CS12320 MAIN ASSIGNMENT

Patience

Word Count : 2199

MAY 20, 2022

JAC105@ABER.AC.UK

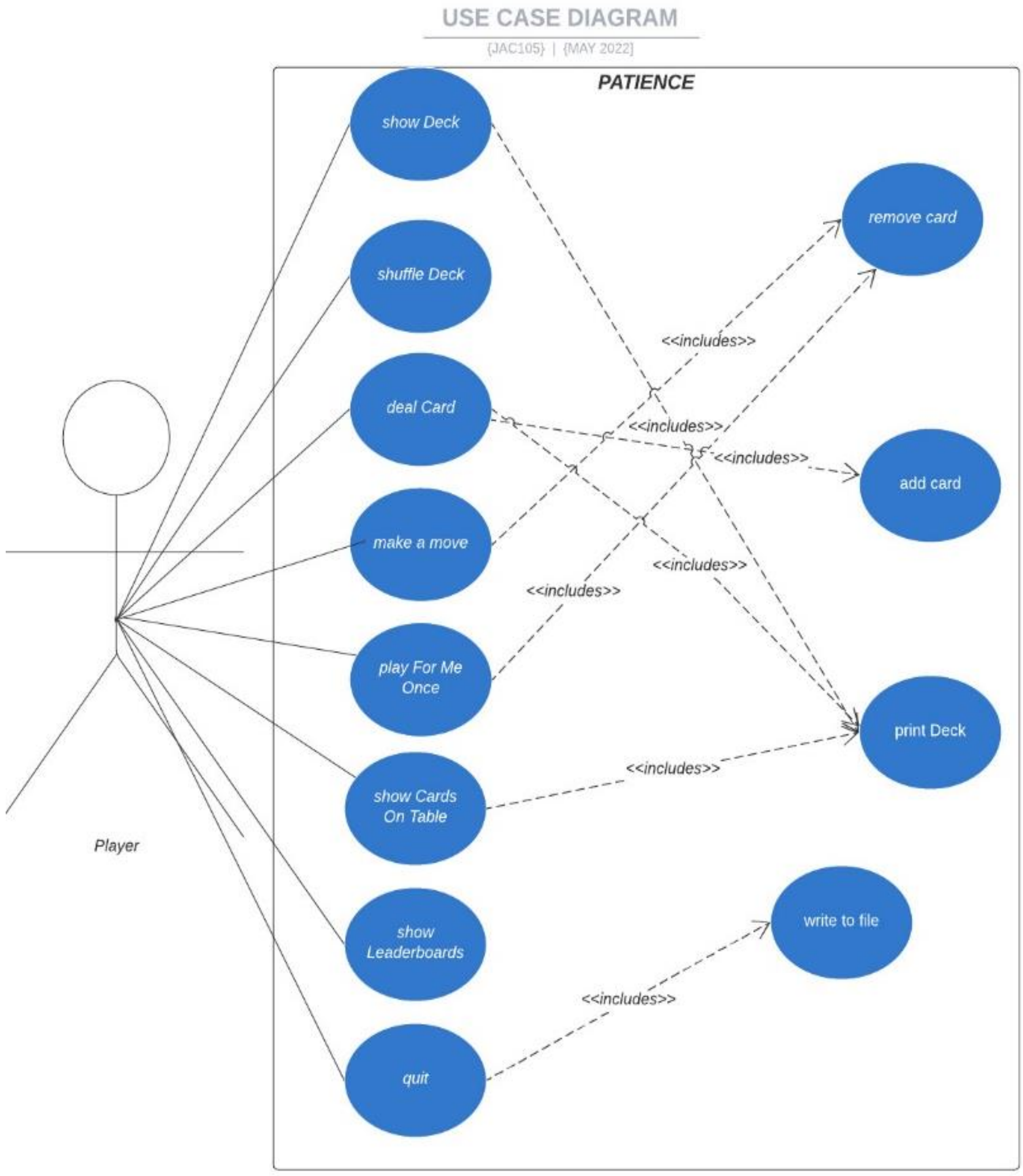
Table of Contents

Introduction

This report will cover a full documentation of the design, testing and evaluation of the Patience application. The purpose of this application is to allow a player to play a game of patience through the implementation of classes and methods. A Game, Card and Player class were created, which utilises the methods that are created to be able to reach these functional requirements:

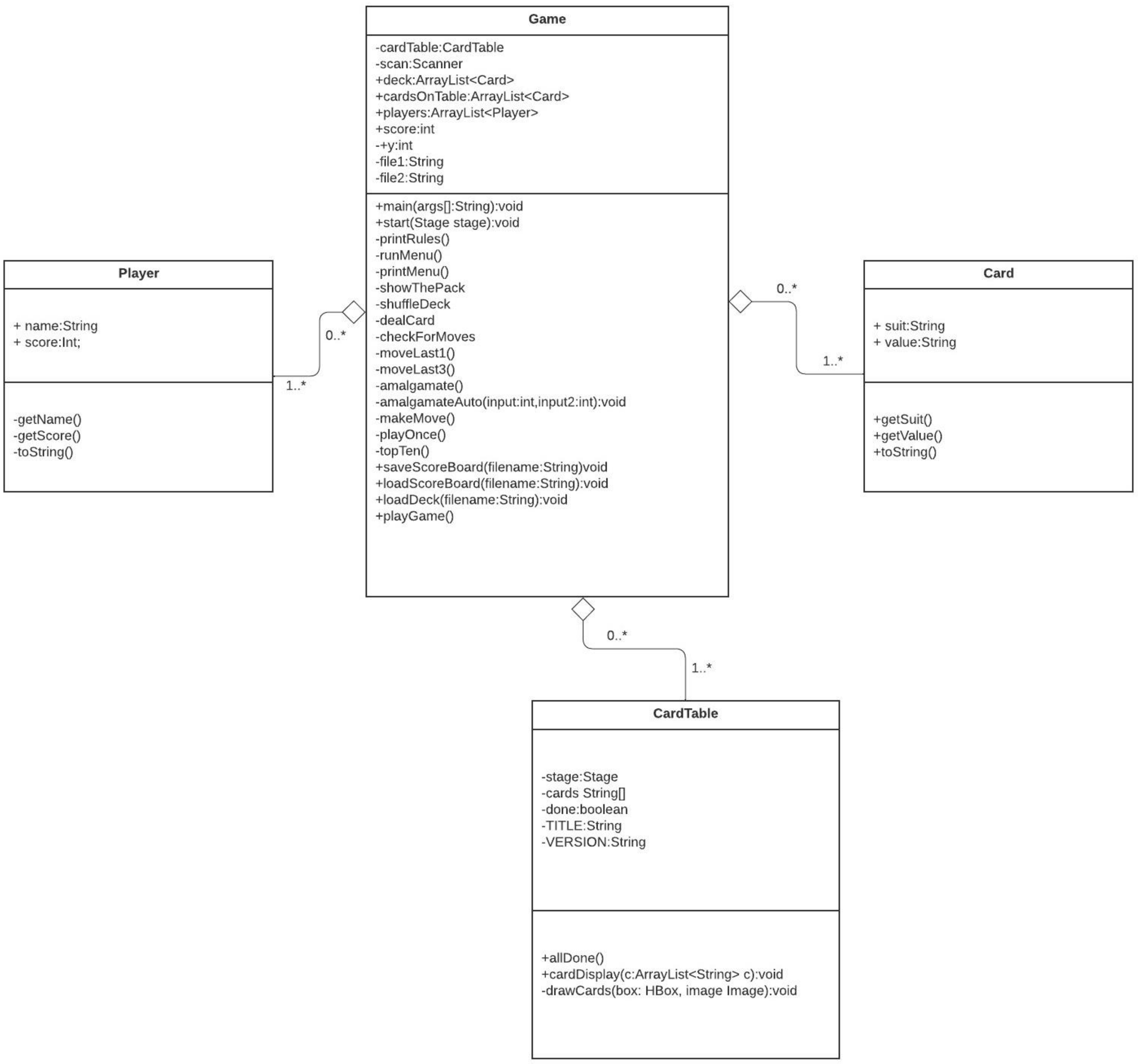
- Show the deck
- Shuffle the deck
- Draw a Card
- Move last pile one
- Move last pile over two
- Amalgamate by position
- Show all displayed cards in text form
- Play for me once
- Show top ten results, sorted
- Quit

Use Case Diagram



Design

Class Diagram



Class Description

Game

This class is the main interface of the Patience application, it uses both Card and Player and its own methods to be able to provide the main functionality of the program. It utilises a scanner to take the input from the player as most of the methods rely on the players input. In addition, there are two Array Lists that store the cards of the deck and the cards on the table, which most methods are implementing. Game consists of these methods:

- **printRules** - prints out all the rules of the game, called within playGame (see below).
- **printMenu** – prints a menu for the user to play the game.
- **runMenu** – allows for all the options in print menu to be taken as inputs and run each method through a switch statement.
- **loadDeck** – takes a string text file using a scanner and file reader, taking every line of text and filling in the parameters value and suit, creating a new Card every two lines. These card objects are stored in the array list deck, until there is a line without text. This also throws an IO Exception in case of the wrong input or output being produced.
- **showThePack** – utilises a for loop to display all cards within the deck.
- **shuffleDeck** - utilises the imported java.util.Collections package's collection.Shuffle method to shuffle the deck. Also, this method will print a message telling the user that the cards have now been shuffled.
- **dealCard** – takes the card at the last index of the array list deck and adds it to the array list cardsOnTable. This will then print out the cardsOnTable array.
- **moveLast1** – removes the card at the index one less than the size of the array list in cardsOnTable.
- **moveLast3** – checks to see there is more than three cards are in cardsOnTable , then creates two cards lastCard and thirdLast and sets there suit and value as the cards at the last index and third from last index , respectfully. It then checks if lastCard and thirdLast share either the same suit or value by using the getSuit and getValue methods, if these parameters match then the card at the index of the size of the cardsOnTable minus three is removed, and lastCard is added to cardsOnTable at this index. Finally the last card in cardsOnTable is removed.

- **amalgamate** – utilises similar fundamentals of moveLast3 but takes the users inputs, using a scanner, instead of the last card index and the one three from it as the indexes and cards that are created. It then follows the same principles of moveLast3 to do the amalgamation.
- **makeMove** – reduces score by 10, prints an option list of the three types of moves previously described and uses a scanner to take the players input as an int and comparing this in if and else if statements to select a specific move and run that method. An else statement is used to print an error message that reads "INVALID INPUT".
- **amalgamateAuto** – takes two int parameters, random numbers are generated from the playOnce method and inserted (see below). These are used as the indexes input that the user usually does for amalgamate but removes the print statements.
- **playOnce** – creates random object choice and creates three integers. These integers are set as choice.nextInt. The first number is used to choose the type of move. The next two numbers are used for the parameters of amalgamateAuto if this is the option chosen. An additional 10 points are deducted from the player.
- **loadScoreBoard** – takes a text file as a string and reads everything on the first line separated by spaces, placing every string followed by an int in a new Player objects parameters of value and suit, respectfully. It saves this in the players array list until there is no longer an int followed by a string or vice versa. Throws an IOException
- **saveScoreBoard** – takes a string as a parameter (text file) and will write all the items in players to the text file, parameter by parameter, on the same line. Before writing to the file, it creates a new Player and asks for the user to input their name and assigns the score variable as the score parameter of player. This player is added to the players array list then saved. Throws IOException.
- **topTen** – utilises Javas Collections.sort and a compareTo method (see *Player* class) to order the players array list by score with the highest score being in the lowest index, and then prints the first 10 players.
- **playGame** – uses loadDeck and loadScoreBoard. Afterwards, it calls printRules, then initialises the array list cardsOnTable and calls runMenu. Finally, after runMenus loop is broken with input 'q' , saveScoreBoard is called and scoreboard.txt's information is inserted as the parameters.
- **Main** – initialises the whole application by creating a new Game and then calling the playGame method onto this new game. Finally, it runs the application.launch method to start the graphical interface. When that is closed so is the application.

Card

This class is used for the creating of a card object and converting it to a string. Without this class the whole application would not be able to run as it is an essential component of any card game. It is the type, of the two main array lists that are used in almost every function within the Game class. Here are its methods:

- **getSuit** – retrieves and returns the suit of the card object
- **getValue** – retrieves and returns the value of the card object
- **toString** – returns the card object as a string combined of the value and suit. Therefore, utilises getSuit and getValue.

Player

This class is used to create a player object that has a name and score as its parameters and is used within the Game class to be used for storing and retrieving the data needed to operate the scoreboard and display the top ten players functional requirements. Here are its methods:

- **getName –** retrieves and returns the name of the player
- **setName** – sets the name of a player
- **getScore** – retrieves and returns the score of the player
- **toString** – returns the player as a string containing the players name and score.
- **compareTo** – compares a player to other players by integers, which is used in Game class to sort the scoreboard to produce the top ten scores.

CardTable

This class is a class that was already made within the assignment template. Its used for creating a stage and setting the attributes of the stage. This is the class which deals with the GUI. Here are its methods:

- **allDone** – called when a player quits the game, stops face-down pack from being shown
- **cardDisplay –** displays the cards graphically
- **drawCards –** draws a card from the deck. This utilises cardDisplay and ImageView to display the card and resize its image to fit the graphical interface

Pseudo Code For Amalgamations

```
Method amalgamate
    Print(“which card do you want to put on top of another, give its position”)
    Int input = scan.nextInt-1
    Print(“which card is this being placed on give its position”)
    Int input2 = scan.nextInt
        If input is bigger than input2 and input2 == input – 3 and input2>0
            new Card rightCard = cardsOnTable.get(input)
            new Card leftCard = cardsOnTable.get(input2)
            if  rightCards suit equals leftCards suit or rightCards value equals leftCards
            value
                cardsOnTable.remove(card at index of input2)
                cardsOnTable.add(right card at the index of input 2)
            else
                print(“THIS IS AN ILLEGAL MOVE”)
        else
            print(“THIS IS AN INVALID OPTION”)
End method
```

Testing

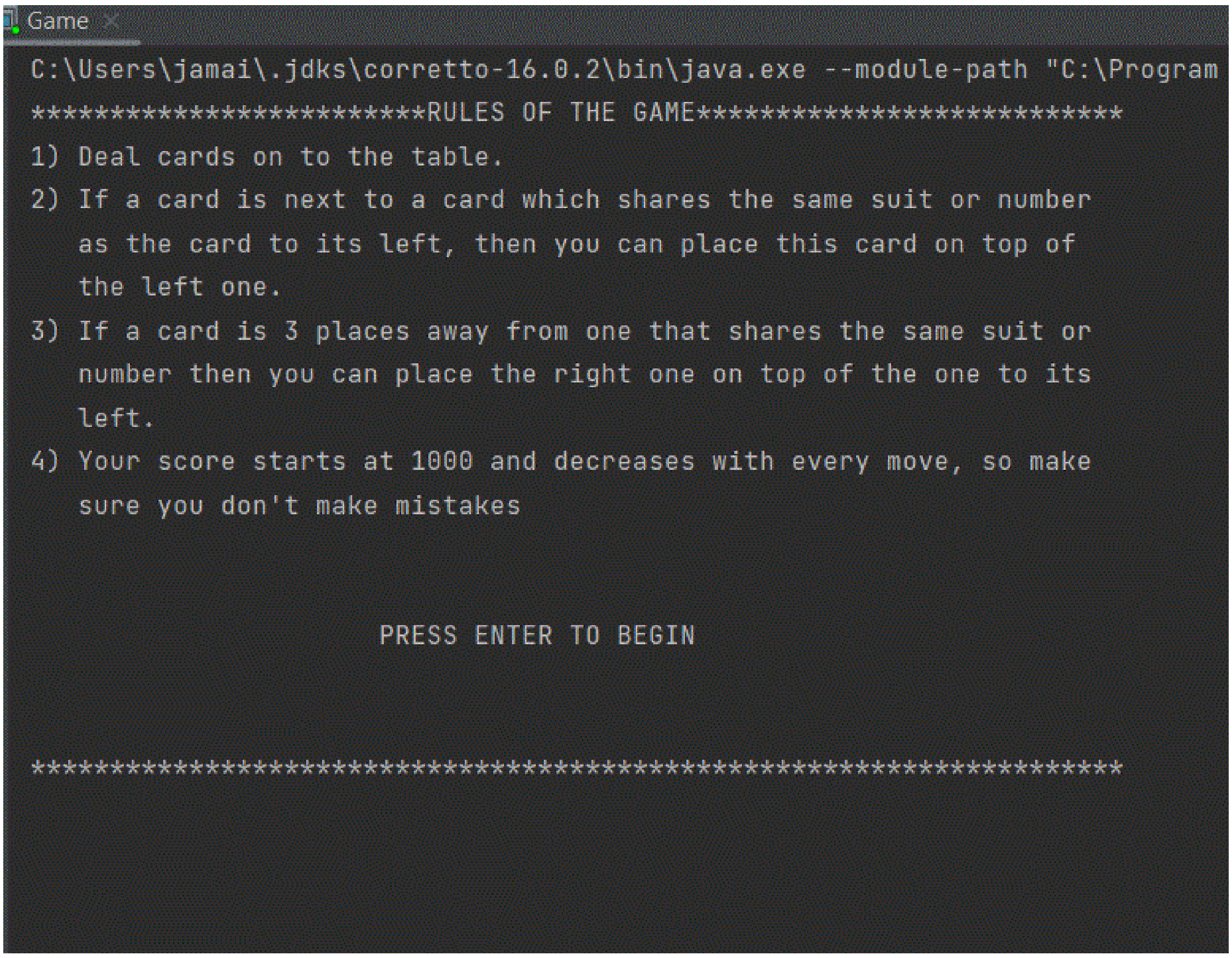
Test Table

ID	Requirement	Description	Expected Outcome	Pass/Fail	Comments
FR1	Show the Pack	Display all the cards from array list	Shown in Figure 2	y	
FR2	Shuffle Cards	Shuffle the array list deck	Shown in Figure 3	y	
FR3	Deal a Card	Remove a card from deck and adds it to cardsOnTable	Shown in Figure 4	Y	
FR4	Move last pile one	Move last card in cardsOnTable to, and replace, the index before its position.	Shown in Figures 5 & 6 & 7	Y	
FR5	Move last pile over two	Move last card in cardsOnTable array to, and replace, the index three places before its position.	Shown in Figure 8 & 9 & 10	Y	
FR6	Amalgamate by position	Takes two indexes in the middle of cardsOnTable and move the card at the larger index to, and replace, the card at the position of the smaller one	Shown in Figure in figure 11 & 12 & 13	Y	

FR7	Show all displayed cards in text form	Print out the array cardsOnTable	Shown in Figure 14	Y	
FR8	Play for me once	Get the application to make a move for the player	card is removed and replaced on its own without choosing the type of move	N	This function runs but doesn't do its intended functionality
FR9	Show top 10 results, sorted	Print out the array list players and sort them by scores	Shown in figure 16	Y	
FR10	Quit	Quit and end the game	Shown in figure 17	Y	
NRF 2	Showing cards in graphical interface	Display the cards visually in a graphical interface	Shown in figure 17	Y&N	Although this requirement is met the game isn't playable with a graphical interface
NFR 3	Storing scores to display sorted top 10 scores	Write all players info to a text file	Shown in figures 18 & 19	Y	

Screenshots

FR1

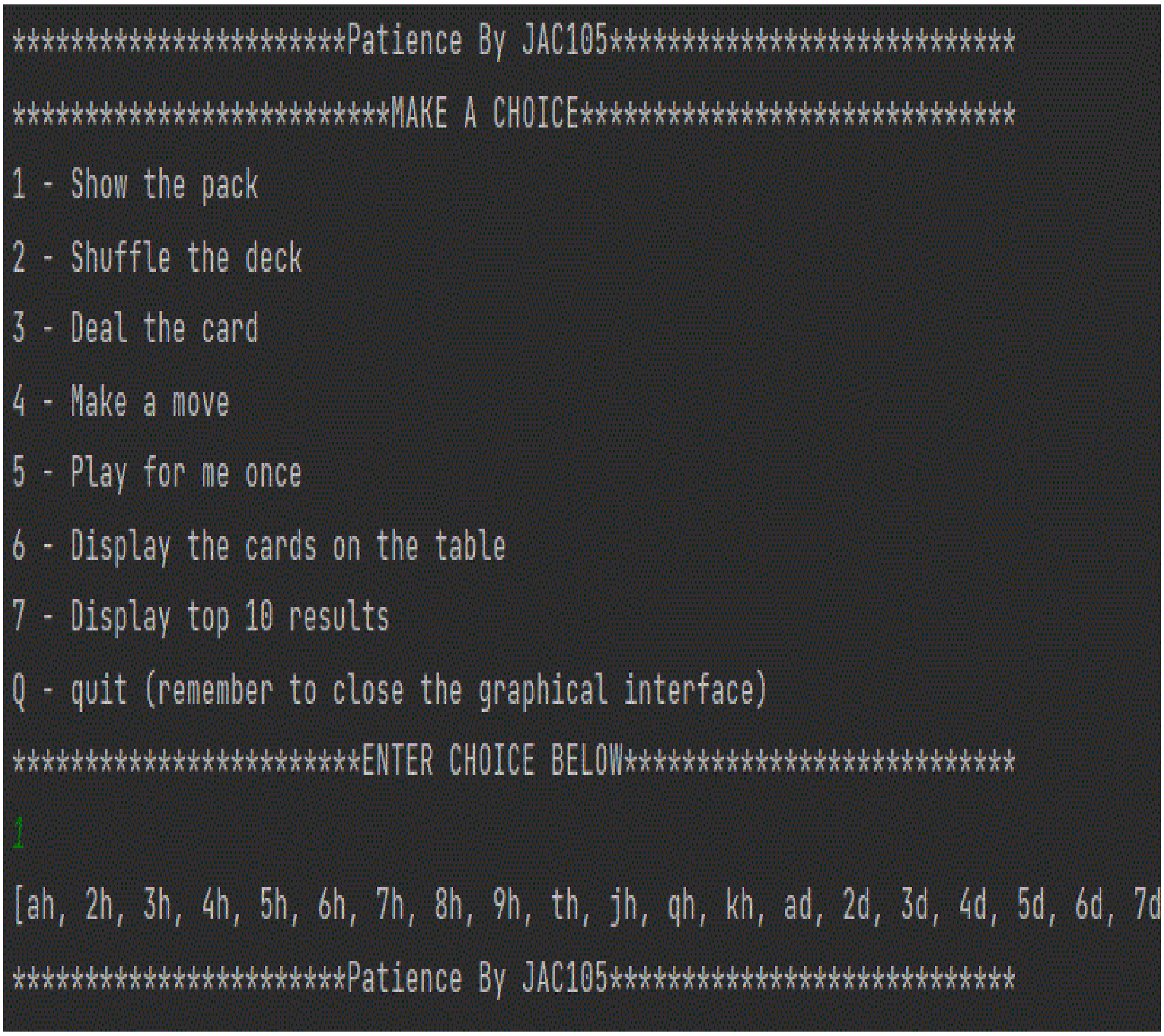


Game

```
C:\Users\jamai\.jdks\corretto-16.0.2\bin\java.exe --module-path "C:\Program
*************************RULES OF THE GAME****************************
1) Deal cards on to the table.
2) If a card is next to a card which shares the same suit or number
   as the card to its left, then you can place this card on top of
   the left one.
3) If a card is 3 places away from one that shares the same suit or
   number then you can place the right one on top of the one to its
   left.
4) Your score starts at 1000 and decreases with every move, so make
   sure you don't make mistakes


                      PRESS ENTER TO BEGIN



*********************************************************************
```

[Figure 1]

Showing the printRules method running and working.

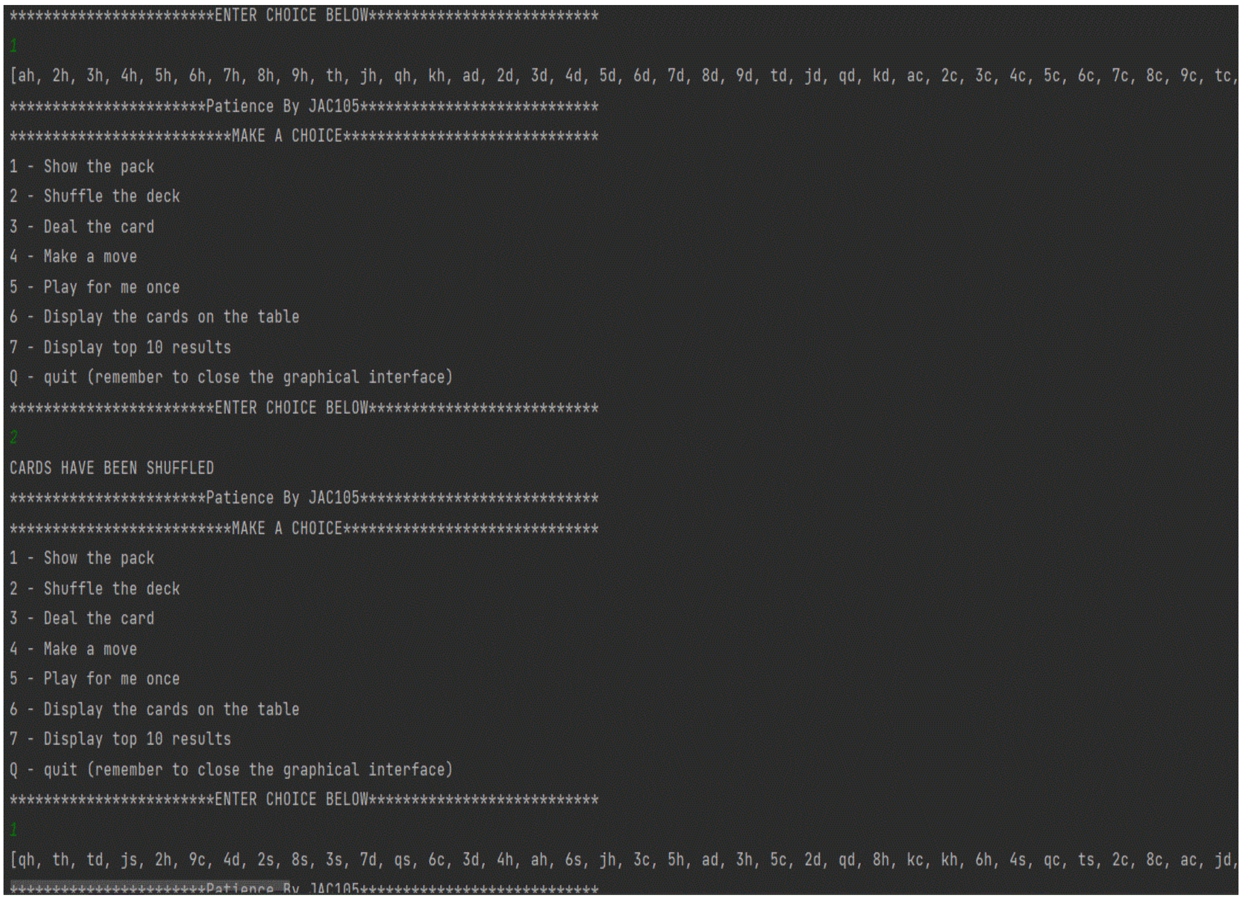


```
**********************Patience By JAC105***************************
*************************MAKE A CHOICE****************************
1 - Show the pack
2 - Shuffle the deck
3 - Deal the card
4 - Make a move
5 - Play for me once
6 - Display the cards on the table
7 - Display top 10 results
Q - quit (remember to close the graphical interface)
***********************ENTER CHOICE BELOW*************************
1
[ah, 2h, 3h, 4h, 5h, 6h, 7h, 8h, 9h, th, jh, qh, kh, ad, 2d, 3d, 4d, 5d, 6d, 7d
**********************Patience By JAC105***************************
```

[Figure 2]

Showing the printMenu, runMenu and showThePack methods all working.

FR2

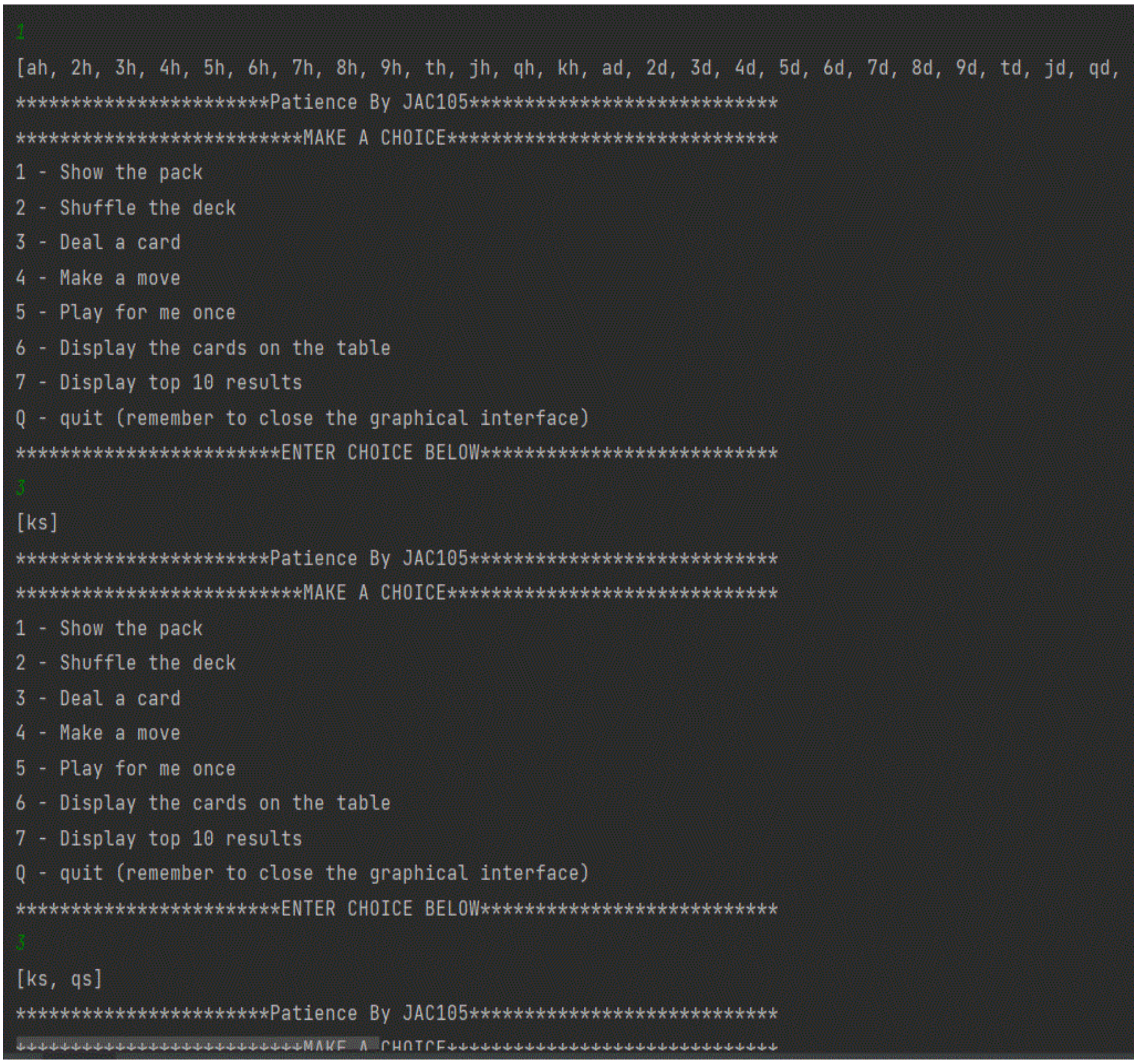


```
***********************ENTER CHOICE BELOW**************************
1
[ah, 2h, 3h, 4h, 5h, 6h, 7h, 8h, 9h, th, jh, qh, kh, ad, 2d, 3d, 4d, 5d, 6d, 7d, 8d, 9d, td, jd, qd, kd, ac, 2c, 3c, 4c, 5c, 6c, 7c, 8c, 9c, tc,
**********************Patience By JAC105***************************
*************************MAKE A CHOICE*****************************
1 - Show the pack
2 - Shuffle the deck
3 - Deal the card
4 - Make a move
5 - Play for me once
6 - Display the cards on the table
7 - Display top 10 results
Q - quit (remember to close the graphical interface)
***********************ENTER CHOICE BELOW**************************
2
CARDS HAVE BEEN SHUFFLED
**********************Patience By JAC105***************************
*************************MAKE A CHOICE*****************************
1 - Show the pack
2 - Shuffle the deck
3 - Deal the card
4 - Make a move
5 - Play for me once
6 - Display the cards on the table
7 - Display top 10 results
Q - quit (remember to close the graphical interface)
***********************ENTER CHOICE BELOW**************************
1
[qh, th, td, js, 2h, 9c, 4d, 2s, 8s, 3s, 7d, qs, 6c, 3d, 4h, ah, 6s, jh, 3c, 5h, ad, 3h, 5c, 2d, qd, 8h, kc, kh, 6h, 4s, qc, ts, 2c, 8c, ac, jd,
**********************Patience By JAC105***************************
```

[Figure 3]

Showing an unsorted deck and the function shuffleDeck working so the deck is now in a random order.

FR3

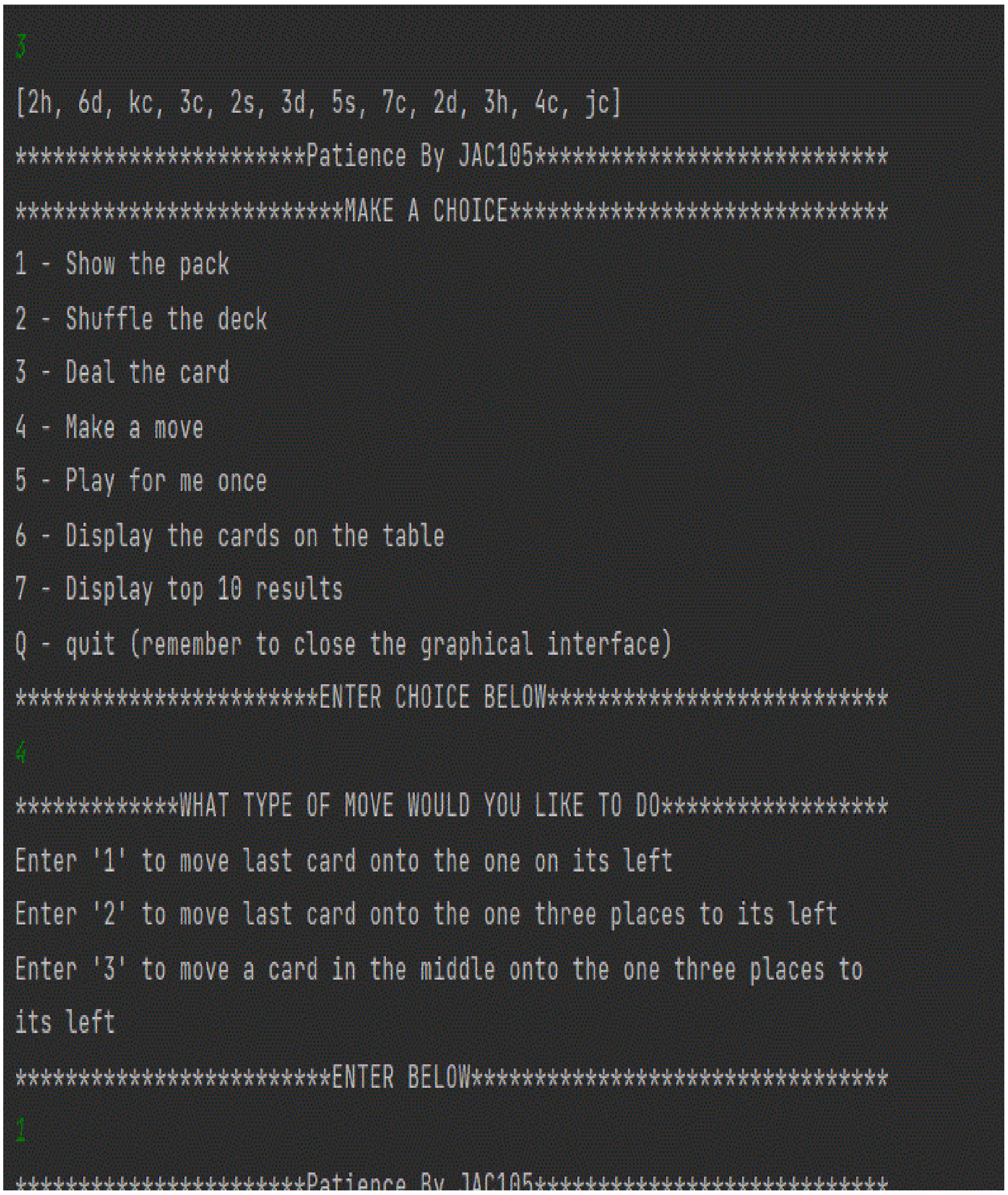


```
1
[ah, 2h, 3h, 4h, 5h, 6h, 7h, 8h, 9h, th, jh, qh, kh, ad, 2d, 3d, 4d, 5d, 6d, 7d, 8d, 9d, td, jd, qd,
***********************Patience By JAC105****************************
**************************MAKE A CHOICE******************************
1 - Show the pack
2 - Shuffle the deck
3 - Deal a card
4 - Make a move
5 - Play for me once
6 - Display the cards on the table
7 - Display top 10 results
Q - quit (remember to close the graphical interface)
************************ENTER CHOICE BELOW***************************
3
[ks]
***********************Patience By JAC105****************************
**************************MAKE A CHOICE******************************
1 - Show the pack
2 - Shuffle the deck
3 - Deal a card
4 - Make a move
5 - Play for me once
6 - Display the cards on the table
7 - Display top 10 results
Q - quit (remember to close the graphical interface)
************************ENTER CHOICE BELOW***************************
3
[ks, qs]
***********************Patience By JAC105****************************
**************************MAKE A CHOICE******************************
```

[Figure 4]

Showing dealCard method working, by removing the last item of the array list deck and adding it to the array list cardsOnTable.

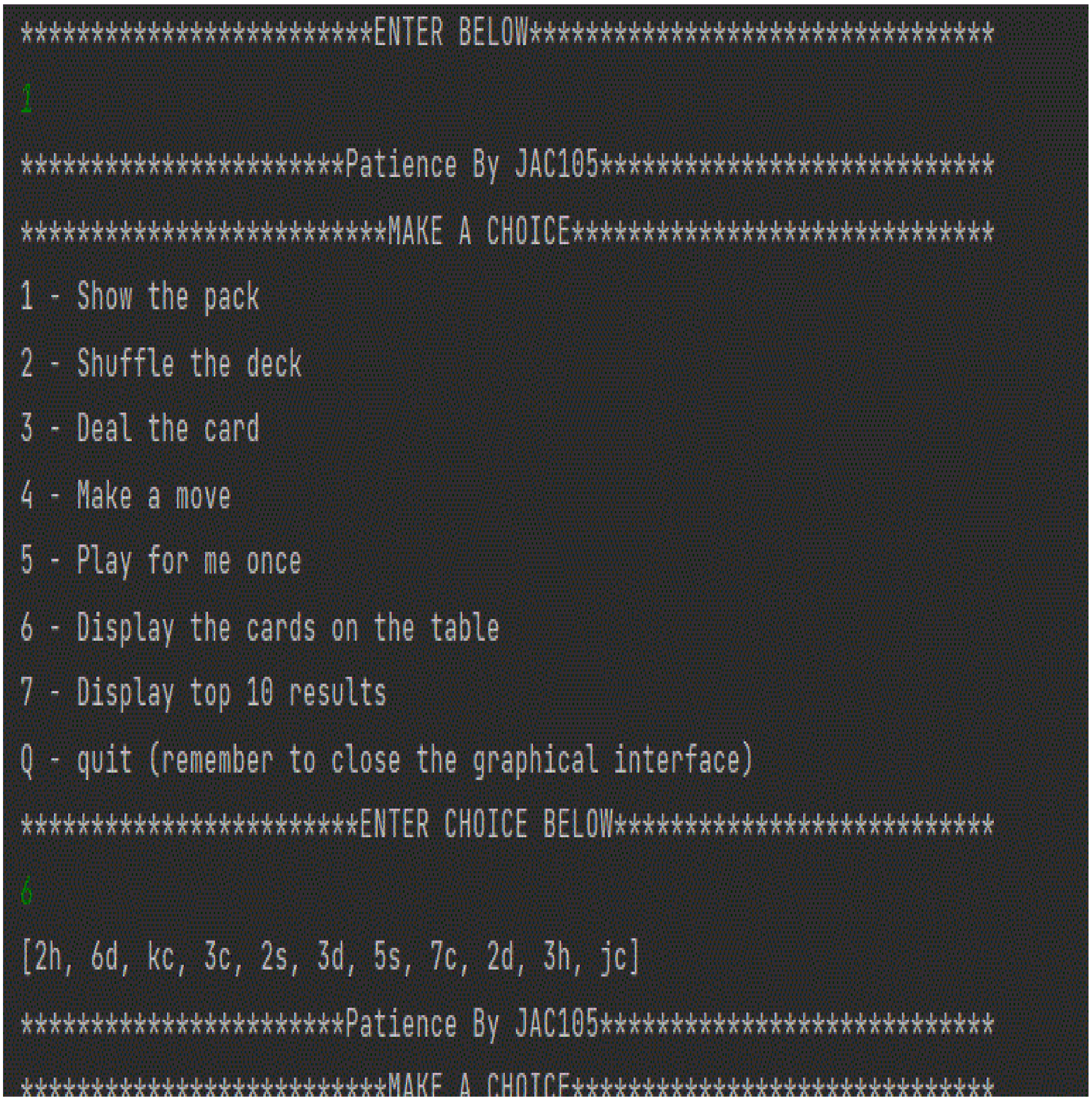
FR4



```
3
[2h, 6d, kc, 3c, 2s, 3d, 5s, 7c, 2d, 3h, 4c, jc]
***********************Patience By JAC105*****************************
**************************MAKE A CHOICE******************************
1 - Show the pack
2 - Shuffle the deck
3 - Deal the card
4 - Make a move
5 - Play for me once
6 - Display the cards on the table
7 - Display top 10 results
Q - quit (remember to close the graphical interface)
************************ENTER CHOICE BELOW****************************
4
*************WHAT TYPE OF MOVE WOULD YOU LIKE TO DO******************
Enter '1' to move last card onto the one on its left
Enter '2' to move last card onto the one three places to its left
Enter '3' to move a card in the middle onto the one three places to
its left
*************************ENTER BELOW*********************************
1
***********************Patience By JAC105*****************************
```

[Figure 5]

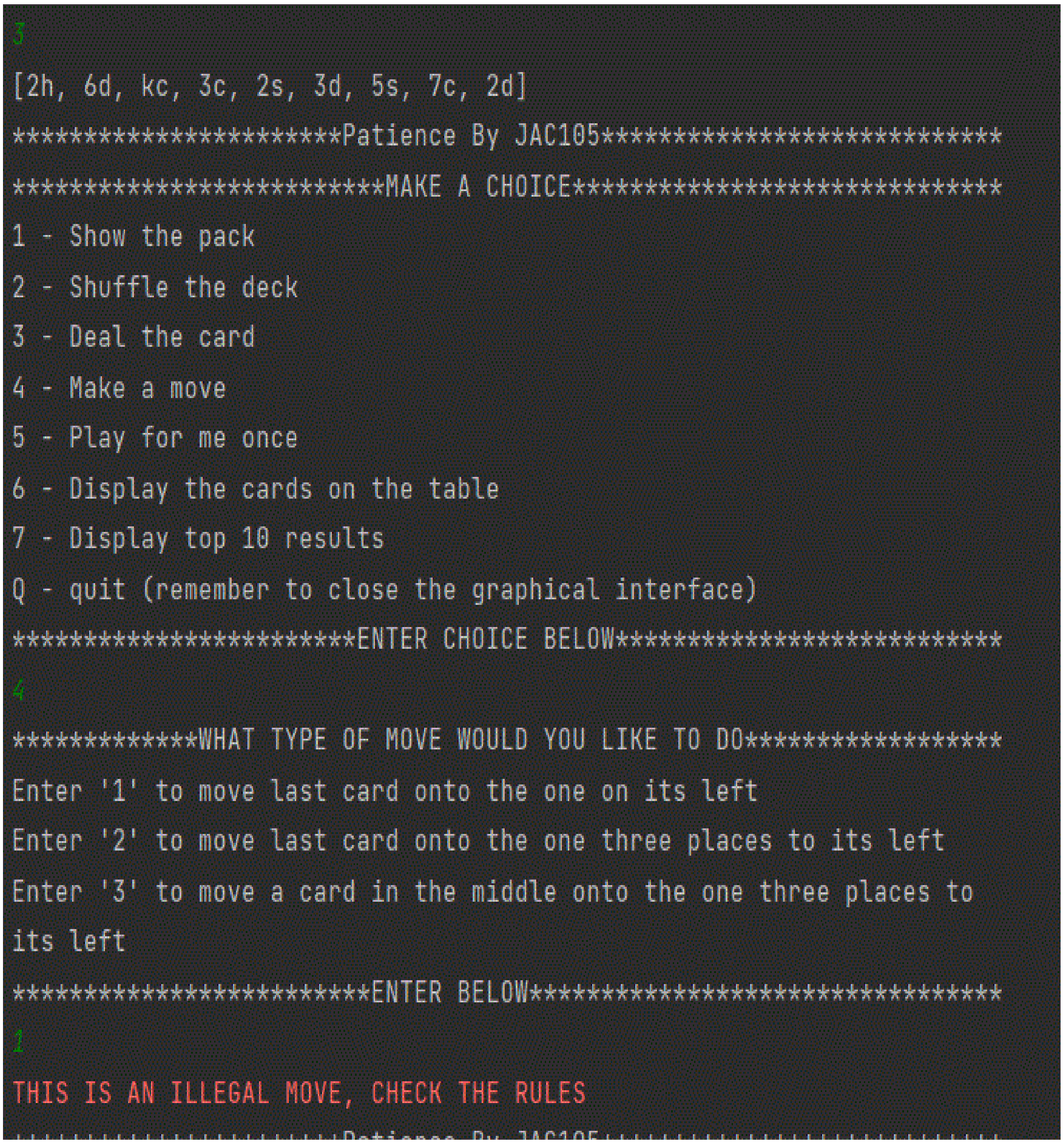
Showing the makeMove function working correctly



```
*************************ENTER BELOW*********************************
1
**********************Patience By JAC105***************************
*************************MAKE A CHOICE*******************************
1 - Show the pack
2 - Shuffle the deck
3 - Deal the card
4 - Make a move
5 - Play for me once
6 - Display the cards on the table
7 - Display top 10 results
Q - quit (remember to close the graphical interface)
***********************ENTER CHOICE BELOW***************************
6
[2h, 6d, kc, 3c, 2s, 3d, 5s, 7c, 2d, 3h, jc]
**********************Patience By JAC105***************************
*************************MAKE A CHOICE*******************************
```

[Figure 6]

Showing cardsOnTable after the move working successfully

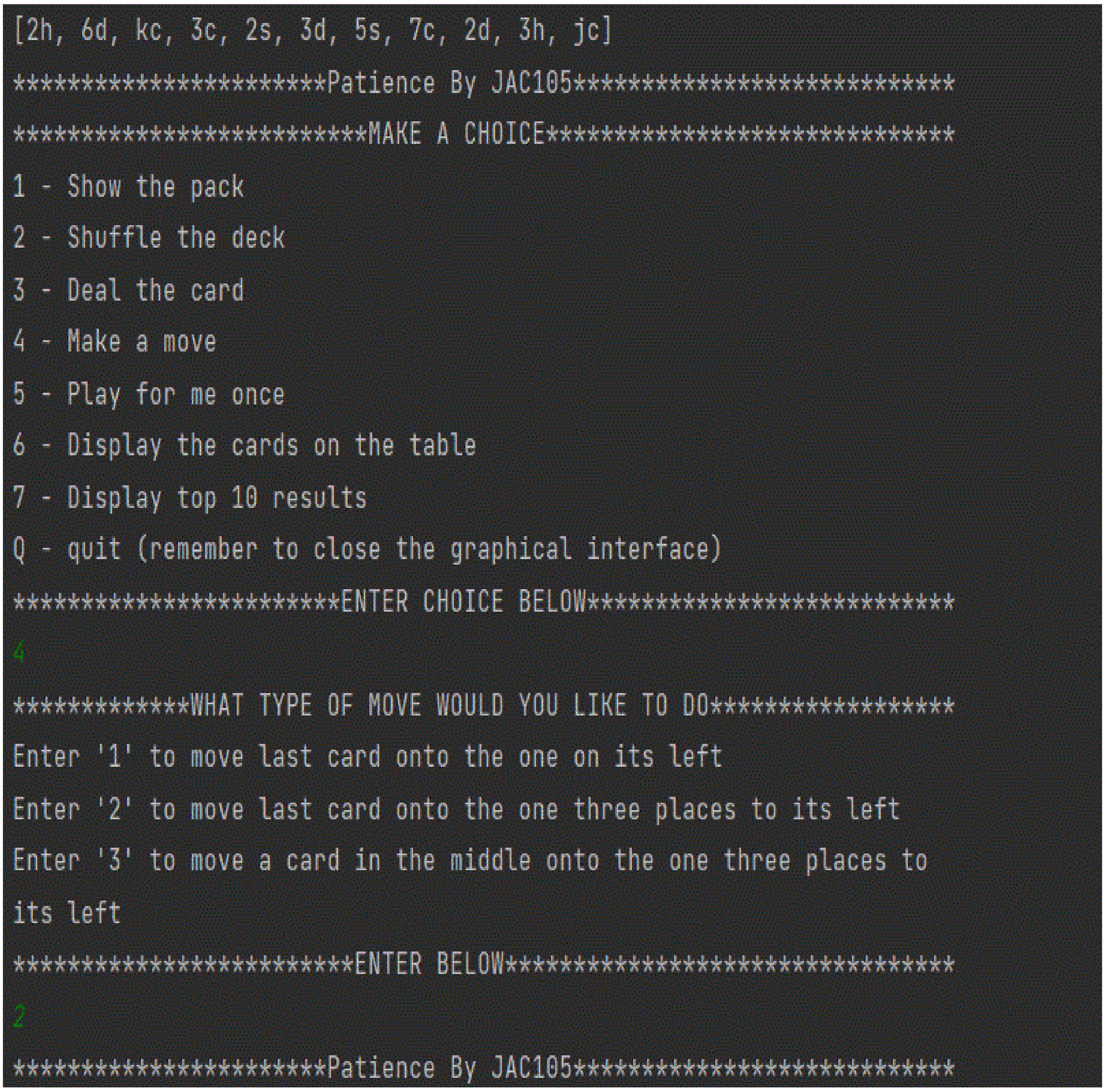


```
3
[2h, 6d, kc, 3c, 2s, 3d, 5s, 7c, 2d]
***********************Patience By JAC105****************************
**************************MAKE A CHOICE******************************
1 - Show the pack
2 - Shuffle the deck
3 - Deal the card
4 - Make a move
5 - Play for me once
6 - Display the cards on the table
7 - Display top 10 results
Q - quit (remember to close the graphical interface)
************************ENTER CHOICE BELOW***************************
4
*************WHAT TYPE OF MOVE WOULD YOU LIKE TO DO******************
Enter '1' to move last card onto the one on its left
Enter '2' to move last card onto the one three places to its left
Enter '3' to move a card in the middle onto the one three places to
its left
*************************ENTER BELOW*********************************
1
THIS IS AN ILLEGAL MOVE, CHECK THE RULES
***********************Patience By JAC105****************************
```

[Figure 7]

Showing what occurs when an illegal move is made on moveLast1 working properly

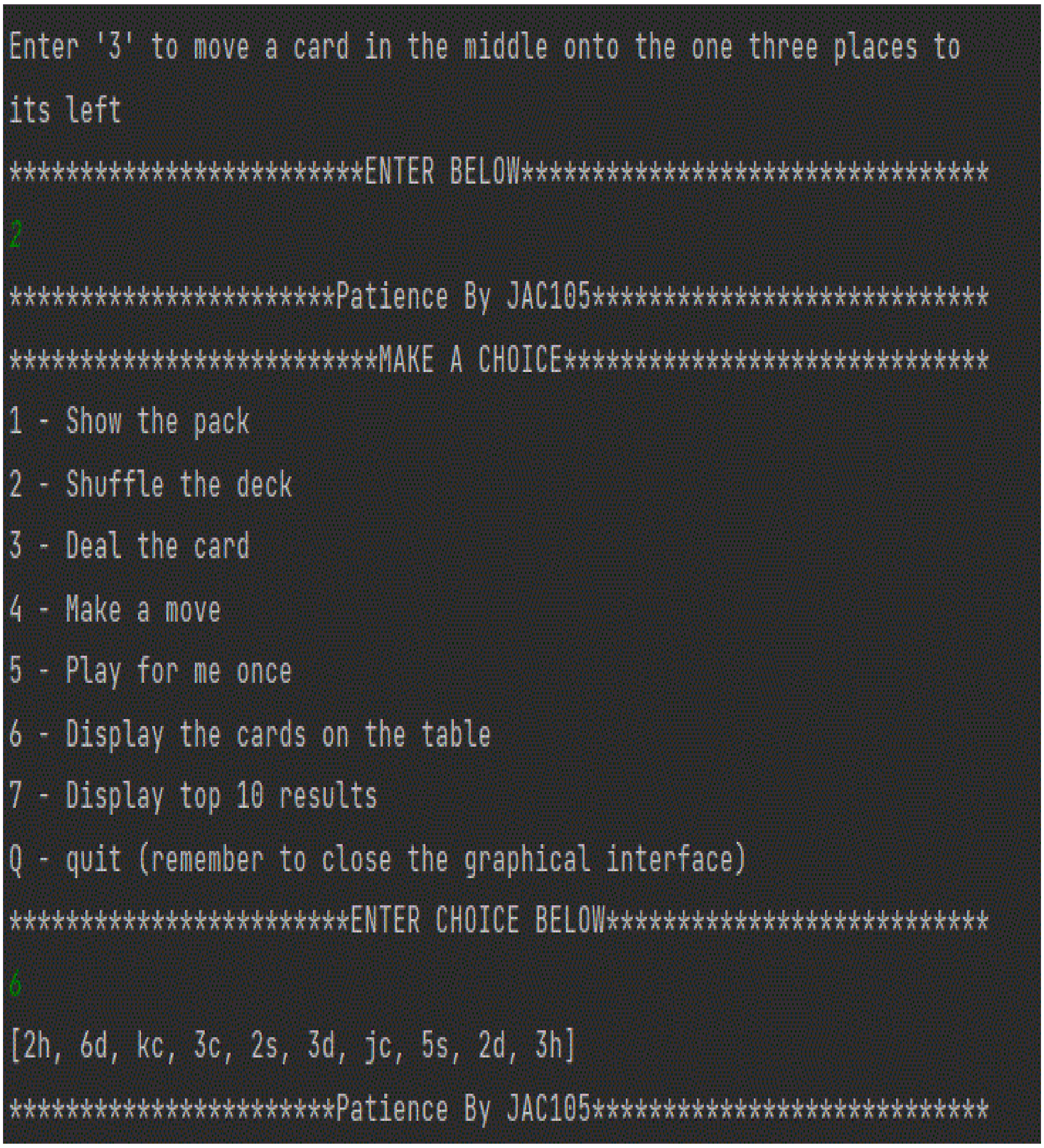
FR5



```
[2h, 6d, kc, 3c, 2s, 3d, 5s, 7c, 2d, 3h, jc]
***********************Patience By JAC105****************************
**************************MAKE A CHOICE*****************************
1 - Show the pack
2 - Shuffle the deck
3 - Deal the card
4 - Make a move
5 - Play for me once
6 - Display the cards on the table
7 - Display top 10 results
Q - quit (remember to close the graphical interface)
************************ENTER CHOICE BELOW***************************
4
*************WHAT TYPE OF MOVE WOULD YOU LIKE TO DO******************
Enter '1' to move last card onto the one on its left
Enter '2' to move last card onto the one three places to its left
Enter '3' to move a card in the middle onto the one three places to
its left
*************************ENTER BELOW*********************************
2
***********************Patience By JAC105****************************
```

[Figure 8]

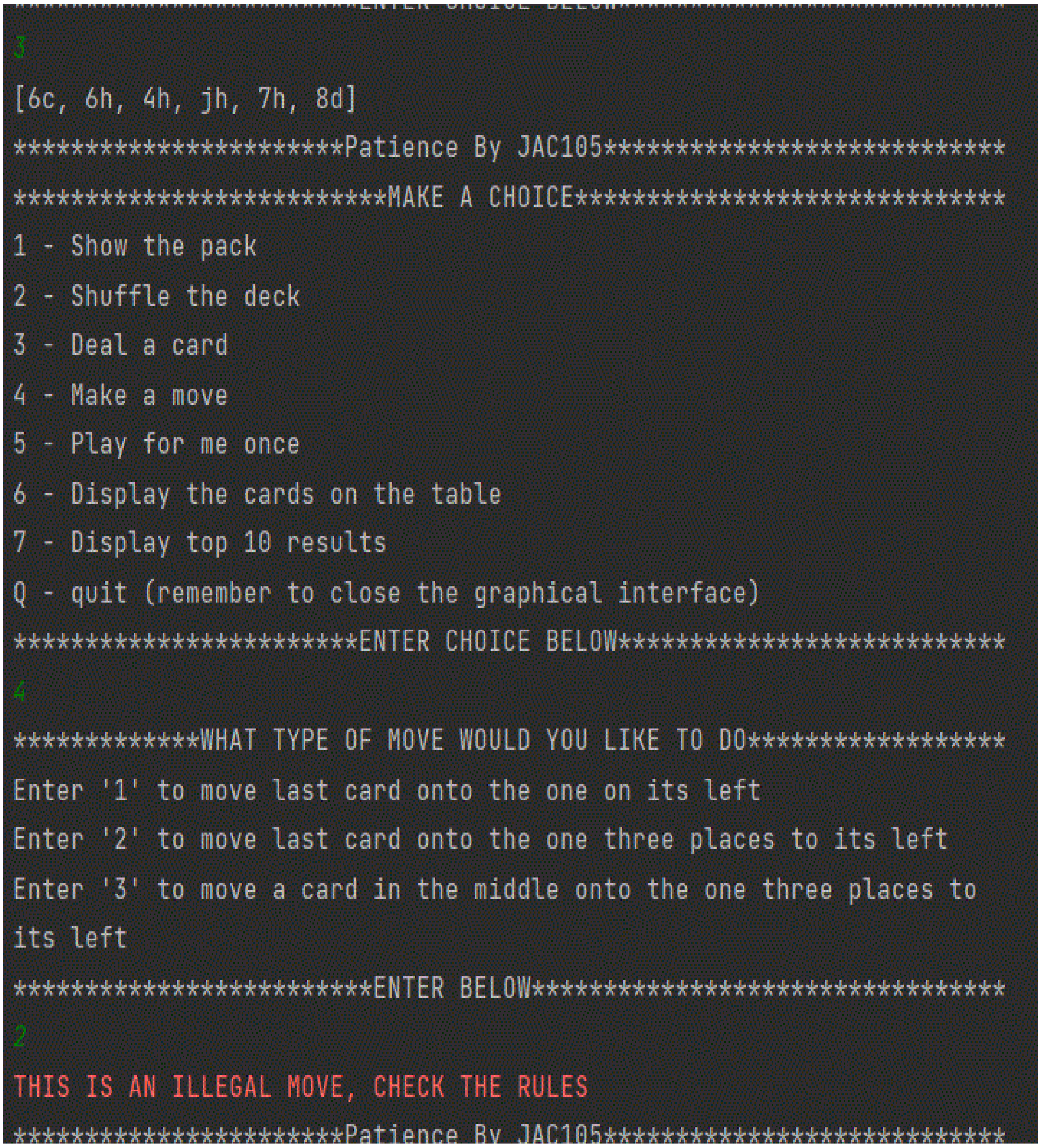
Showing the cards on deck and that option 2 on make a move works



```
Enter '3' to move a card in the middle onto the one three places to
its left
************************ENTER BELOW********************************
2
**********************Patience By JAC105***************************
*************************MAKE A CHOICE*****************************
1 - Show the pack
2 - Shuffle the deck
3 - Deal the card
4 - Make a move
5 - Play for me once
6 - Display the cards on the table
7 - Display top 10 results
Q - quit (remember to close the graphical interface)
***********************ENTER CHOICE BELOW***************************
6
[2h, 6d, kc, 3c, 2s, 3d, jc, 5s, 2d, 3h]
**********************Patience By JAC105***************************
```

[Figure 9]

Showing the result of moveLast3 being a success

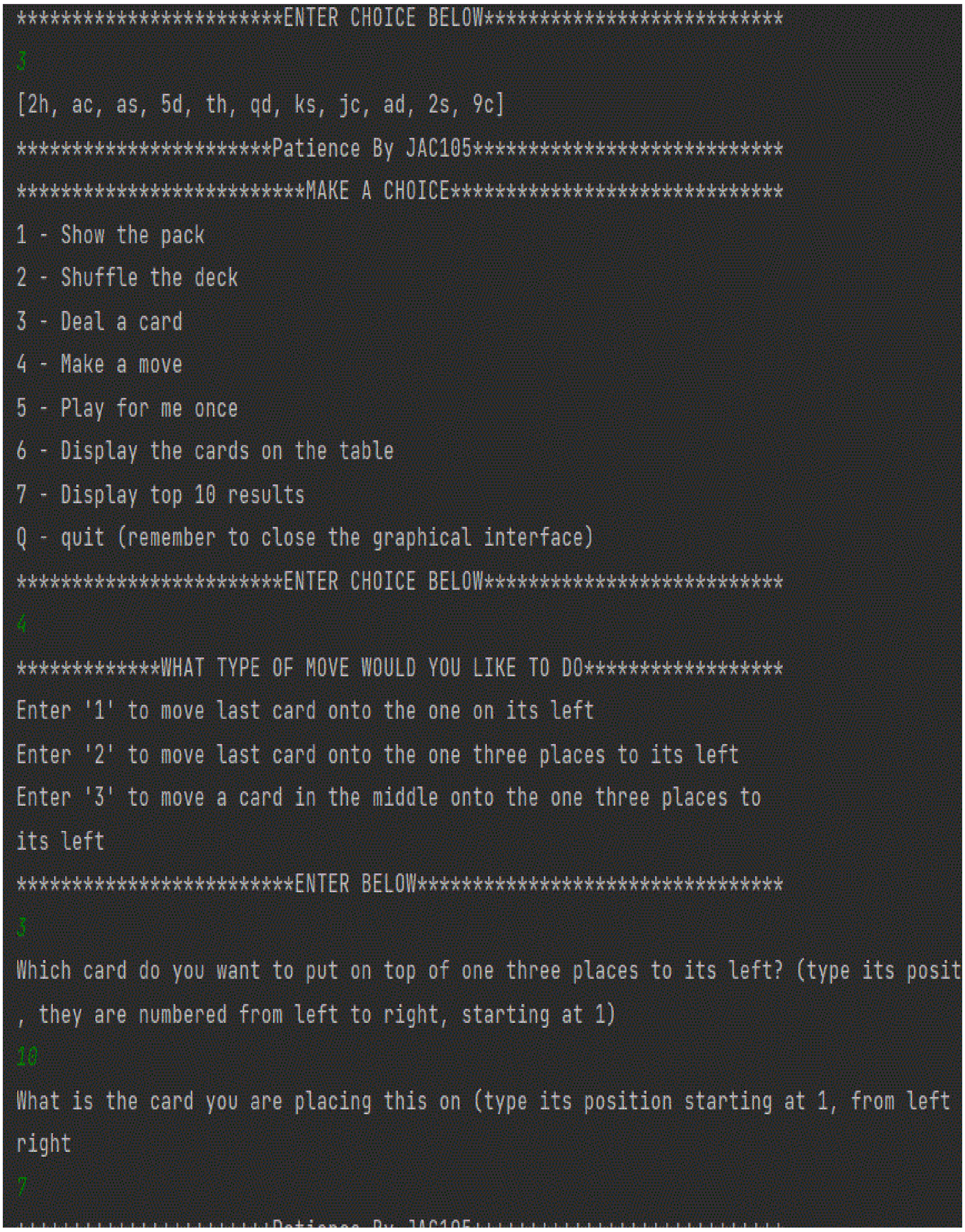


```
************************ENTER CHOICE BELOW***************************
3
[6c, 6h, 4h, jh, 7h, 8d]
***********************Patience By JAC105****************************
**************************MAKE A CHOICE******************************
1 - Show the pack
2 - Shuffle the deck
3 - Deal a card
4 - Make a move
5 - Play for me once
6 - Display the cards on the table
7 - Display top 10 results
Q - quit (remember to close the graphical interface)
************************ENTER CHOICE BELOW***************************
4
*************WHAT TYPE OF MOVE WOULD YOU LIKE TO DO*****************
Enter '1' to move last card onto the one on its left
Enter '2' to move last card onto the one three places to its left
Enter '3' to move a card in the middle onto the one three places to
its left
*************************ENTER BELOW*********************************
2
THIS IS AN ILLEGAL MOVE, CHECK THE RULES
***********************Patience By JAC105****************************
```

[Figure 10]

Showing moveLast3 else statement working

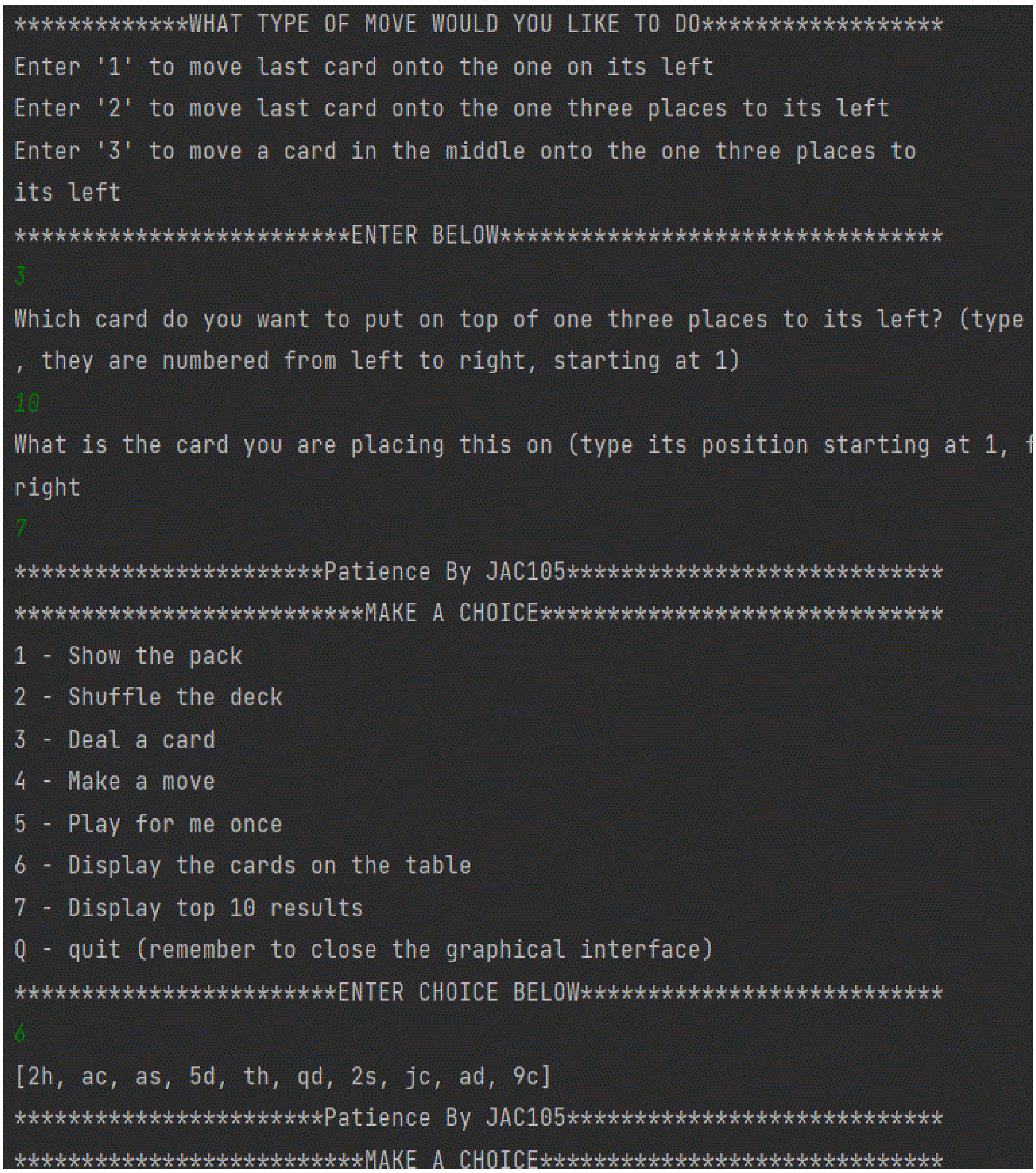
FR6



```
************************ENTER CHOICE BELOW***************************
3
[2h, ac, as, 5d, th, qd, ks, jc, ad, 2s, 9c]
***********************Patience By JAC105****************************
**************************MAKE A CHOICE******************************
1 - Show the pack
2 - Shuffle the deck
3 - Deal a card
4 - Make a move
5 - Play for me once
6 - Display the cards on the table
7 - Display top 10 results
Q - quit (remember to close the graphical interface)
************************ENTER CHOICE BELOW***************************
4
*************WHAT TYPE OF MOVE WOULD YOU LIKE TO DO******************
Enter '1' to move last card onto the one on its left
Enter '2' to move last card onto the one three places to its left
Enter '3' to move a card in the middle onto the one three places to
its left
*************************ENTER BELOW*********************************
3
Which card do you want to put on top of one three places to its left? (type its posit
, they are numbered from left to right, starting at 1)
10
What is the card you are placing this on (type its position starting at 1, from left
right
7
```

[Figure 11]

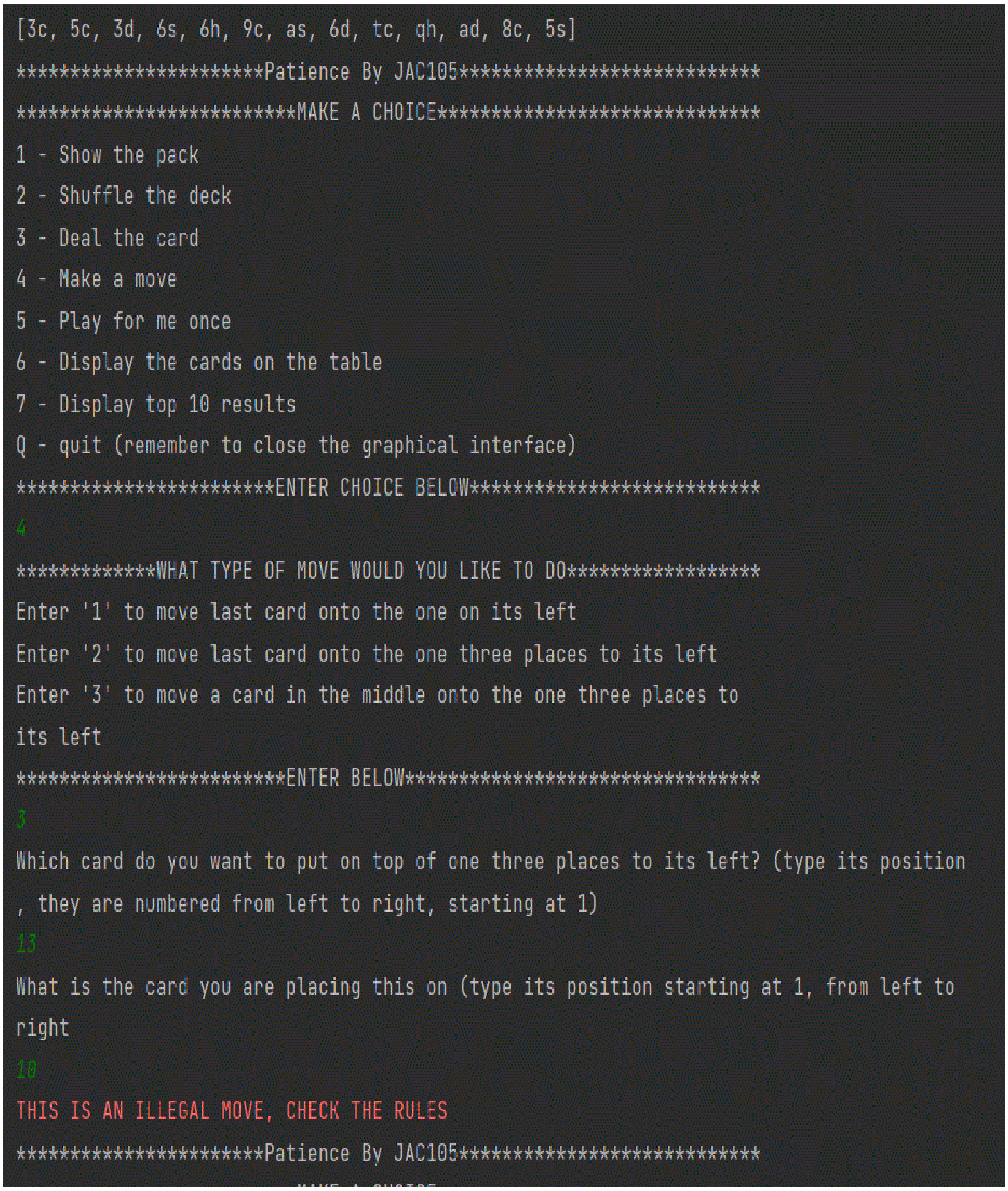
showing makeMove() option 3 working



```
*************WHAT TYPE OF MOVE WOULD YOU LIKE TO DO*****************
Enter '1' to move last card onto the one on its left
Enter '2' to move last card onto the one three places to its left
Enter '3' to move a card in the middle onto the one three places to
its left
*************************ENTER BELOW*********************************
3
Which card do you want to put on top of one three places to its left? (type
, they are numbered from left to right, starting at 1)
10
What is the card you are placing this on (type its position starting at 1, f
right
7
***********************Patience By JAC105****************************
**************************MAKE A CHOICE******************************
1 - Show the pack
2 - Shuffle the deck
3 - Deal a card
4 - Make a move
5 - Play for me once
6 - Display the cards on the table
7 - Display top 10 results
Q - quit (remember to close the graphical interface)
************************ENTER CHOICE BELOW***************************
6
[2h, ac, as, 5d, th, qd, 2s, jc, ad, 9c]
***********************Patience By JAC105****************************
**************************MAKE A CHOICE******************************
```

[Figure 12]

Showing the amalgamation working

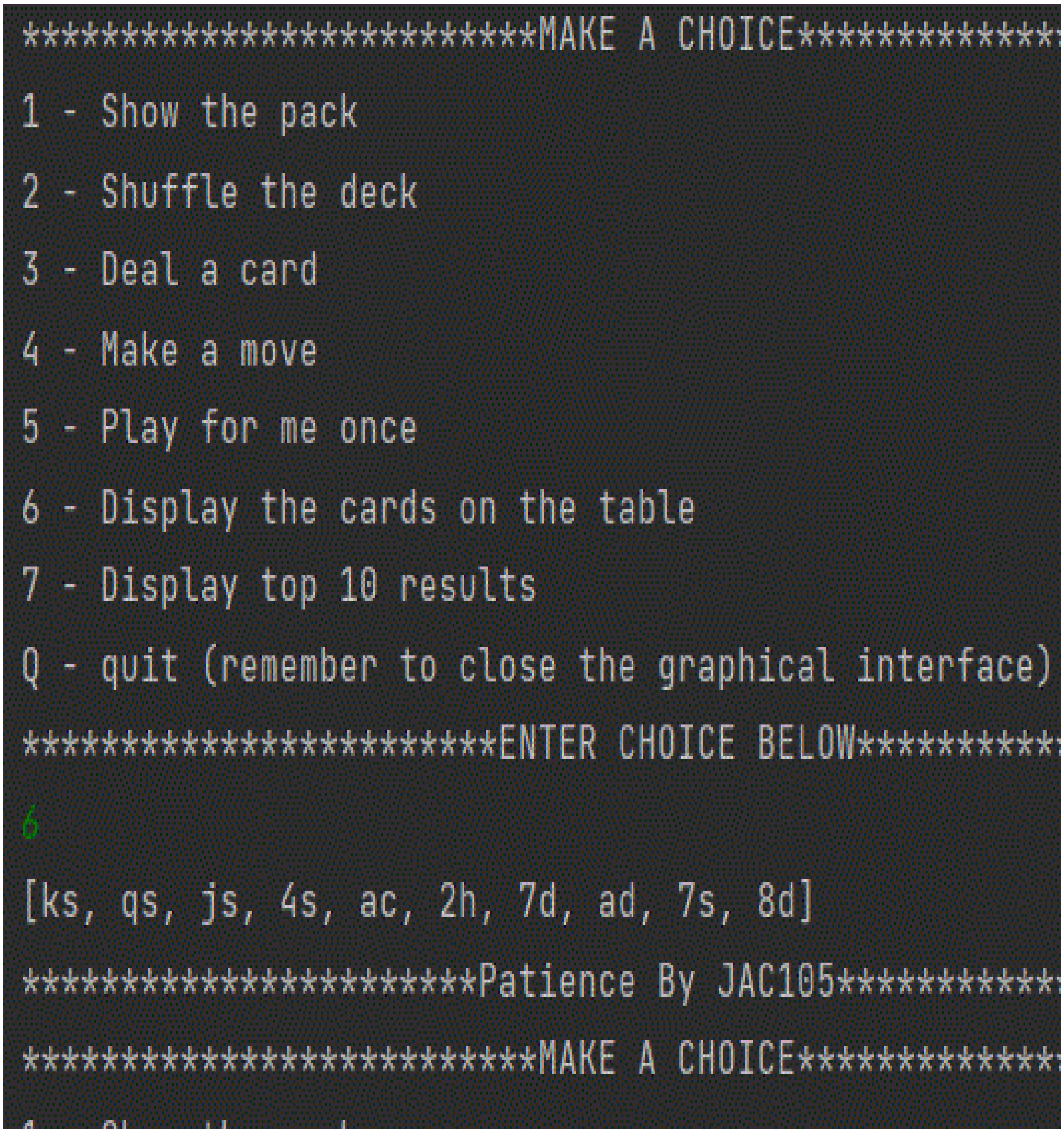


```
[3c, 5c, 3d, 6s, 6h, 9c, as, 6d, tc, qh, ad, 8c, 5s]
***********************Patience By JAC105****************************
**************************MAKE A CHOICE******************************
1 - Show the pack
2 - Shuffle the deck
3 - Deal the card
4 - Make a move
5 - Play for me once
6 - Display the cards on the table
7 - Display top 10 results
Q - quit (remember to close the graphical interface)
************************ENTER CHOICE BELOW***************************
4
************WHAT TYPE OF MOVE WOULD YOU LIKE TO DO*****************
Enter '1' to move last card onto the one on its left
Enter '2' to move last card onto the one three places to its left
Enter '3' to move a card in the middle onto the one three places to
its left
************************ENTER BELOW*********************************
3
Which card do you want to put on top of one three places to its left? (type its position
, they are numbered from left to right, starting at 1)
13
What is the card you are placing this on (type its position starting at 1, from left to
right
10
THIS IS AN ILLEGAL MOVE, CHECK THE RULES
***********************Patience By JAC105****************************
```

[Figure 13]

Showing amalgamate methods else statement working

FR7

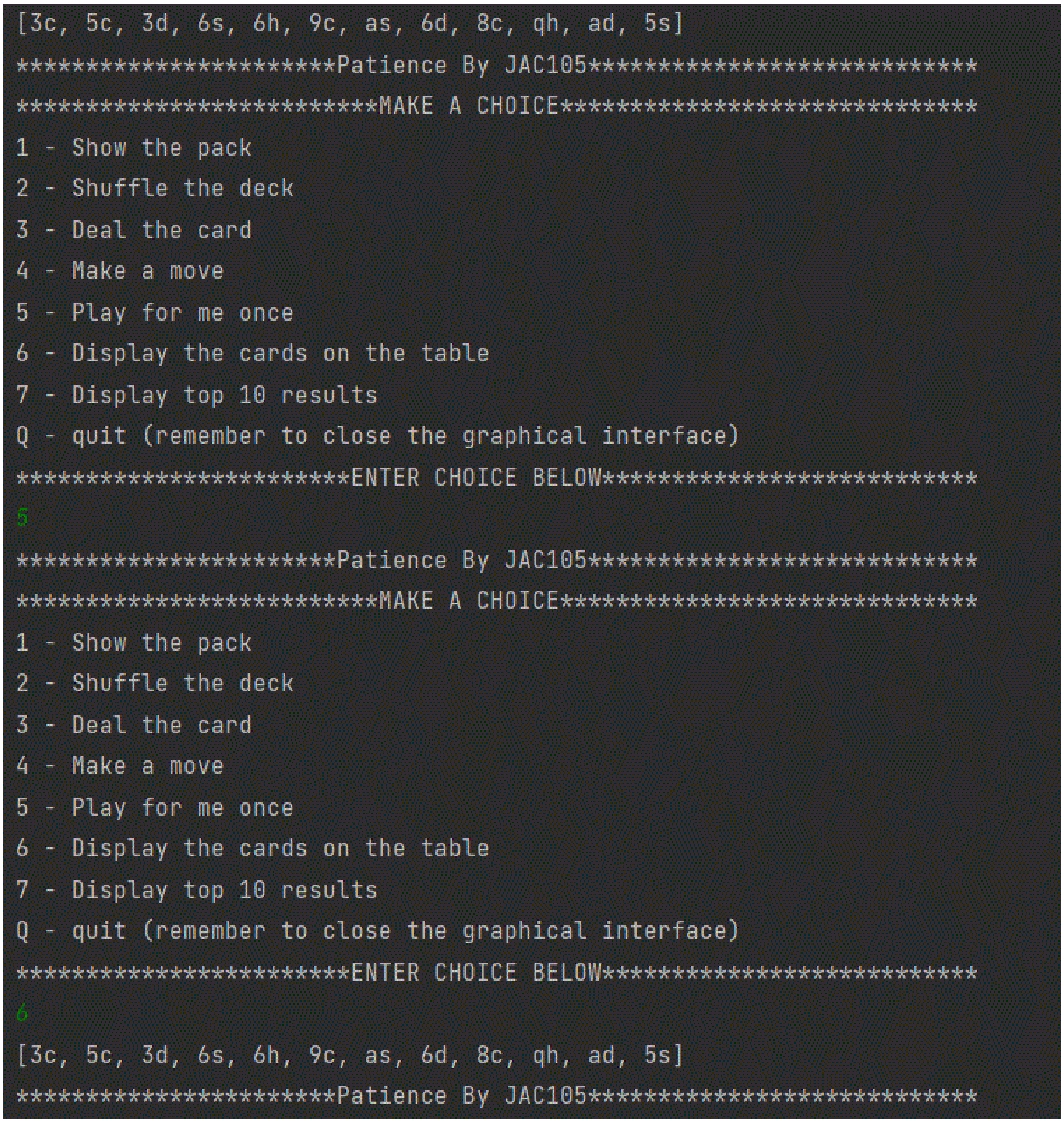


```
**************************MAKE A CHOICE*************
1 - Show the pack
2 - Shuffle the deck
3 - Deal a card
4 - Make a move
5 - Play for me once
6 - Display the cards on the table
7 - Display top 10 results
Q - quit (remember to close the graphical interface)
************************ENTER CHOICE BELOW**********
6
[ks, qs, js, 4s, ac, 2h, 7d, ad, 7s, 8d]
***********************Patience By JAC105***********
**************************MAKE A CHOICE*************
```

[Figure 14]

Showing the showThePack method working

FR8

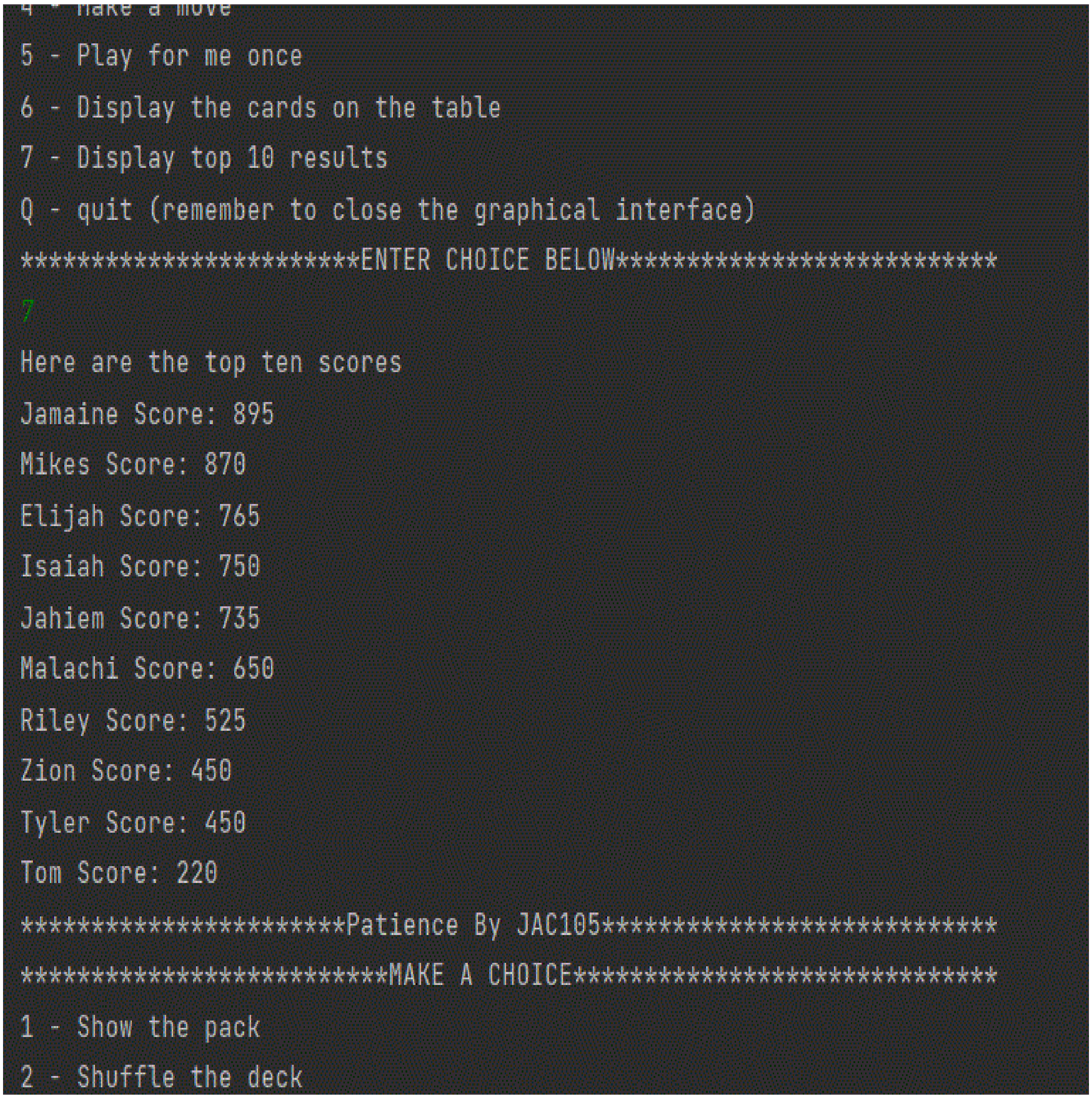


```
[3c, 5c, 3d, 6s, 6h, 9c, as, 6d, 8c, qh, ad, 5s]
***********************Patience By JAC105****************************
**************************MAKE A CHOICE******************************
1 - Show the pack
2 - Shuffle the deck
3 - Deal the card
4 - Make a move
5 - Play for me once
6 - Display the cards on the table
7 - Display top 10 results
Q - quit (remember to close the graphical interface)
************************ENTER CHOICE BELOW***************************
5
***********************Patience By JAC105****************************
**************************MAKE A CHOICE******************************
1 - Show the pack
2 - Shuffle the deck
3 - Deal the card
4 - Make a move
5 - Play for me once
6 - Display the cards on the table
7 - Display top 10 results
Q - quit (remember to close the graphical interface)
************************ENTER CHOICE BELOW***************************
6
[3c, 5c, 3d, 6s, 6h, 9c, as, 6d, 8c, qh, ad, 5s]
***********************Patience By JAC105****************************
```

[Figure 15]

showing playOnce running but not behaving how intended

FR9

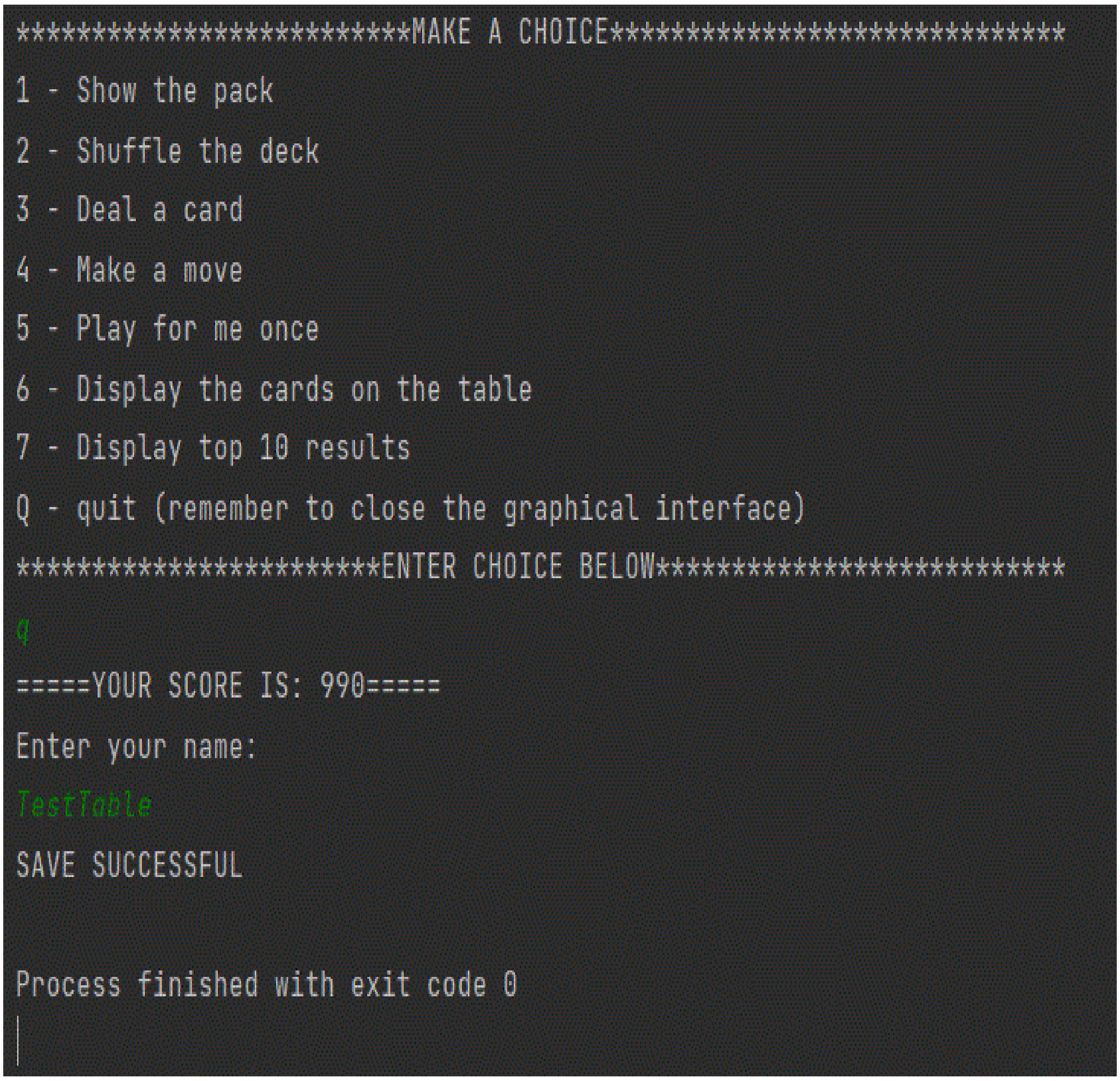


```
4 - Make a move
5 - Play for me once
6 - Display the cards on the table
7 - Display top 10 results
Q - quit (remember to close the graphical interface)
************************ENTER CHOICE BELOW***************************
7
Here are the top ten scores
Jamaine Score: 895
Mikes Score: 870
Elijah Score: 765
Isaiah Score: 750
Jahiem Score: 735
Malachi Score: 650
Riley Score: 525
Zion Score: 450
Tyler Score: 450
Tom Score: 220
***********************Patience By JAC105****************************
**************************MAKE A CHOICE******************************
1 - Show the pack
2 - Shuffle the deck
```

[Figure 16]

Showing the top ten scores

FR10

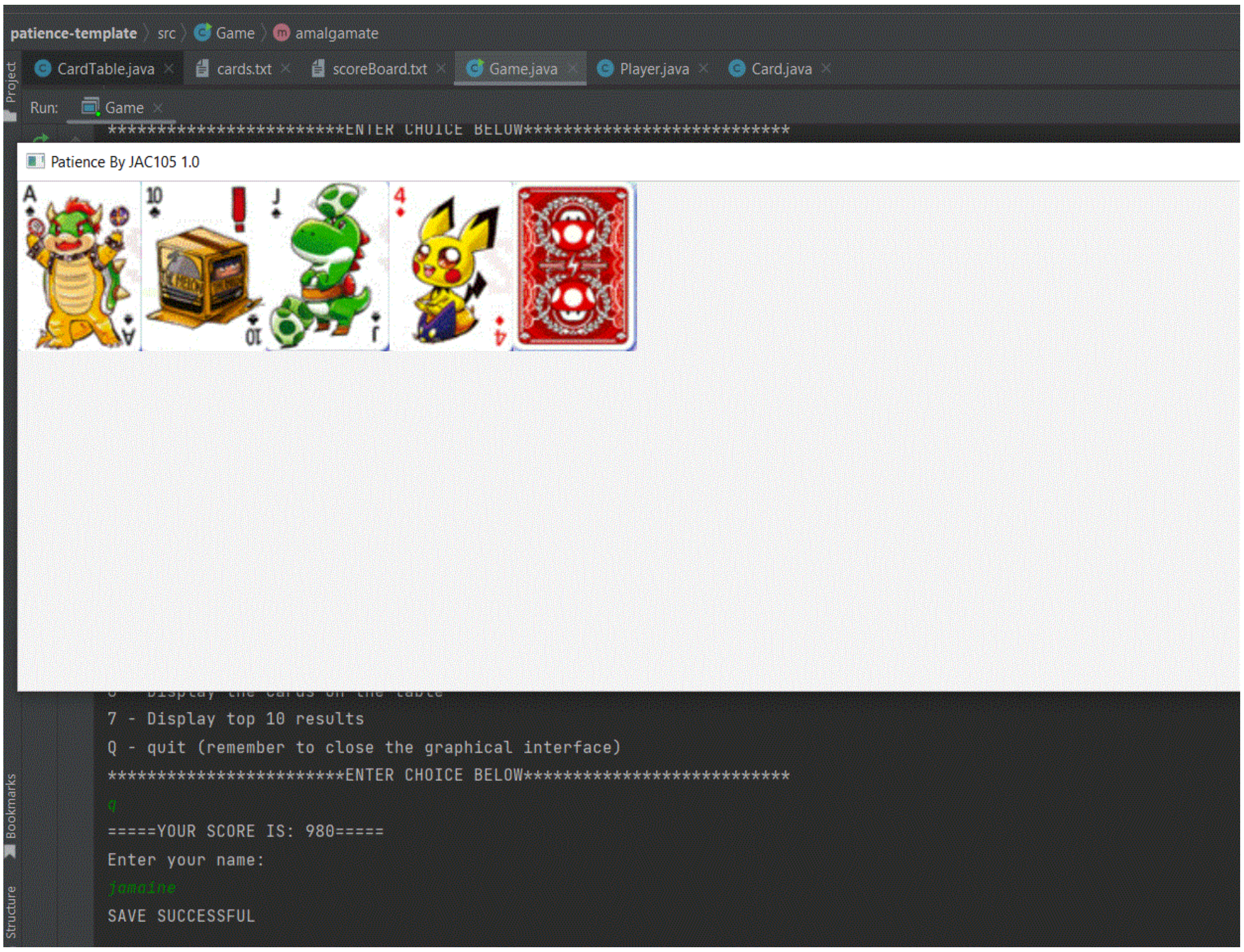


```
**************************MAKE A CHOICE******************************
1 - Show the pack
2 - Shuffle the deck
3 - Deal a card
4 - Make a move
5 - Play for me once
6 - Display the cards on the table
7 - Display top 10 results
Q - quit (remember to close the graphical interface)
************************ENTER CHOICE BELOW***************************
q
=====YOUR SCORE IS: 990=====
Enter your name:
TestTable
SAVE SUCCESSFUL

Process finished with exit code 0
```

[Figure 17]

Showing quit working

NFR2

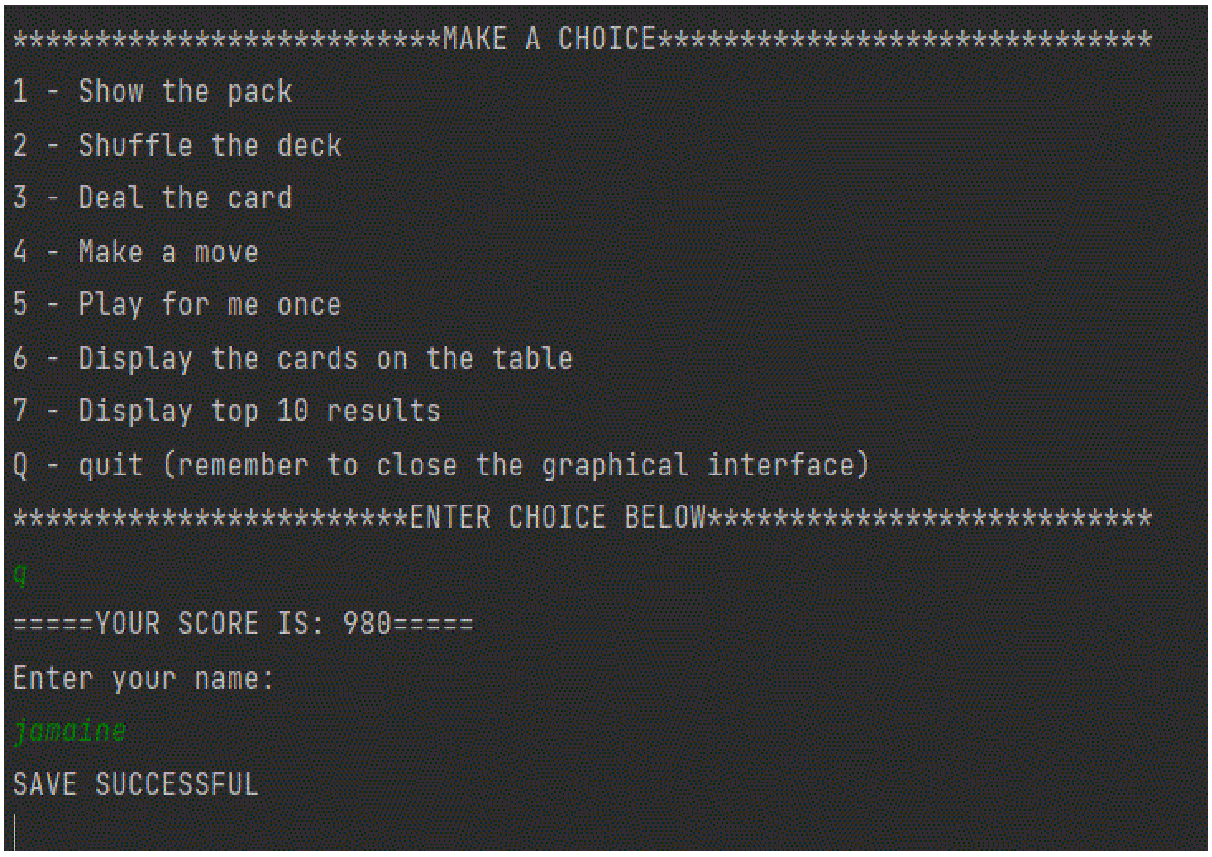


```
************************ENTER CHOICE BELOW***************************
6 - Display the cards on the table
7 - Display top 10 results
Q - quit (remember to close the graphical interface)
************************ENTER CHOICE BELOW***************************
q
=====YOUR SCORE IS: 980=====
Enter your name:
jamaine
SAVE SUCCESSFUL
```

[Figure 18]

Showing the cards displayed visually

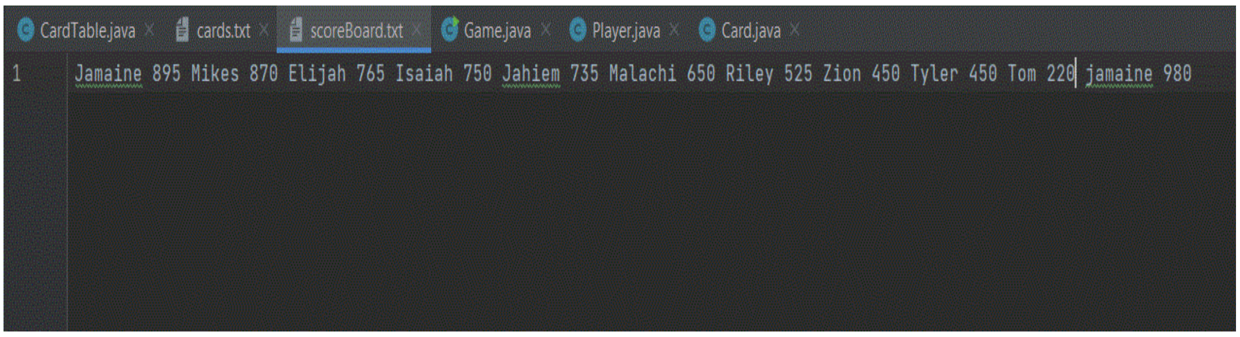
NFR3



```
**************************MAKE A CHOICE******************************
1 - Show the pack
2 - Shuffle the deck
3 - Deal the card
4 - Make a move
5 - Play for me once
6 - Display the cards on the table
7 - Display top 10 results
Q - quit (remember to close the graphical interface)
************************ENTER CHOICE BELOW***************************
q
=====YOUR SCORE IS: 980=====
Enter your name:
jamaine
SAVE SUCCESSFUL
```

[Figure 19]

Showing saveScoreBoard is running



```
Jamaine 895 Mikes 870 Elijah 765 Isaiah 750 Jahiem 735 Malachi 650 Riley 525 Zion 450 Tyler 450 Tom 220 jamaine 980
```

[Figure 20]

Showing saveScoreBoard is working

Evaluation

What was difficult?

The first issue was trying to get the GUI to work, I managed to get through this after installing javaFx on to my computer and adding the library to my project, however I did not manage to make the game playable with graphics. I also came across an error with reading my cards.txt file when I had the load function as static and in the Card class. Every item of the array was replaced with the most recent Card added. In, addition trying to get my amalgamate to function correctly was hard but after doing calculations on pen and paper I figured it out.

What remains to be done?

The playOnce and checkMoves methods need to be fixed so they output what was intended. I also need to implement the mechanics of playing the game graphically and create new classes for this and use more inheritance.

What have I learned?

I learned that having static methods can cause functionality problems, although the syntax is fine, if they are not used correctly. I also learned about casting objects although I removed its implementation because I was struggling with the inheritance of class Card from class Deck. I then removed deck and done everything in Card, this was before I noticed my issue with the static methods. I changed all the static methods to non-static and moved these methods to the class Game.

What mark am I expecting?

I am expecting to achieve around 75% because I believe I have done the documentation very well and will score high marks on that section. I also managed to do most of the required tasks for the implementation and design but also spent the time on my user interface so it looks more professional and changed the gif files of the cards to better looking ones, therefore, I believe I deserve marks for flair.