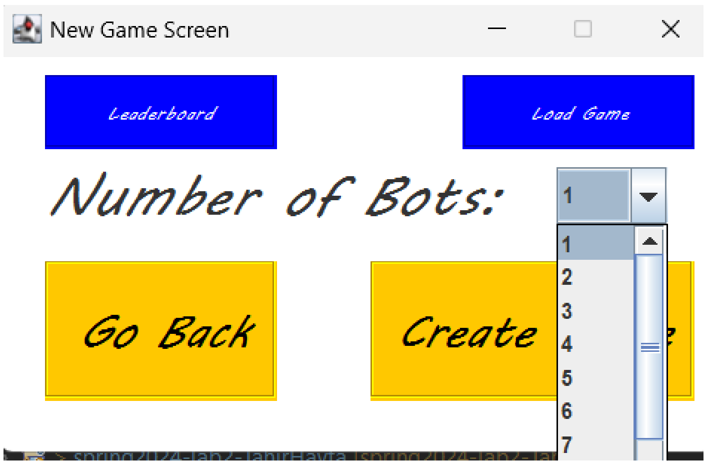
When the application starts, this screen appears. You can create a new user by entering a new username and password. After that, you can click sign up. So, you created a user, ready to play.

If you have a user already who exists, you can enter your username and password. Then you can click login.



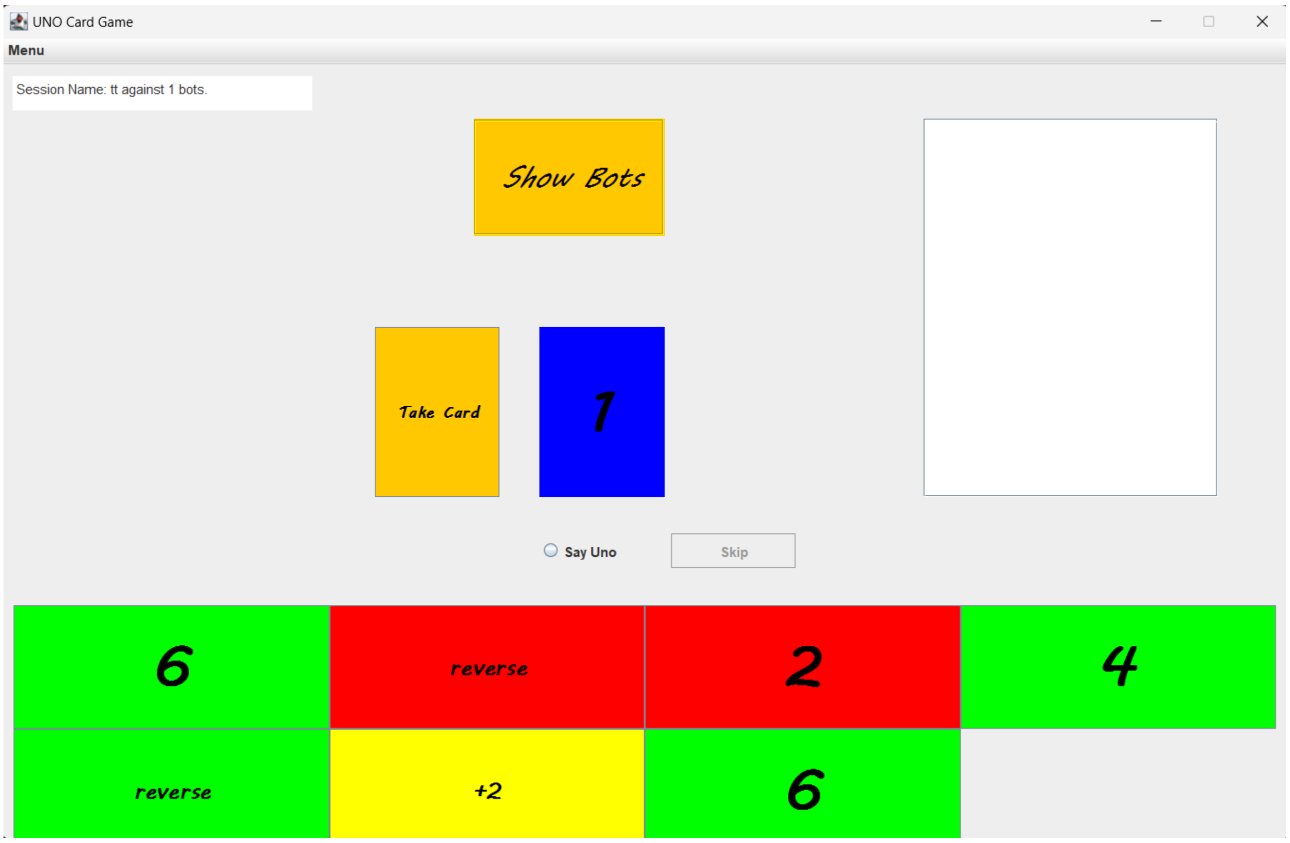
After sign up/login page, this screen appears. You can click Go Back Button if you want to go back to previous screen.

On this page, you can either load a previously saved game, start a new game, or see leaderboard.



To start a new game, you must at first choose how many bots you want to play with. You can do it by clicking combobox which you can see in the picture.

Then you can click to Create Game. After clicking it, game starts.

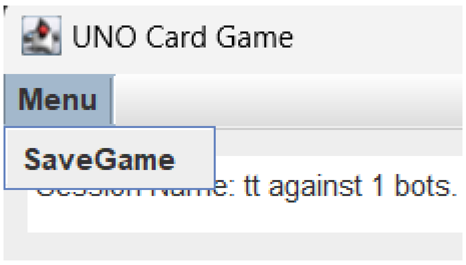


When game starts, this screen appears. 7 cards at the down(green 6, red reverse, red 2, green 4, green reverse, yellow +2, green 6) are the cards at your hand. You can simply play any one of them by clicking on them.

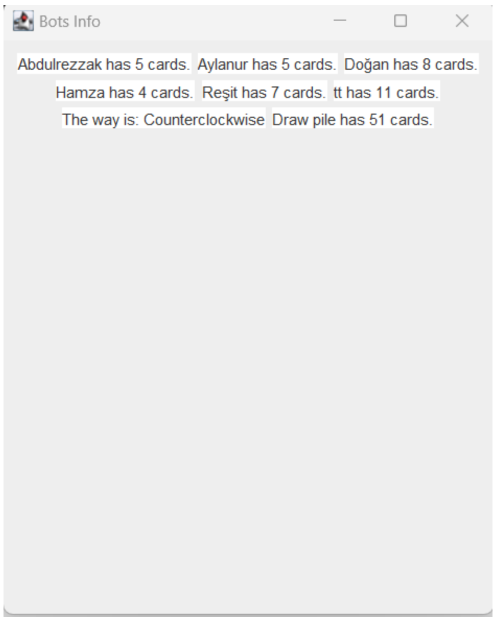
Card in the middle (blue 1) is the top card on the discard pile. If you want to play a card, it must match with this card. Otherwise, you cannot play that card. As you can see in the picture, any of our cards do not match with blue 1. So you must take a card by clicking Take Card button near blue 1. When you click it, it will give you a new card, which you will see down, near your cards.

After taking a new card, if it matches with blue 1, you can play it by clicking it. If it does not match, or if you do not want to play any card, you can either take a new card, or can skip your turn. In order to skip, you must click on the Skip button near Say Uno button. It is unclickable at this screenshot, but it will be clickable after you take a new card.

You can say Uno by clicking on the Say Uno button. If you do not click it, when you have 1 card, you will be automatically punished. It means two cards will be added to your deck.



If you click on menu button, you will see SaveGame button. Anytime in the game, you can save your game by clicking it. Your saved game's name will be "username+TTT+numberofbots". For instance, if you saved a game as user tt when you were playing against 3 bots, your saved game's name will be "tt TTT 3".

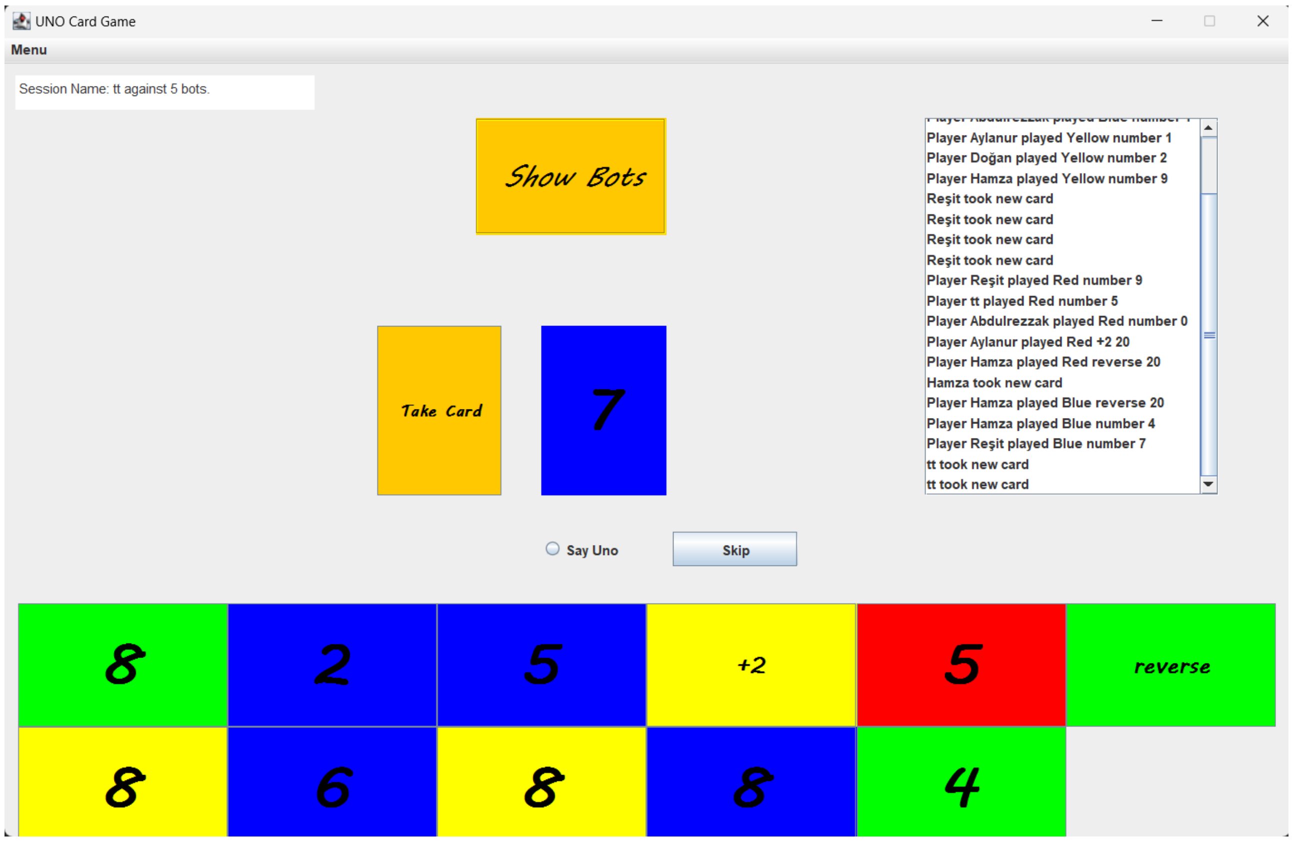


Bots Info

Abdulrezzak has 5 cards. Aylanur has 5 cards. Doğan has 8 cards.
Hamza has 4 cards. Reşit has 7 cards. tt has 11 cards.
The way is: Counterclockwise Draw pile has 51 cards.

If you click on the Show Bots button, this screen will appear. In this screen, you can see which player has how many cards. You can also see the way of the game.

After you play by playing a card or taking a new card, bots will play very fast. You will see the last card played by the last bot on the discard pile. If you want to learn who did what, you can look at screenshot:



UNO Card Game

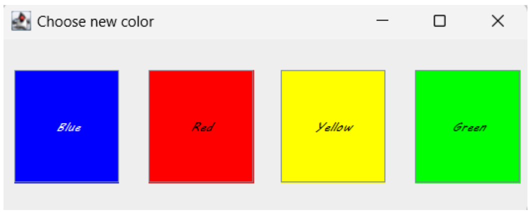
Menu

Session Name: tt against 5 bots.

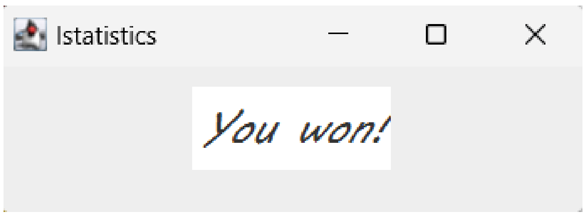
```
Player Aylanur played Yellow number 1
Player Doğan played Yellow number 2
Player Hamza played Yellow number 9
Reşit took new card
Reşit took new card
Reşit took new card
Reşit took new card
Player Reşit played Red number 9
Player tt played Red number 5
Player Abdulrezzak played Red number 0
Player Aylanur played Red +2 20
Player Hamza played Red reverse 20
Hamza took new card
Player Hamza played Blue reverse 20
Player Hamza played Blue number 4
Player Reşit played Blue number 7
tt took new card
tt took new card
```

This is a screenshot from middle of a game. You can see who did what on the right of the screen. In there, you can see every move every player did in this game.

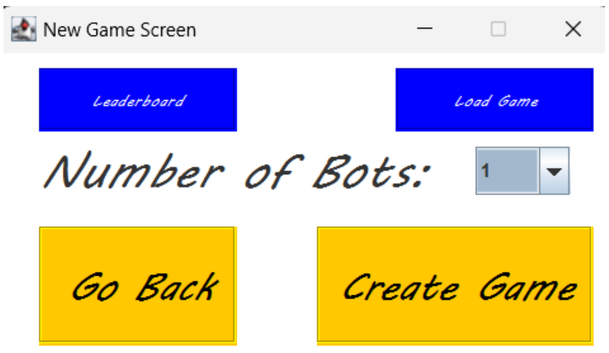
Also in this screenshot, you can see that mainplayer has 11 cards. As you can see, when player takes new cards, size of the cards in the deck becomes smaller.



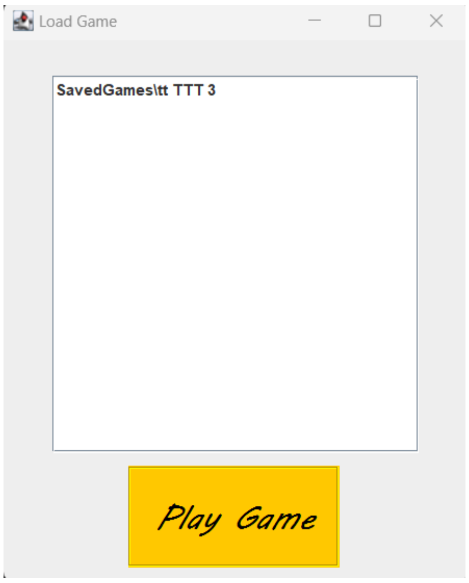
If you ever play a wild card or +4, this screen appears. You can choose any color that you want. Then, game will continue.



After a game is finished, this screen appears to show who has won. Also, new game screen appears, so you can start to a new game.

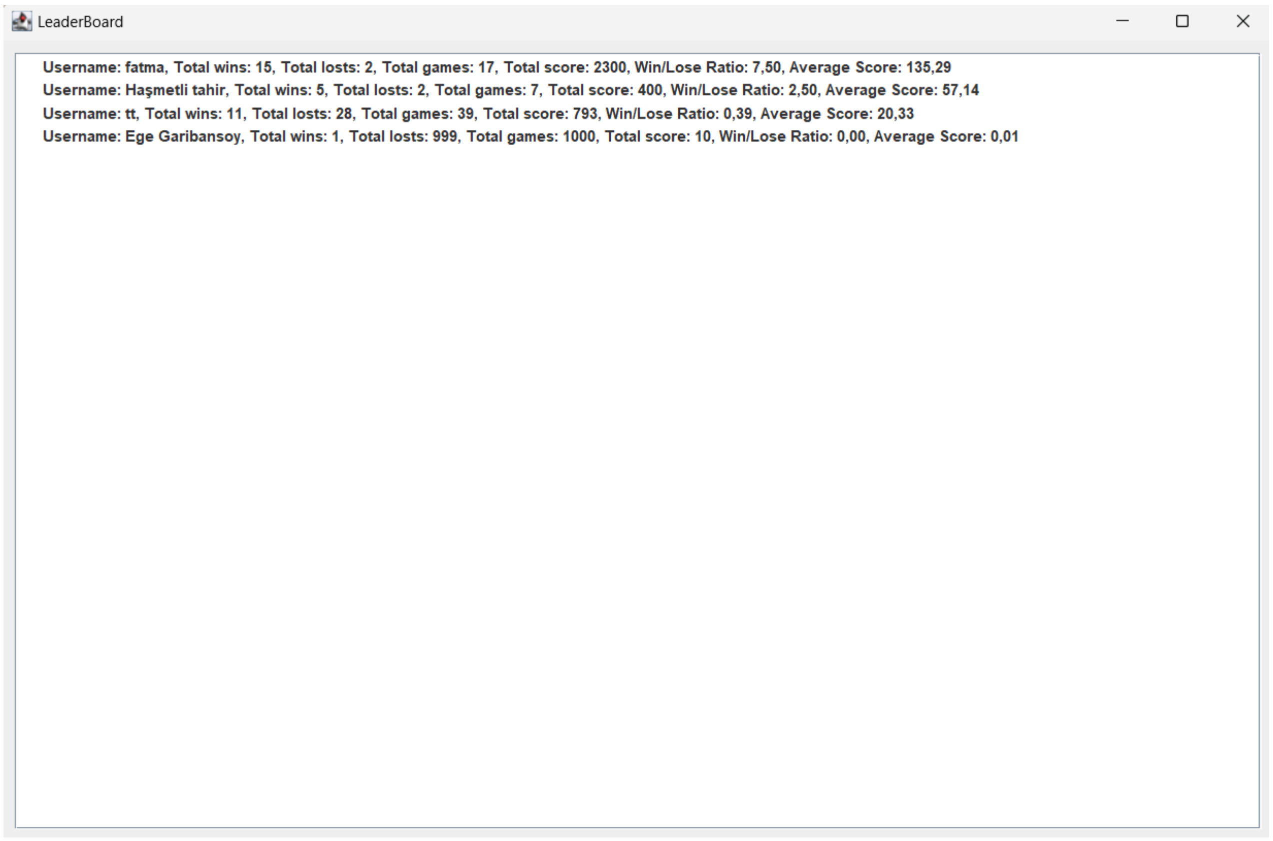


In New Game Screen, you can choose load game if you want to load a game that you have saved.



```
SavedGames\tt TTT 3
```

When you click it, this screen appears. You can choose a game that you saved by clicking it. Then if you click Play Game, you can continue playing that game in Play Screen. You can only see saved games that your user saved.

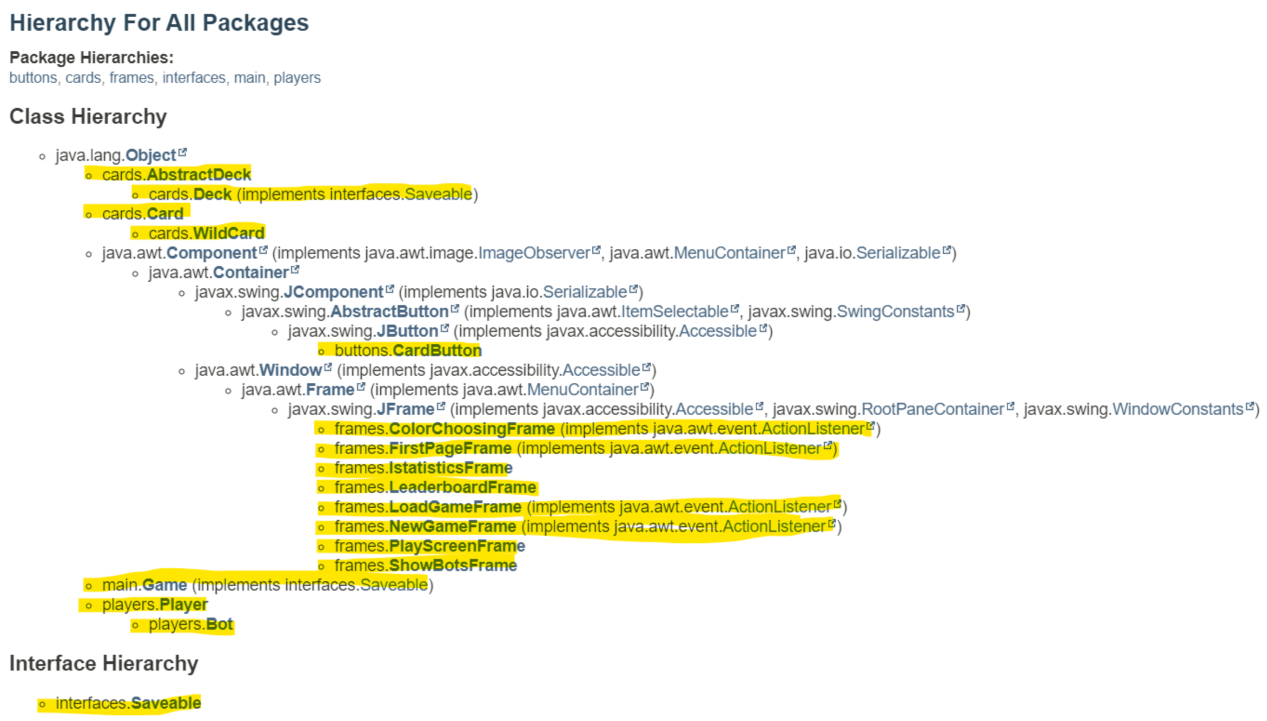


```
Username: fatma, Total wins: 15, Total losts: 2, Total games: 17, Total score: 2300, Win/Lose Ratio: 7,50, Average Score: 135,29
Username: Haşmetli tahir, Total wins: 5, Total losts: 2, Total games: 7, Total score: 400, Win/Lose Ratio: 2,50, Average Score: 57,14
Username: tt, Total wins: 11, Total losts: 28, Total games: 39, Total score: 793, Win/Lose Ratio: 0,39, Average Score: 20,33
Username: Ege Garibansoy, Total wins: 1, Total losts: 999, Total games: 1000, Total score: 10, Win/Lose Ratio: 0,00, Average Score: 0,01
```

There is also a leaderboard for existing users. You can open this screen from New Game Screen by clicking Leaderboard button in the upper left. In here, users are sorted according to their average score. You can see their information.

Part 2

Project Design Description:



Hierarchy For All Packages

Package Hierarchies:
buttons, cards, frames, interfaces, main, players

Class Hierarchy

- java.lang.Object
  - cards.AbstractDeck
    - cards.Deck (implements interfaces.Saveable)
  - cards.Card
    - cards.WildCard
  - java.awt.Component (implements java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable)
    - java.awt.Container
      - javax.swing.JComponent (implements java.io.Serializable)
        - javax.swing.AbstractButton (implements java.awt.ItemSelectable, javax.swing.SwingConstants)
          - javax.swing.JButton (implements javax.accessibility.Accessible)
            - buttons.CardButton
      - java.awt.Window (implements javax.accessibility.Accessible)
        - java.awt.Frame (implements java.awt.MenuContainer)
          - javax.swing.JFrame (implements javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.WindowConstants)
            - frames.ColorChoosingFrame (implements java.awt.event.ActionListener)
            - frames.FirstPageFrame (implements java.awt.event.ActionListener)
            - frames.IstatisticsFrame
            - frames.LeaderboardFrame
            - frames.LoadGameFrame (implements java.awt.event.ActionListener)
            - frames.NewGameFrame (implements java.awt.event.ActionListener)
            - frames.PlayScreenFrame
            - frames.ShowBotsFrame
  - main.Game (implements interfaces.Saveable)
  - players.Player
    - players.Bot

Interface Hierarchy

- interfaces.Saveable

Class relations:

In this picture, you can see all class hierarchy. Yellow underlined ones are all classes that I created. AbstractDeck is superclass of Deck. Card is superclass of WildCard. Player is superclass of Bot. Game and Deck classes implement Saveable.

Number cards and action cards are Card because they are almost the same. Wildcards are WildCard, because they have the attribute New Color.

There are Cards in Decks. Every player has a Deck. Game has players, and in addition to these players, Game has two Decks. These two Decks are discard pile and draw pile.
I named draw pile as deckToTakeCards, and discard pile as deckToGiveCards in codes. It is important to state that both Player and Game classes have deckToTakeCards and deckToGiveCards, which are references to the same Deck objects. In other words, if player takes Cards from deckToTakeCards Deck, both in Game and Player's deckToTakeCards Deck will change.

Deck is probably the most important part of the program. There are several methods in the Decks, to take a card from a Deck and give a card to a Deck. Player class's methods to take or play Cards call the methods from the Deck.

Inheritances, type hierarchies, interfaces, abstract classes:

Deck inherits methods from AbstractDeck. WildCard inherits methods from Card. Bot inherits methods from Player. One can see inheritances visually from the picture above.

There is one interface which is called Saveable. Game and Deck classes implement it. It has a method called Save. Game and Deck classes can be saved to a .txt file thanks to that method.

In my code, there is only one abstract class, which is AbstractDeck. It has very little importance for code. I only created it so that I could plan what methods to add to Deck class. Other than that, my code has not needed any abstract classes. One can say that Card class could be implemented as Abstract. I also thought about it. However, it only makes code longer and complicated, because my Card class has objects. It has objects which are number card and action card. I did not implement additional classes for them, as they are almost same. In conclusion, my code did not really need abstract classes. Therefore, I only made one abstract class.

GUI components:

I used several GUI components. I made 8 JFrames. I added JPanels called content pane to most of them. Mostly, my Layout was null, so I could place other components wherever I wanted. I used JTextPanes to write somethings on the screen. In first page, I used TextFields to take username and password. I used JComboBox for choosing how many bots one wants to play with, as bot number is in the range of 1 to 10.

I used JScrollPane in order to show every event in a game in PlayScreenFrame.

I used buttons to show the main player's cards, so he can click them if wants to play them. I enabled them every time when it is user's turn, and disabled them after that. However, I made the card on the top of the discard pile JLabel. Because it is not clickable, it only shows the color and the name.

.txt file processing details:

There are 3 significant .txt files and one SavedGames folder, which I hide .txt files which are saved games. AllGameIstatistics.txt includes all the game information which has been finished. You can see every movement in every game that players do in there. In other words, this file is where I keep every event or action every player (bots included) within the game makes for review.

In leaderboard.txt, I have game user statistics. In each line, it has username, games won, games lost, total games, and total score.

In users.txt, I have usernames and passwords.

I used buffered writer to write on these files, as BufferedWriter can write to the end of files. I also used BufferedReader.

Game session loop implementation:

My gameloop is in the Game class. It starts in the beginning of the game. However, it stops when it is main player's turn and saves the turn number to Game.turn. After main player plays card or skips his turn, it restarts. When it stops, It contunies from where it stopped by looking at Game.turn. This progress repeats until someone wins.

I used a while loop to make game loop. It checks whether someone has won before the start of the game. I initialized a variable "i" to check whose turn it is. It increases every turn. When it becomes bigger than player.size, loop makes it 0.

We cannot see bots' play, because they immediately play and turn comes to main player very quickly. However, we can see what each bot does from the JScrollPane in PlayScreenFrame.

Bot implementation:

Bots find possible matching cards by looking at the discard pile Deck. Adds these cards to a list. Then they look at them in a for loop, if there is a number card. They play that number card. If there not exist a number card, they can play either an action card or a wild card.

If they do not have any card which matches the card in discard pile, they took a new card. They repeat it until there comes a playable card.
I named bots with TA and Professor names from Comp132. So technically, I played Uno with TAs. Did I get bonus points :)

Note: My main function and pledge of honor is on the FirstPageFrame

References

https://www.tutorialspoint.com/java/java_documentation.htm
https://www.oracle.com/java/technologies/javase/javadoc-tool.html
https://docs.oracle.com/javase%2Ftutorial%2Fuiswing%2F%2F/components/index.html
https://docs.oracle.com/javase%2Ftutorial%2Fuiswing%2F%2F/layout/visual.html#box
https://docs.oracle.com/javase/tutorial/uiswing/components/list.html
https://docs.oracle.com/javase/8/docs/api/java/io/BufferedReader.html
https://www.youtube.com/watch?v=6XoVf4x-tag
https://youtu.be/Kmgo00avvEw?si=SBa9-cCUnn-yN6_b
https://sites.google.com/ku.edu.tr/comp132/lecture-notes?pli=1&authuser=1