KOÇ ÜNİVERSİTESİ
COMPUTER ENGINEERING

COMP132: Advanced Programming
Programming Project Report

UNO_Baran_Koçak

Baran Koçak 86703

Second Semester/First Year(2024)

Part 1 General Demo Information:

List of Users

User : “admin”

- Nickname: admin
- Password: 123
- INFORMATION ABOUT EACH USER
- Existing game sessions:
    - ngl
    - niceGame
    - seven player game
    - TestGame
    - TestGame2

User: “Longinus”

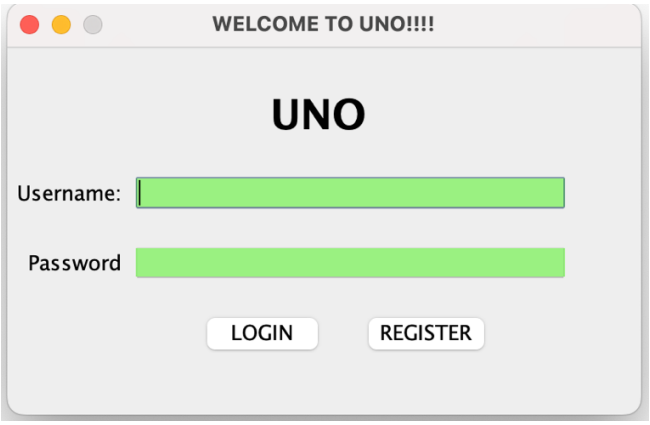
- Nickname: Longinus
- Password: spear
- INFORMATION ABOUT EACH USER
- Existing game sessions:
    - Eva-01
    - Eva-02
    - geofront
    - Human Instrumentality Project
    - Thanatos
    - You can(not) play

User: “MAGI”

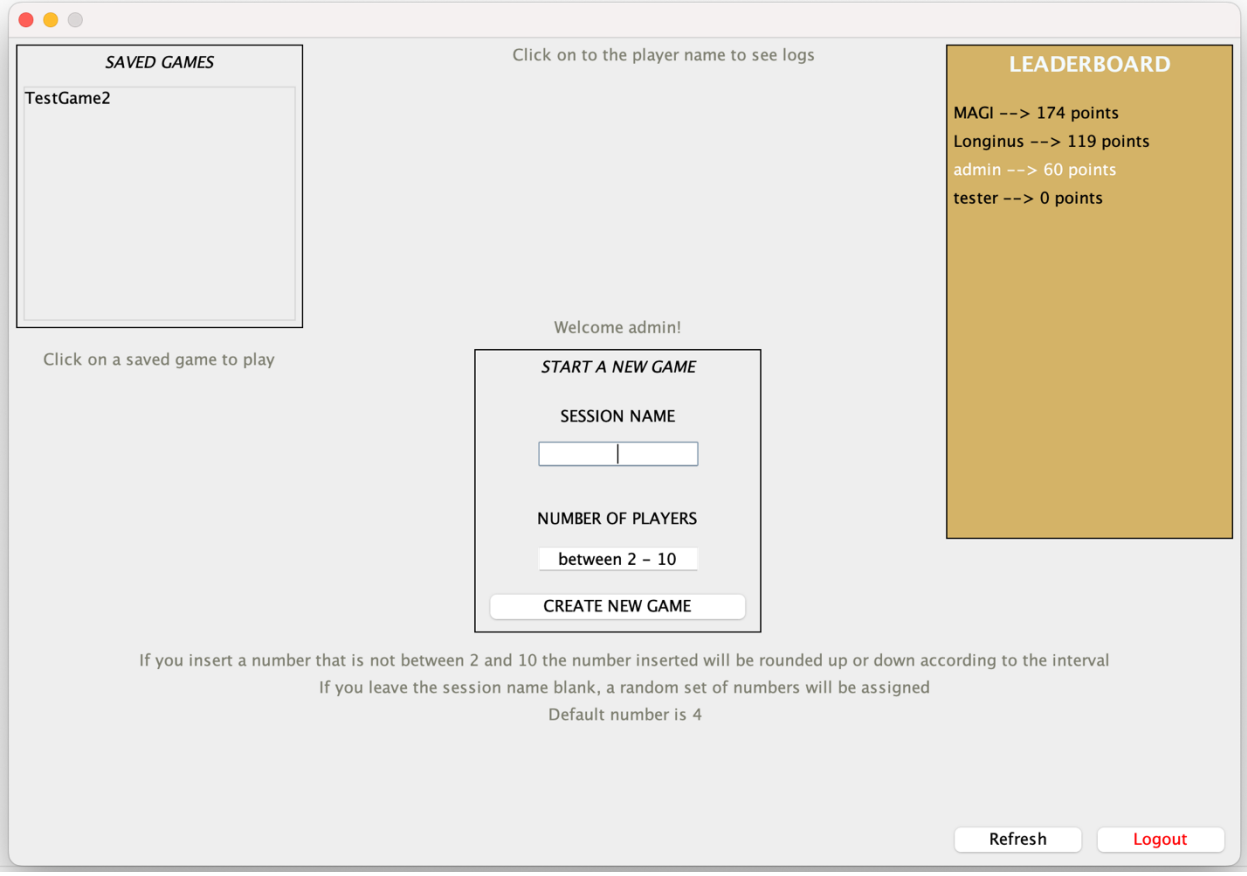
- Nickname: MAGI
- Password: naokoakagi
- INFORMATION ABOUT EACH USER
- Existing game sessions:
    - Balthasar
    - Caspar
    - Melchior
    - Central Dogma

Application Usage Information

When the application is opened a login/register window appears. Through this the user can either create a new account or login to an existing account.



Once the login is successful the main menu window opens. The main menu contains these as shown in the screenshot:



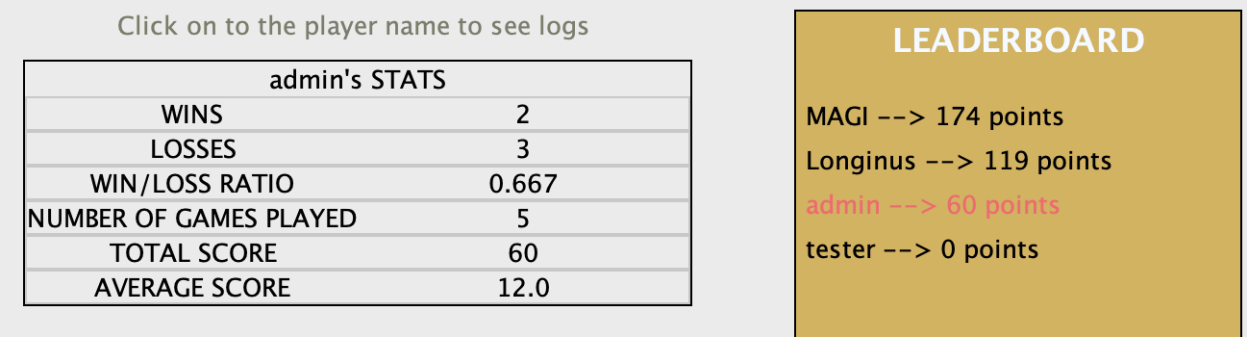
At the centre the user can create a new game by typing in the session name and choosing the number of players that are going to participate in the game. As written below the box there are certain information given. (If the user puts in a number that is not between 2 and 10 the game automatically sets the inserted number to 2 or 10 according to the number inserted. Also typing in characters that are not numbers is not allowed. If the session name is left blank a random session name will be assigned. If number of bots is untouched the game will create a game with 4 players.) By clicking “CREATE A NEW GAME” button user can create and start playing the game.

At the right-bottom corner the user can logout by clicking “Logout” button or can refresh the page by clicking “Refresh” due to some problems that user may face.

At the left-top corner there are saved games which can be opened by simply clicking on the saved game’s name.

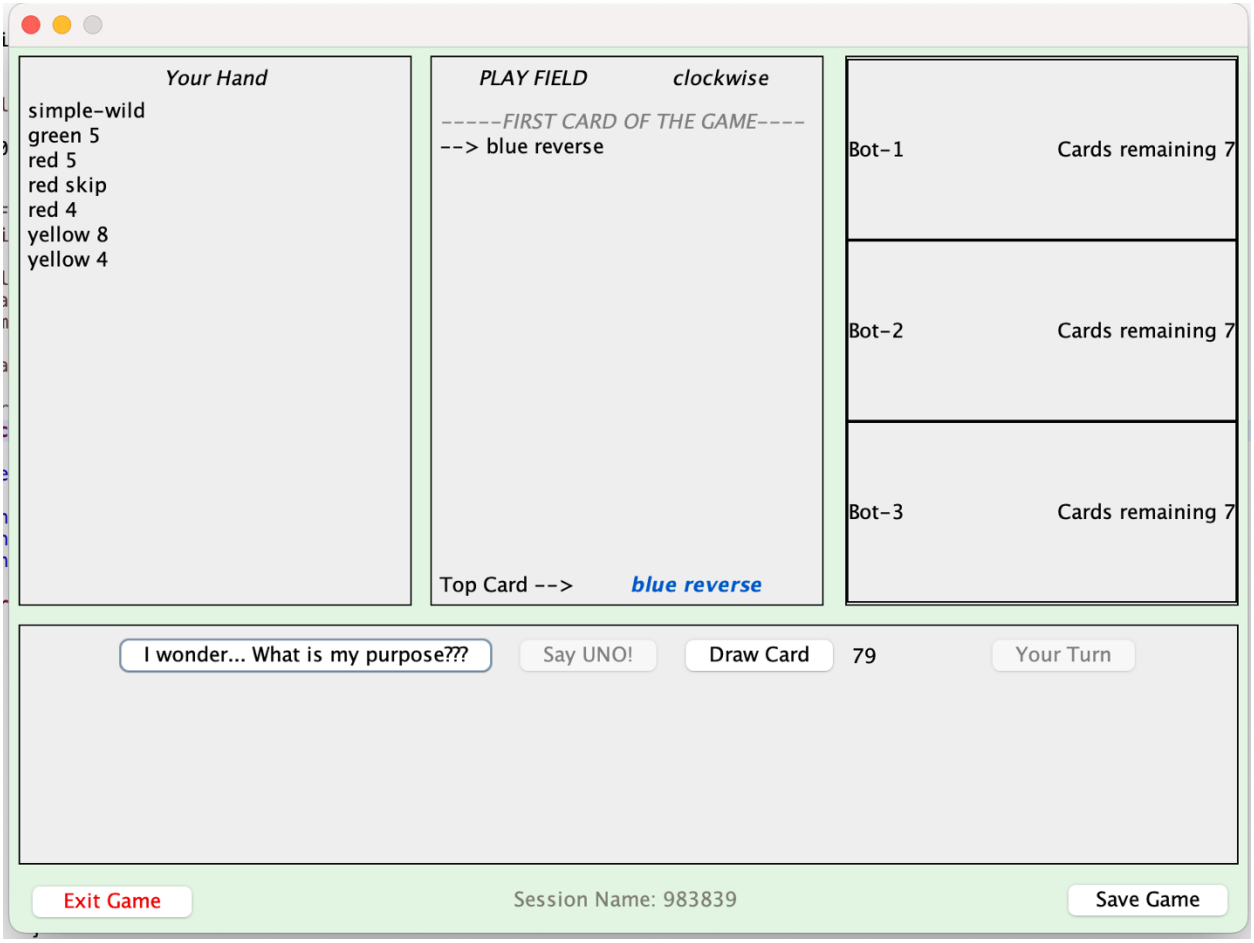
At the right-top corner there is a leader board that consist of users that have registered with total

points they have gained over all games played. By simply just hovering over the username, one can access the stats of the player as shown:



Also, by clicking on the username you can access the logs of played games by the user. A log screen appears upon clicking the username as shown in the first picture. If the user clicks on a game, the user can access that game's log which looks like this (second picture).

After the user creates a new game, game session window appears as shown:

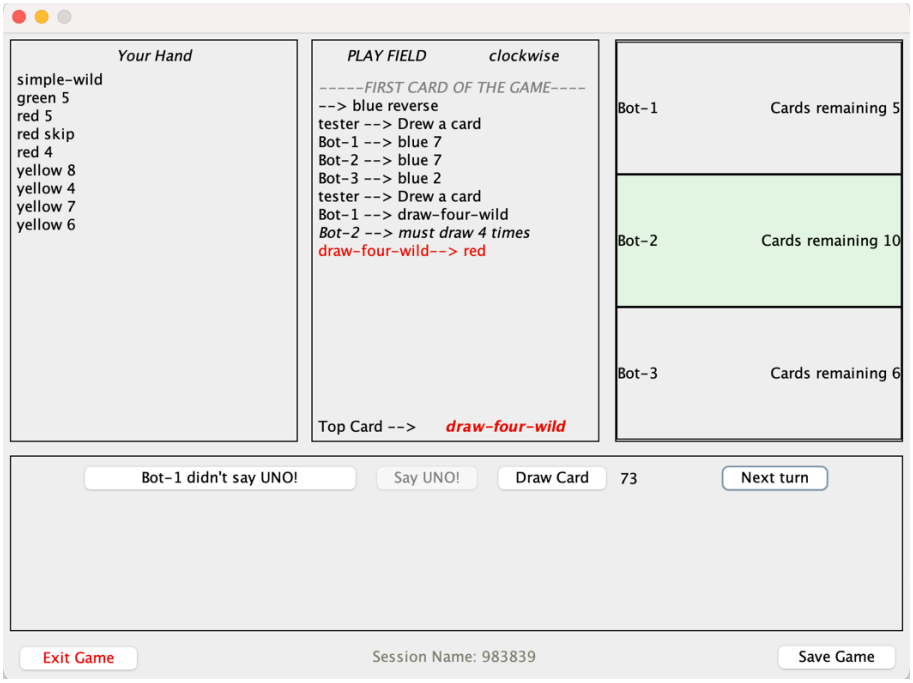


The user then starts to play the game. At the bottom of the window the user can see the session name (the user has left the name blank, so it is assigned to a random session name). The user can save the game or exit the game at any time s(he) wants (bottom). On left the user can see and interact with their cards according to the rules of the game. If the user can play the card it is thrown into the game field. On right the user can see other players(bots) with their names and remaining cards in their hands. On the middle of the window user can see the play field which generally shows what happens in game. In play field panel next to the text “PLAY FIELD” there is an indicator that informs the player the direction of the game. At the bottom of the play filed the top card is shown with colour for better understanding.

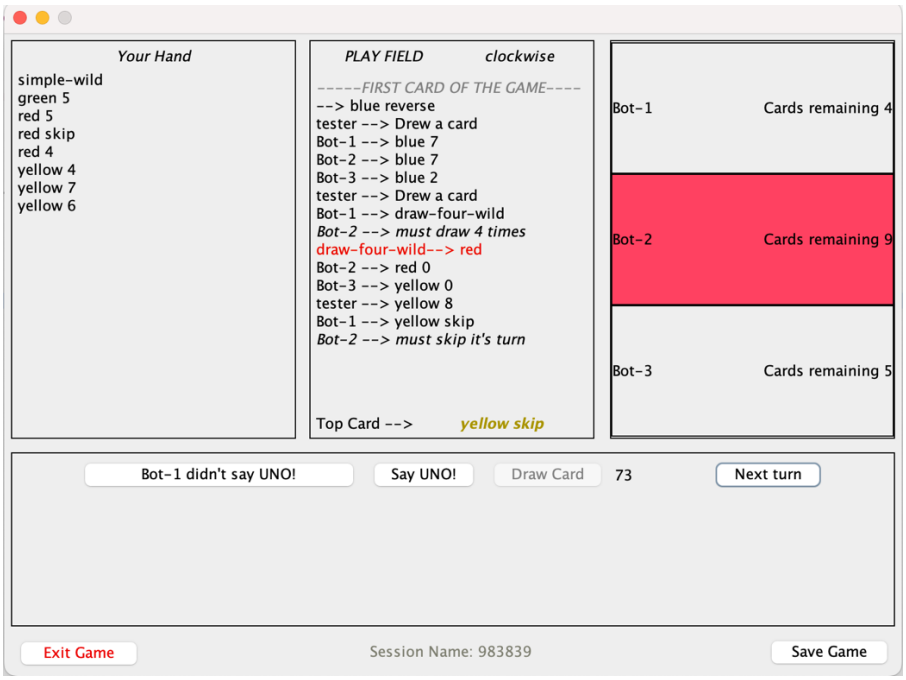
Below all of these there is the action panel for the player. Starting from right: “Next Turn” button (if it’s players turn and the player cannot end his/her turn the text on the button says “Your Turn”, If the player can end it’s turn the text then converts into “End Turn” which ends

the player's turn). A text that shows how many cards are left in the deck (which starts with 108). "Draw Card" button which draws a card from the top of the deck if the player hasn't played a card yet (If the drawn card is not playable the player can draw cards until the player draws a playable card). "Say UNO!" button lets a player say UNO! When s(he) has only one card left (If the player doesn't say uno even though s(he) has one card, If other players(bots) notices the player gets penalized (draw two cards)). At the end there is "<bot name> didn't say UNO!" button which penalize(the player which played last turn) the player(bot) if s(he) didn't say UNO when s(he) had one card left (on the picture it is not written as I have mentioned because no one has played yet once the requirements are met the button automatically changes its text).

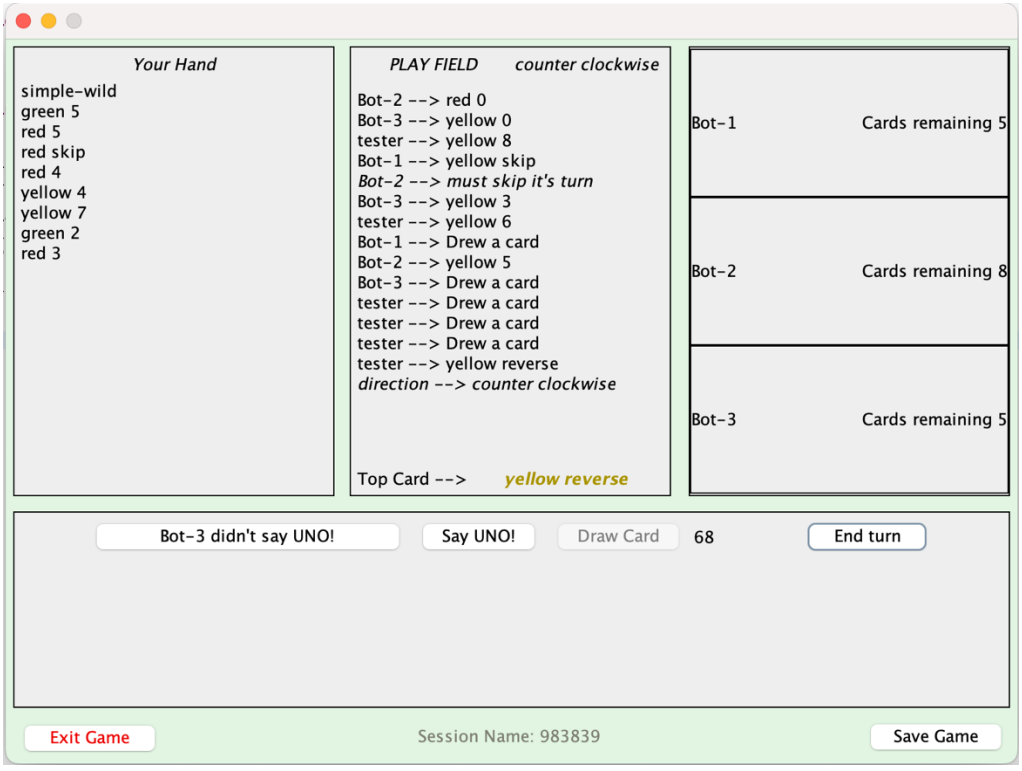
Let me now show some gameplay footages:



After some turns the game looks like this. To understand whose turn, it is one can check the background of other players or the background of the user. If the background is green, it means that the player can play the game (it looks different when h(she) cannot). Also, as I have mentioned before the texts of buttons have changed. The effects of the cards are written in italic font.

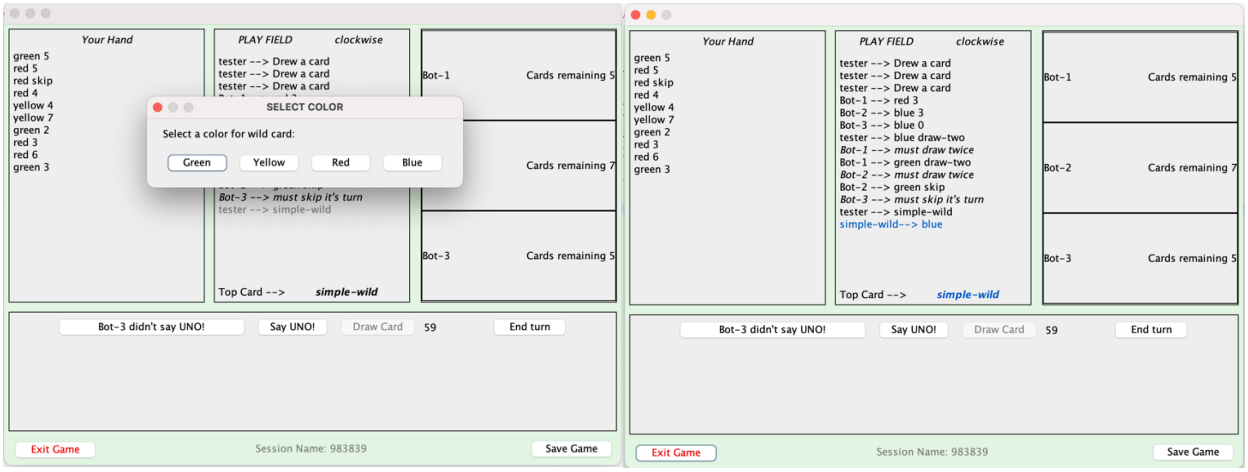


Bot-1 played a skip card which makes the next player skip its turn. The background colour of the player becomes red (the user's background as well).

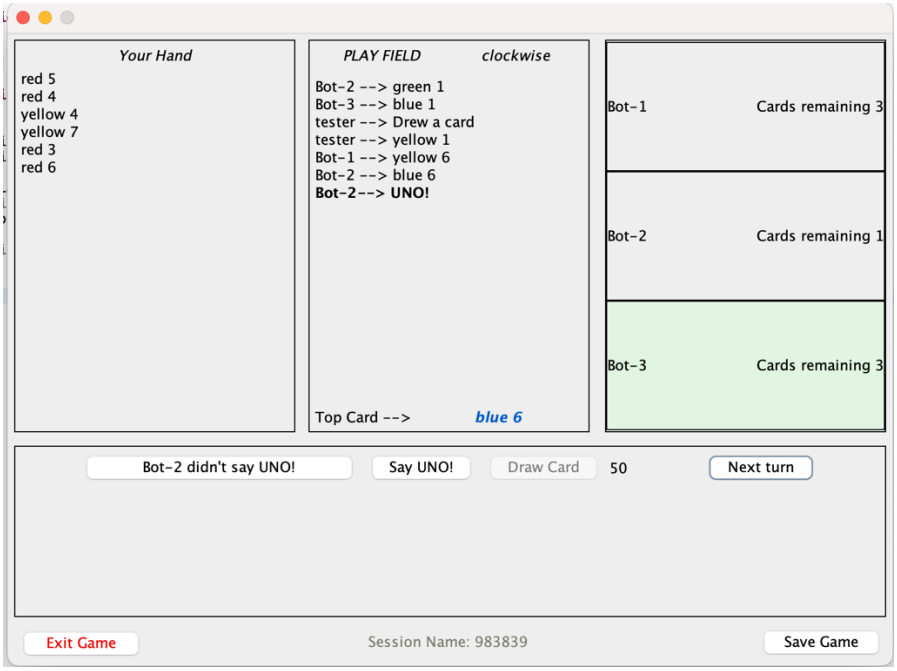


The user played a reverse card which changed the direction of the game(at top clockwise

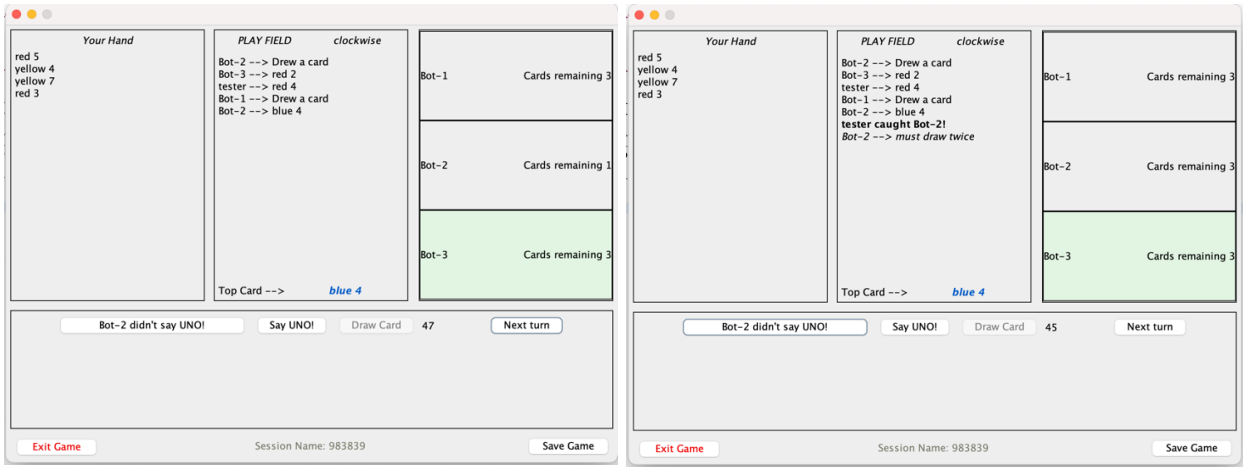
became counter clockwise).



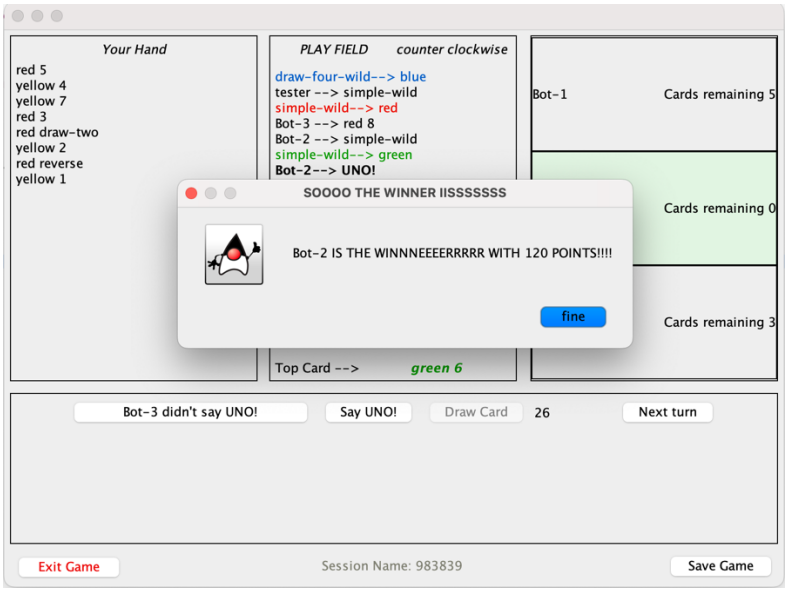
The user played a wild card which allows the player who played it to change the colour of the wild card. The user must choose form 4 options: “Green”, “Yellow”, “Red”, “Blue” then it changes colour accordingly (in the picture above the user choose blue so the colour of the card turned into blue). The AI chooses the colour which s(he) has most of.



Bot-2 played its card and said UNO because s(he) is left with one card. Because s(he) said UNO the user can't penalise the bot.



Bot-2 now has one card but forgot to say UNO. So, the user noticed and pressed the button. By doing so bot-2 was forced to draw twice.



The game ended with the victory of Bot-2. After the game ends a window pops out and declares the winner of the game with the point s(he) gained (which is calculated by the summation of cards' value that were in the hands of players). Upon clicking "fine" the user gets back to the main menu.

PART 2 Project Design Description

Class Relations

In my project I have used these classes and packages:

- accountManagement(package)
  - AccountWizard.java
- cards(package)
  - Card.java(abstract class)
    - ActionCard.java(abstract class and subclass of Card)
      - DrawTwoCard.java(subclass of ActionCard)
      - ReverseCard.java(subclass of ActionCard)
      - SkipCard.java(subclass of ActionCard)
    - NumberCard.java(subclass of Card)
    - WildCard.java(abstract class and subclass of Card)
      - SimpleWildCard(subclass of WildCard)
      - DrawFourWildCard(subclass of WildCard)
- frames(package)
  - BotStatusPanel.java(subclass of JPanel)
  - DisplayLogScreen.java(subclass of JFrame)
  - GameSession.java(subclass of JFrame)
  - LoginScreen.java(subclass of JFrame)
  - MainMenu.java(subclass of JFrame)
  - PlayerStatPanel.java(subclass of JPanel)
- gameCharacter(package)
  - GameCharacter.java
    - Bot.java(subclass of GameCharacter)
    - Player.java(subclass of GameCharacter)
- gameManagement(package)

- DeckManager.java
- LeaderBoardComparator.java(implements Comparator)

❖ thyGameRunner(package)

- GameMaster.java
- Main.java

As I have written above these are the class relations, inheritances, abstract classes and type hierarchies.

GUI Design

I used Java Window Builder (1- reference) to easily design and control java Swing. The GUIs I have used in this project are the following:

❖ LoginScreen.java (JFrame)

I have used JLabel, JTextField, JButton and JPanel(contentPane):

- I have used JLabels to inform the user for what s(he) should do. Also, I have used them to warn the user in the case of unaccepted type of username and password.
- I have used JTextFields to get the user input of "username" and "password".
- Finally, I have used JButton to give the user the choices of logging in and logging out.

(JPanel is there for keeping the element together with an "absolute layout".)

❖ MainMenu.java (JFrame)

I have used JLabel, JTextField, JButton, JPanel, Borders.

- I have used JLabels to inform the user for clarity acting as little tutorials. Also, I have

used them to warn the user to type in appropriate session name. I have used them as interactable components in the leader board and saved games. I made them interactable by adding mouseListeners.

- I have used JTextFields to get the user input of "Session Name" and "Number of Players".
- I have used JButtons to give the user the choices to "Create New Game", "Refresh", "Logout".
- I have used JPanels to keep the components together and make it easier to add new components to a certain place. I have used Absolute Layout in the contentPane, Start new game panel, the border panel for leader board, the border panel for saved games to organize the components in panes freely. I have used Grid Layout with fixed column and rows to list users in leader board. I have used Box Layout to store saved games. I have used Flow Layout to display desired user's stats.
- Finally, I have used borders for a great look and to emphasize zones.

❖ PlayerStatPanel.java (JPanel)

I have used JLabel, JPanel and Borders.

- I have used JLabels to display the users' stats and indicate which number represents what.
- I have used JPanels to keep things organized. The class itself is a JPanel which has Grid Layout with fixed number of columns and rows. Inside the main JPanel there are little JPanels that also has Grid Layout to keep things organized.
- Finally, I have used borders for a great look and to emphasize zones.

❖ DisplayLogScreen.java (JFrame)

I have used JLabel, JPanel, JScrollPane and Borders.

- I have used JLabels to display game sessions and to corresponding log texts. Also, I have

used to navigate the user.

- I have used JPanels to keep things organized. I have used Absolute Layout in contentPane, game session's border, log border. I have used Grid Layout with fixed columns and rows to list game sessions. I have used Box Layout to display log that I add line by line from log.txt of the corresponding game session.
- Inside the panel that I use to display log, I added JScrollPane to let user to see all the log file.
- Finally, I have used borders for a great look and to emphasize zones.

❖ GameSession.java (JFrame)

I have used JLabel, JButton, JPanel, JScrollPane and Borders.

- I have used JLabels to inform the user for clarity acting as little tutorials. I also have used them as cards. I have added mouse listener to make them interactable and with necessary adjustments I have made them playable as well. More is explained in gameplay section of the report.
- I have used JButtons to let user play the game and perform actions. Also, I have added them as "Exit Game" and "Save Game" buttons.
- I have used JPanels to keep things together and a zone to add other components to certain positions. I have used Absolute Layout in all border panels to arrange them freely. I have used Box Layout for Player's hand and the game field. I have used grid layout to display bots as well.
- I have used JScrollPane in player's hand in case it becomes full but didn't use it in play field because I wrote an algorithm that would delete past actions to display new ones.
- Finally, I have used borders for a great look and to emphasize zones.

❖ BotStatusPanel.java (JPanel)

I have used 1 panel and 2 JLabels. The panel has grid layout. JLabels are used to display bot's name and remaining cards.

Text file processing details

I have used text files to store information. To manage accounts and text files I used AccountWizard class. Inside the class there are methods that handle .txt related stuff:

- Login and Register

  I get inputs from JTextFields and check if the inputs are:

  - Are there any blank inputs?
  - Does that username already exist? (register)
  - What is the length of the username (username cannot be more than 10 characters)
  - Are there any accounts for the username (login)
  - Is the password correct?

  If the user enters appropriate username and password the account is created, or the user is logged in.

  I have stored usernames and passwords like this:

  Username∑password

  Inside AccountWizard I splitted the line and got the username and password separately to login the player.

- Stats

  I have two methods for stats one creates the file the other updates it. I stored Wins, Losses and points gained per game like this with 2 lines:

  0∑0∑1∑1∑1∑0∑

23∑0∑43∑90∑11∑0∑48

In the first line 0 represents loss 1 represents win. On the second line each number represents the score gained that game. To get information from the text file I splitted lines and displayed them in a panel.

- Log

To record log, I have used 2 methods first was to append to the log string, second was to add the final log to the game folder. Every time a game character does something It displays on the screen and just after it displays, it also adds to the log string. When the game finishes the second method is called and the log is saved into a text file.

- Save Game

To save the game I have stored: session name, bot count, player's hand, bots' hands, the turn index, the game direction, the state of the deck, discard pile and who has said uno. An example of save.txt:

```
TestGame2
1
R6∑RG
R4∑BT∑B4∑YT∑R5∑Y5∑R1
0
-1
Y7∑Y1∑Y5∑G1∑B1∑G2∑WJ∑Y2∑RT∑B1∑G1∑Y3∑WJ∑B8∑R8∑WF∑Y3∑R1∑B0∑RS∑YS∑Y4∑R0∑G4∑Y7∑Y2∑R3∑B7∑BG∑B6∑R2∑WJ∑B3∑R9∑Y8∑G5∑Y0∑R2∑BS∑G7∑Y1∑YG∑YG∑G3∑G5∑WF∑BG∑BT∑GT∑GG∑GG∑R3∑G6∑R7∑GS∑R6
B9∑B5∑B5∑RJ∑YF∑Y9∑Y6∑YS∑RS∑RT∑R9∑GF∑G9∑G6∑G4∑G8∑B8∑B2∑B2∑BS∑GS∑G8∑R8∑R5∑R7∑RG∑R4∑B4∑Y4∑YT∑Y9∑Y8∑Y6∑B6∑B7∑B9∑B3∑G3∑G0∑GT∑G7∑G9∑G2
00
```

- Session name
- Bot count
- Player's hand (first character indicates the colour, second character indicates the type)
- Turn index (who's turn is it?)
- The game direction (+1 clockwise, -1 counter clockwise)
- Current state of deck (same rule as player's hand)
- Current state of discard pile (same rule as player's hand)
- Who said uno (every player is stored in the same line according to the indexes of

characters)

Game Session Loop Implementation

I created a character list to store the ordering of the players thanks to this I had set indexes. Through next turn button the user was able to make the game go on. By incrementing index number every turn, I was able to determine who's turn it is. When it would exceed the number of players, the program sets the index back to the interval. I have made it go forward when it is in clockwise and go backward when it is in counter clockwise.

Computer Bot Implementation

When it is bot's turn the bot iterates through its hand and plays the first card playable if not the bot draws a card and if it is playable, it plays it else ends its turn. Also, when the bot plays a wild card it makes the colour of which it has the most. Each bot has 22% chance to detect If a player forgets to say UNO. This makes it harder to get away with not saying UNO while playing with more players. Also each bot has 60% chance to say UNO when it has one card left.

References

I have used 1- video to learn all about swing. I have used others generally for other coding related questions.

1- https://youtu.be/Kmgo00avvEw?si=GNaJVqHoJucX0Lf3
2- https://www.geeksforgeeks.org
3- https://stackoverflow.com