

Main Assignment: Patience – A Card Game

CS12320 – OOP

Muntazir Rashid – mrm19

Contents:

● <u>Introduction</u>	3
● <u>Design</u>	6
● <u>Testing</u>	8
● <u>Evaluation</u>	14

Introduction

Problem Statement:

My friend Chris developed a form of patience (solitaire) card game that I would like to be able to play on my computer. It is very simple, runs from command line but has a basic graphical interface.

This is how it works:

- The object of the game is to end up with one pile/stack of cards.
- Shuffle a deck of cards. • Lay down the first card face up.
- Lay down the next one next to it, and so on.
- This would lead you to 52 cards all facing upwards, except that you can put one pile on another if any of the following apply:
 - They are next to each other and the same number or suit.
 - There are two piles between them, and they are the same number or suit.

Game:

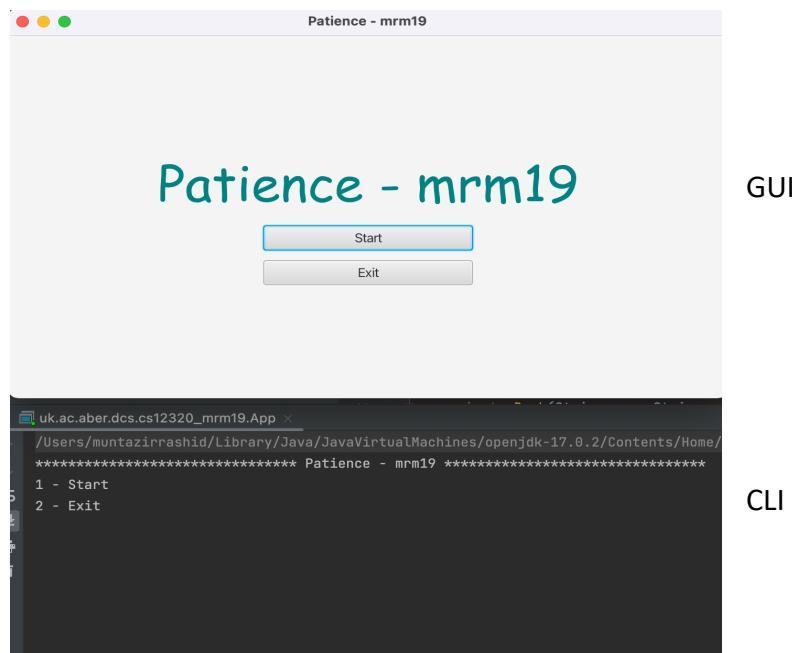
The game can be played using either the Graphical User Interface (GUI) or the Command Line Interface (CLI). CLI lists down a menu of different commands by which you can play the game, and GUI is the graphical representation of the game and is fully responsive to mouse clicks.

There is also a Rule/Help Section (Pop-up) in the GUI of the game which reminds the player of the rules and also has a button there which automatically makes a move for the user.

When the user is done playing the game, an 'Exit' button appears on the screen, which can be used to close the game. An exit option is also present in the CLI menu of the game.

Before the actual game, there is a menu which has two options: Start and Exit, by which you can start or exit the game. Both of these options are present on the CLI as well as the GUI of the game.

Start Menu:

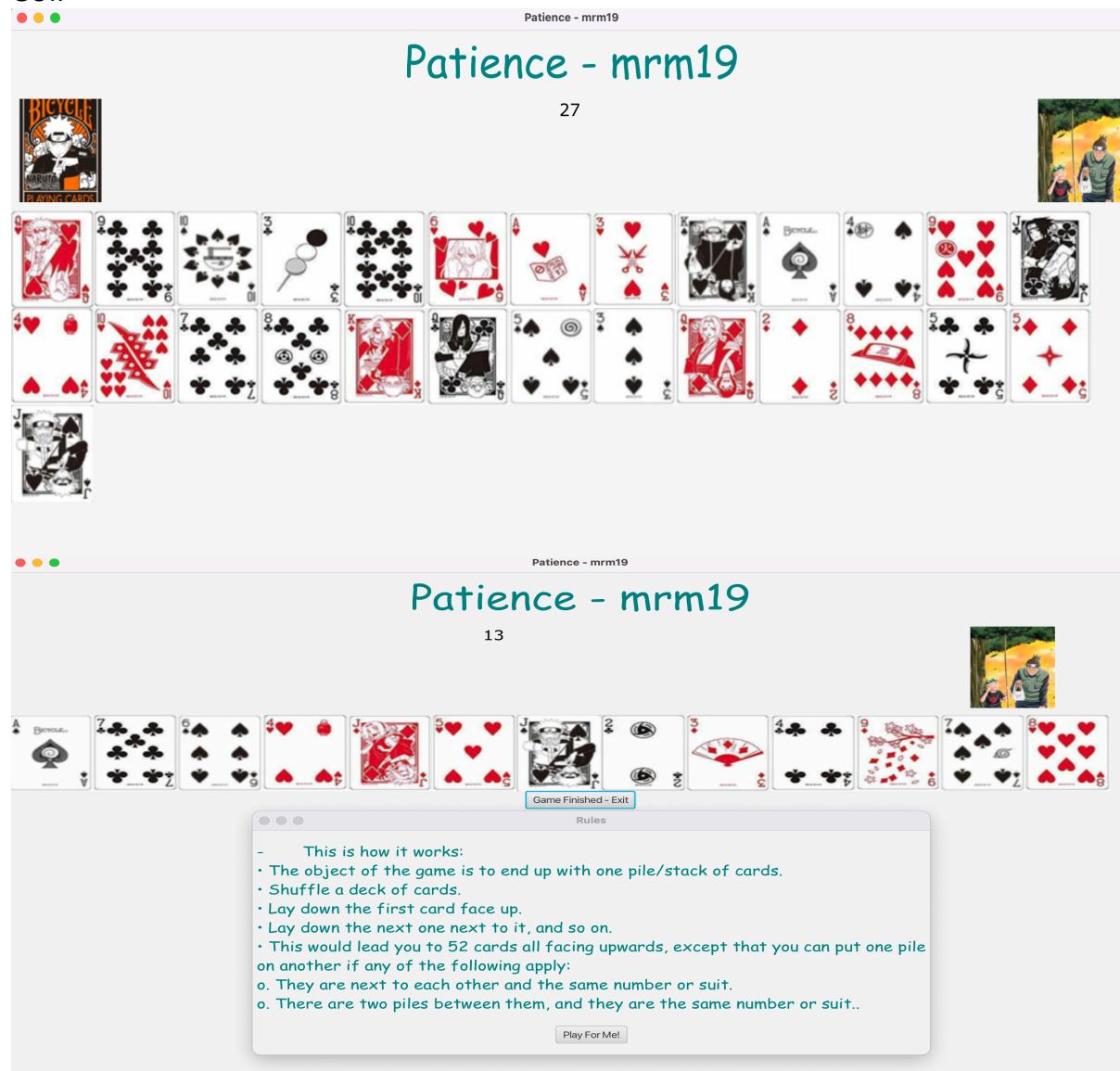


Game:

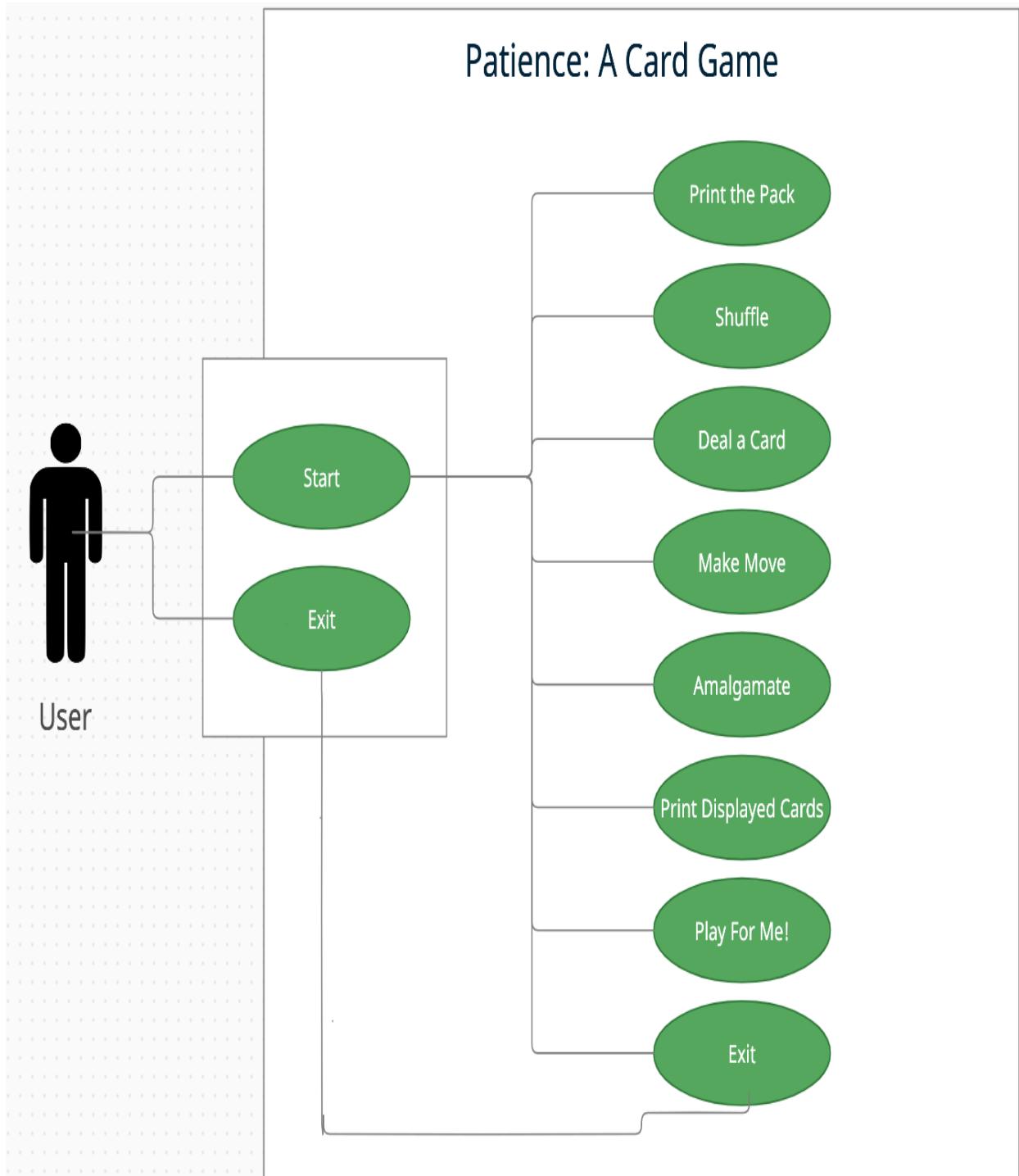
CLI:

```
***** Patience - mrm19 *****
1 - Start
2 - Exit
1
Shuffling completed
8 - Play For Me!
1 - Print the pack out (this is so you can check that it plays properly)
2 - Shuffle
3 - Deal a card
4 - Make a move, move last pile onto previous one
5 - Make a move, move last pile back over two piles
6 - Amalgamate piles
7 - Print the displayed Cards on the command line
9 - Exit
```

GUI:

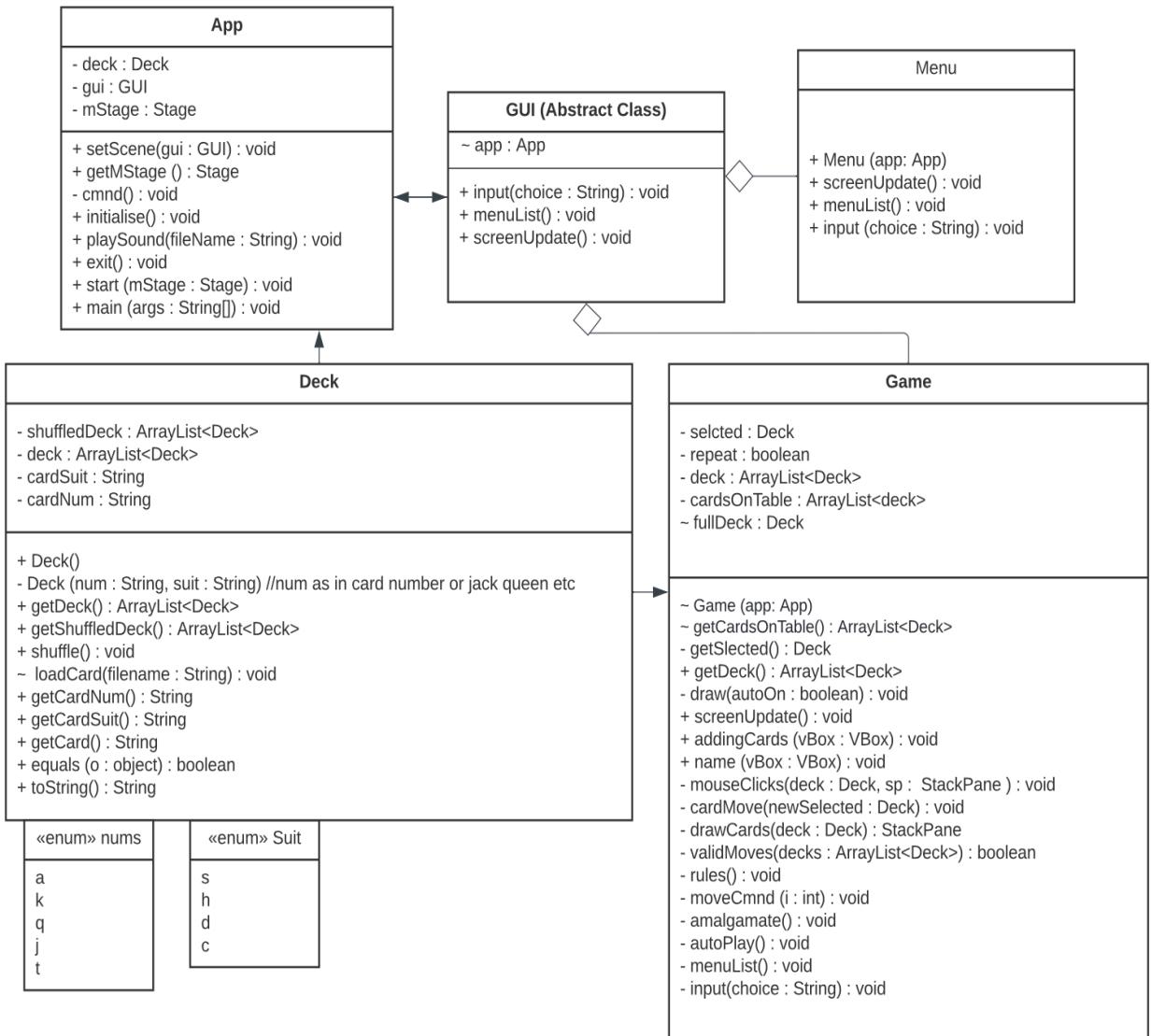


Use Case Diagram:



Design

Class Diagram:



Class Description:

1. The "App class" includes the *main* method and so serves as the application's starting point. It also extends the "Application class" from "JavaFx" and handles layouts and initialization of components within itself. It also includes a "playSound" method I created to add sound effects within the game, it works (kind of) and doesn't give any errors but sound is not there.
2. "nums" (Enum) contains the non-numerical values of the card like q for Queen. "nums" and "Suit" are looped through in "Deck" class to create and add the cards to deck.

3. The “Deck” class represents the collection of cards. The “shuffle” method in the “Deck” class is used to shuffle the cards.
4. The “GUI” (Graphical User Interface) is an abstract class which acts as the super class for all other classes responsible for the graphical interface of the game. Abstract methods like “menuList”, “screenUpdate” etc are declared inside of this abstract class which are later implemented by “children” classes according to their need.
5. The “Menu” class extends the “GUI” and uses “menuList” methods to create the Start menu (CLI) and also the Graphical interface of the start menu using “screenUpdate” method. Start menu has 2 options “Start” and “Exit” which on click (in GUI clicking the button triggers the respective methods or in CLI “input” method is used to get input from the user to select the required option) start the game or close the application respectively.
6. The “Game” class extends the GUI and is responsible for all the major functionalities of the game including the CLI menu and graphical interface.
It uses the “menuList” to print out all the commands like printing out the deck, shuffle etc.
It uses the “draw” and “move” methods to draw a card out of the deck and make a move respectively. The “autoPlay” method can be used to automate the game. The “rules” method is another graphical interface which labels out all the rules of game which is triggered when the Image on screen which has an event listener (mouse click) on it is clicked. It shows up a screen with all the rules and a button for to make use of “autoPlay” method (Button saying “Play for me!” which has an event listener on for click).

I didn't get time to implement a save and load method to save the score of users and to load it when asked for in the game. I was working on the graphics of the game and also had to write the documentation. That's why I left it out and started working on the documentation as I was running out of time but I'll keep working on it and add the save and load methods.

Amalgamation - Pseudo-code:

```

Take in 2 inputs of integer type.
//card u want to move and on top of u want to move position starting from 1
If (2nd Input >0)
  Right card = 1st input;
  Left card = 2nd input;
  If (right card suit == left card suit or right card value == left card value)
    Remove 2nd input card from table;
    Add 1st input card in place of 2nd input card;
  Else
    Print invalid move for both if statements;

```

Testing

The assignment brief requested that we provide test tables as well as evidence that the application is working. To address this need, I created a table that checks for and logs any mistakes that may occur during the application. I have also submitted screen shots and written content to demonstrate that my application satisfies the criteria specified in the project brief.

Testing Table:

ID	Requirement	Description	Inputs	Expected Output	Pass / Fail	Comments
1	NFR 1	CLI Menu	Normal Input: 1..9	Does the required Thing	P	Done 2 Menus: Both Work Fine
			Abnormal Input: Any other Number / character	Some Times Prints 8 (AutoPlay) Number On Menu First this does not affect the functionality at all.		
2	FR1	Show the Pack	1 On Menu	Shows the Pack	P	Works Fine
3	FR3	Deal a Card	3 On Menu	Deals a card	P	Works Fine
4	FR4 FR5	Make Move	4 and 5 on Menu	Makes a move	P	Works Fine
5	FR6	Amalgamate	6 on Menu	Amalgamate	P	Works Fine
6	FR2	Shuffle the cards	2 on Menu	Shuffle	P	Works Fine
7	FR7	Show all Displayed Cards	7 on Menu	Print Cards on Table	P	Works Fine
8	FR8	Play For Me Once	8 on Menu	auto play	P	Works Fine
9	FR9	Show the top ten results	n/a	Show top 10 scores	F	Didn't Implement
10	FR10	Quit	9 on Menu	Ends the Game	P	Works Fine
11	NFR2	GUI	Graphics Implemented	Show cards in GUI	P	Shows Cards in GUI and can even Play in GUI.

						Works Excellent
12	NFR3	Save Score in a File	n/a	Saves the game score	F	Haven't Implemented

Screenshots:

1 – Prints the Pack Out.

```

1
[7s, tc, 9s, 8c, 6s, 6h, ts, ks, as, ad, 9h, 6c, ah, jh, kc, 7h, 2c, 2h, 4d, 2d, 4c, 3c, 4h, 9d, 3
1 - Print the pack out (this is so you can check that it plays properly)
2 - Shuffle
3 - Deal a card
4 - Make a move, move last pile onto previous one
5 - Make a move, move last pile back over two piles
6 - Amalgamate piles
7 - Print the displayed Cards on the command line
9 - Exit
8 - Play For Me!
|
```

2 – Shuffle (Deck In Different Order Now).

```

2
8 - Play For Me!
1 - Print the pack out (this is so you can check that it plays properly)
2 - Shuffle
3 - Deal a card
4 - Make a move, move last pile onto previous one
5 - Make a move, move last pile back over two piles
6 - Amalgamate piles
7 - Print the displayed Cards on the command line
9 - Exit
1
[2s, 9h, 7s, 8h, 7d, 8d, 7h, 4h, ks, qc, 2c, 9d, 6h, 4d, 5d, 4s, jh, 6c, 9c, kd, 8c, 3h, 2h, kc, js, 3s, 5s, kh,
1 - Print the pack out (this is so you can check that it plays properly)
2 - Shuffle
3 - Deal a card
4 - Make a move, move last pile onto previous one
5 - Make a move, move last pile back over two piles
6 - Amalgamate piles
```

3 – Deals the Cards Out.

```

3
8 - Play For Me!
1 - Print the pack out (this is so you can check that it plays properly)
2 - Shuffle
3 - Deal a card
4 - Make a move, move last pile onto previous one
5 - Make a move, move last pile back over two piles
6 - Amalgamate piles
7 - Print the displayed Cards on the command line
9 - Exit
3
1 - Print the pack out (this is so you can check that it plays properly)
2 - Shuffle
3 - Deal a card
4 - Make a move, move last pile onto previous one
5 - Make a move, move last pile back over two piles
6 - Amalgamate piles
7 - Print the displayed Cards on the command line
9 - Exit
8 - Play For Me!
```

4 – Make a Move

```
1 - Print the pack out (this is so you can check that it plays properly)
2 - Shuffle
3 - Deal a card
4 - Make a move, move last pile onto previous one
5 - Make a move, move last pile back over two piles
6 - Amalgamate piles
7 - Print the displayed Cards on the command line
9 - Exit
8 - Play For Me!
```

```
4
1 - Print the pack out (this is so you can check that it plays properly)
2 - Shuffle
3 - Deal a card
4 - Make a move, move last pile onto previous one
5 - Make a move, move last pile back over two piles
6 - Amalgamate piles
7 - Print the displayed Cards on the command line
9 - Exit
```

The Cards need to be of same suit or number

```
8 - Play For Me!
6 - Amalgamate piles
7 - Print the displayed Cards on the command line
9 - Exit
[3c, 2s, 9h, 7s, 8h, 7d, 8d]
```

```
1 - Print the pack out (this is so you can check that it plays properly)
2 - Shuffle
3 - Deal a card
4 - Make a move, move last pile onto previous one
5 - Make a move, move last pile back over two piles
6 - Amalgamate piles
7 - Print the displayed Cards on the command line
9 - Exit
```

```
4
1 - Print the pack out (this is so you can check that it plays properly)
2 - Shuffle
3 - Deal a card
4 - Make a move, move last pile onto previous one
5 - Make a move, move last pile back over two piles
6 - Amalgamate piles
7 - Print the displayed Cards on the command line
9 - Exit
8 - Play For Me!
```

```
7
[3c, 2s, 9h, 7s, 8h, 8d]
1 - Print the pack out (this is so you can check that it plays properly)
2 - Shuffle
3 - Deal a card
4 - Make a move, move last pile onto previous one
5 - Make a move, move last pile back over two piles
6 - Amalgamate piles
7 - Print the displayed Cards on the command line
```

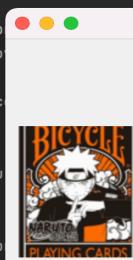
```
1 - Print the pack out (this is so you can check that it plays properly)
2 - Shuffle
3 - Deal a card
4 - Make a move, move last pile onto previous one
5 - Make a move, move last pile back over two piles
6 - Amalgamate piles
7 - Print the displayed Cards on the command line
9 - Exit
```

```
5
1 - Print the pack out (this is so you can check that it plays properly)
2 - Shuffle
3 - Deal a card
4 - Make a move, move last pile onto previous one
5 - Make a move, move last pile back over two piles
6 - Amalgamate piles
7 - Print the displayed Cards on the command line
9 - Exit
8 - Play For Me!
```

```
7
[3c, 7s, 8d, 4h, ks, qc, 2c, 9d, 6h, 4d, 5d, 4s, jh, 6c, 9c, kd, 8c, 3h, 2h, kc, js, 3s, 5s, kh, 2d, 8s, ac, 5h, 3d, as, 5c, th]
1 - Print the pack out (this is so you can check that it plays properly)
2 - Shuffle
3 - Deal a card
```

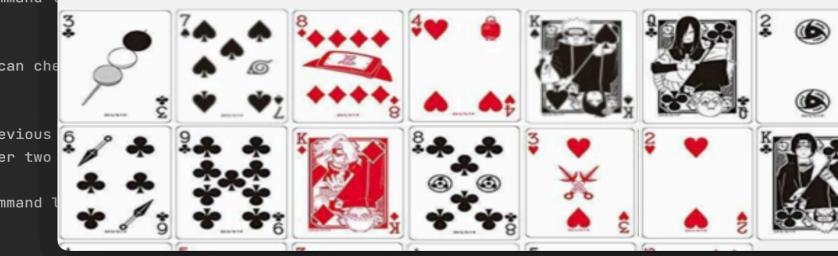
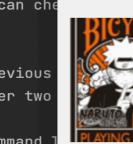


Patience



Patience

6

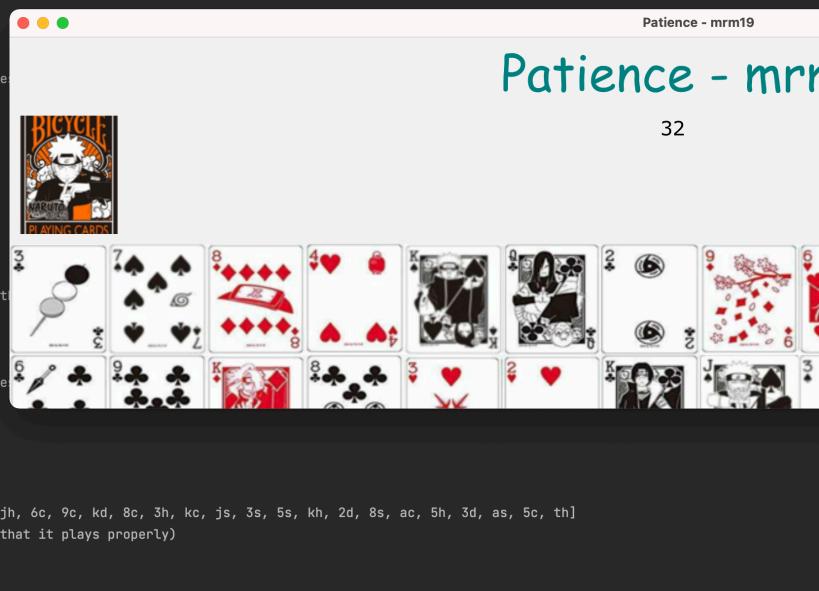


3

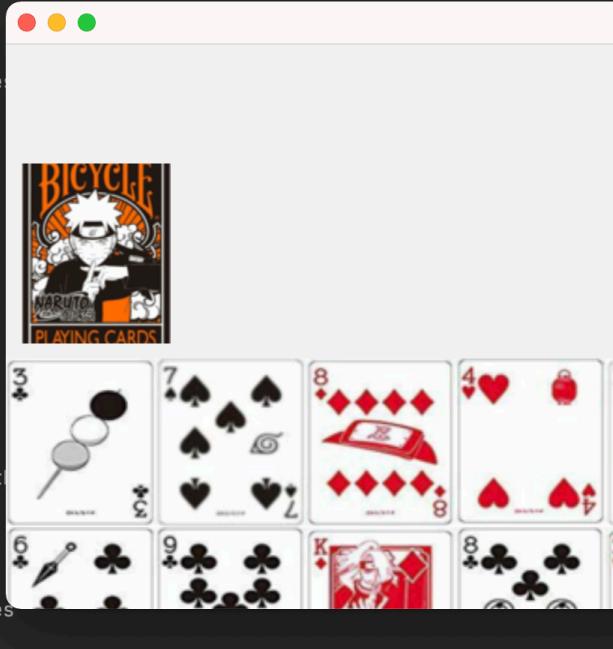
```
[3c, 7s, 8d, 4h, ks, qc, 2c, 9d, 6h, 4d, 5d, 4s, jh, 6c, 9c, kd, 8c, 3h, 2h, kc, js, 3s, 5s, kh, 2d, 8s, ac, 5h, 3d, as, 5c, th]
1 - Print the pack out (this is so you can check that it plays properly)
2 - Shuffle
3 - Deal a card
```

5 – Amalgamate:

```
[3c, 7s, 8d, 4h, ks, qc, 2c, 9d, 6h, 4d, 5d, 4s, jh, 6c, 9c, kd, 8c, 3h, 2h, kc, js, 3s, 5s, kh, 2d, 8s, ac, 5h, 3d, as, 5c, th]
1 - Print the pack out (this is so you can check that it plays properly)
2 - Shuffle
3 - Deal a card
4 - Make a move, move last pile onto previous one
5 - Make a move, move last pile back over two piles
6 - Amalgamate piles
7 - Print the displayed Cards on the command line
9 - Exit
8 - Play For Me!
6
Which Card Do you want To Move:
10
On top of which card do u want to place it:
10
1 - Print the pack out (this is so you can check that it plays properly)
2 - Shuffle
3 - Deal a card
4 - Make a move, move last pile onto previous one
5 - Make a move, move last pile back over two piles
6 - Amalgamate piles
7 - Print the displayed Cards on the command line
9 - Exit
8 - Play For Me!
7
[3c, 7s, 8d, 4h, ks, qc, 2c, 9d, 6h, 4d, 5d, 4s, jh, 6c, 9c, kd, 8c, 3h, kc, js, 3s, 5s, kh, 2d, 8s, ac, 5h, 3d, as, 5c, th]
1 - Print the pack out (this is so you can check that it plays properly)
2 - Shuffle
3 - Deal a card
```



```
[7s, 8d, 4h, ks, 3c, 2c, 9d, 6h, 4d, 5d, 4s, jh, 6c, 9c, kd, 8c, 3h, kc, js, 3s, 5s, kh, 2d, 8s, ac, 5h, 3d, as, 5c, th]
1 - Print the pack out (this is so you can check that it plays properly)
2 - Shuffle
3 - Deal a card
4 - Make a move, move last pile onto previous one
5 - Make a move, move last pile back over two piles
6 - Amalgamate piles
7 - Print the displayed Cards on the command line
9 - Exit
8 - Play For Me!
6
Which Card Do you want To Move:
4
On top of which card do u want to place it:
2
Invalid
8 - Play For Me!
1 - Print the pack out (this is so you can check that it plays properly)
2 - Shuffle
3 - Deal a card
4 - Make a move, move last pile onto previous one
5 - Make a move, move last pile back over two piles
6 - Amalgamate piles
7 - Print the displayed Cards on the command line
9 - Exit
7
[7s, 8d, 4h, ks, 3c, 2c, 9d, 6h, 4d, 5d, 4s, jh, 6c, 9c, kd, 8c, 3h, kc, js, 3s, 5s, kh, 2d, 8s, ac, 5h, 3d, as, 5c, th]
1 - Print the pack out (this is so you can check that it plays properly)
```



6 – Show Displayed Cards:

```

1 - Print the pack out (this is so you can check that it plays properly)
2 - Shuffle
3 - Deal a card
4 - Make a move, move last pile onto previous one
5 - Make a move, move last pile back over two piles
6 - Amalgamate piles
7 - Print the displayed Cards on the command line
8 - Exit
9 - Play For Me!
10
[5c, 8s, 2d, as, th]
1 - Print the pack out (this is so you can check that it plays properly)
2 - Shuffle
3 - Deal a card
4 - Make a move, move last pile onto previous one
5 - Make a move, move last pile back over two piles
6 - Amalgamate piles
7 - Print the displayed Cards on the command line
8 - Exit
9 - Play For Me!
10

```

Patience - mrm19

5

7 – Auto play:

```

[5c, 8s, 2d, as, th, ts, td, 4c, jd, 7c, jc]
1 - Print the pack out (this is so you can check that it plays properly)
2 - Shuffle
3 - Deal a card
4 - Make a move, move last pile onto previous one
5 - Make a move, move last pile back over two piles
6 - Amalgamate piles
7 - Print the displayed Cards on the command line
8 - Exit
9 - Play For Me!
10
[5c, 8s, 2d, as, th, ts, td, jc, jd, 7c]
1 - Print the pack out (this is so you can check that it plays properly)
2 - Shuffle
3 - Deal a card
4 - Make a move, move last pile onto previous one
5 - Make a move, move last pile back over two piles
6 - Amalgamate piles
7 - Print the displayed Cards on the command line
8 - Exit
9 - Play For Me!
10
[5c, 8s, 2d, as, th, ts, td, jc, jd, 7c]
1 - Print the pack out (this is so you can check that it plays properly)
2 - Shuffle
3 - Deal a card
4 - Make a move, move last pile onto previous one
5 - Make a move, move last pile back over two piles
6 - Amalgamate piles
7 - Print the displayed Cards on the command line
8 - Exit
9 - Play For Me!
10

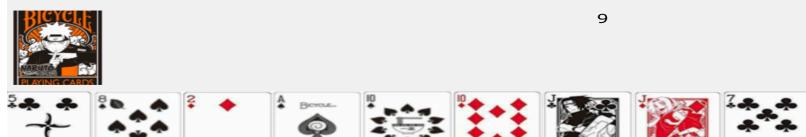
```

Patience - mrm19

10

Patience - mrm19

9



Rules

```

- This is how it works:
- The object of the game is to end up with one pile/stack of cards.
- Shuffle a deck of cards.
- Lay down the first card face up.
- Lay down the next one next to it, and so on.
- This would lead you to 52 cards all facing upwards, except that you can put one pile on another if any of the following apply:
o. They are next to each other and the same number or suit.
o. There are two piles between them, and they are the same number or suit..

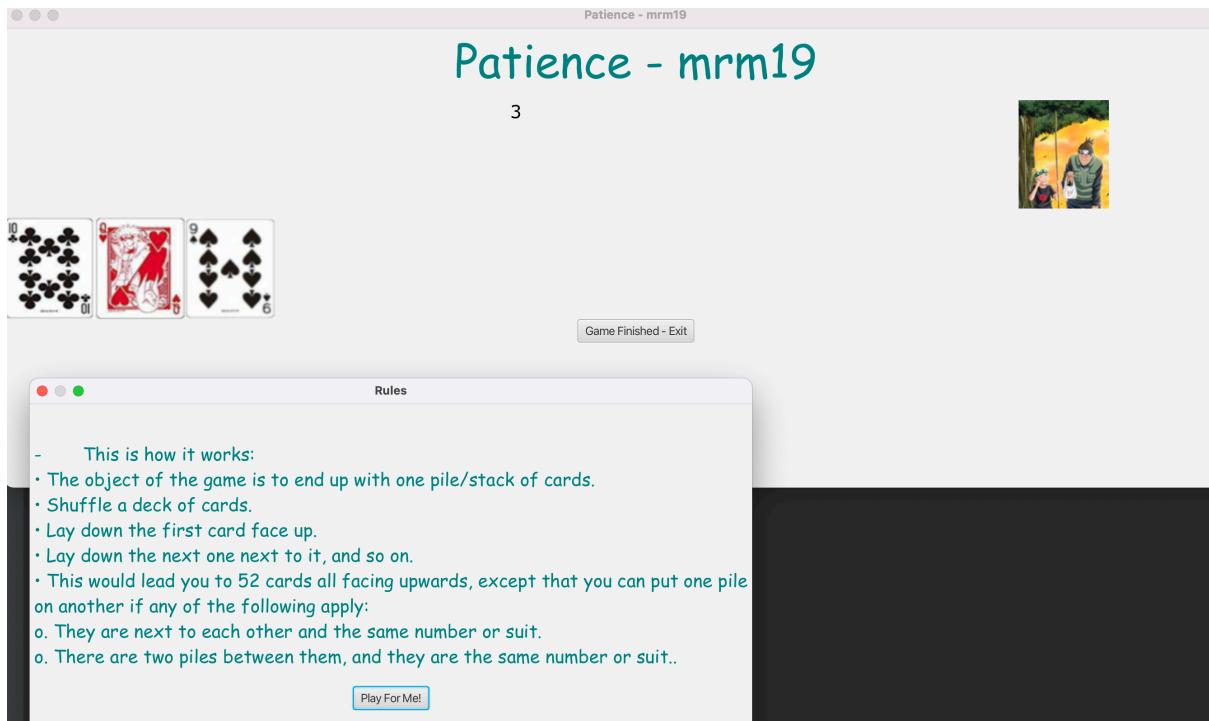
```

Play For Me!

```

9 - Exit
8 - Play For Me!
[5c, 8s, 2d, as, th, ts, td, jc, jd, 7c]
1 - Print the pack out (this is so you can check that it plays properly)
2 - Shuffle
3 - Deal a card
4 - Make a move, move last pile onto previous one
5 - Make a move, move last pile back over two piles
6 - Amalgamate piles
7 - Print the displayed Cards on the command line
8 - Exit
9 - Play For Me!
10

```



8 – Quit

```
1 - Print the pack out (this is so you can check that it plays properly)
2 - Shuffle
3 - Deal a card
4 - Make a move, move last pile onto previous one
5 - Make a move, move last pile back over two piles
6 - Amalgamate piles
7 - Print the displayed Cards on the command line
9 - Exit
8 - Play For Me!
9
Good Bye!
```

```
Process finished with exit code 0
```

You can Play Using Command Line or GUI both are working fine.

Evaluation

First, I started by reading the assignment brief aiming to comprehend the requirements and workings of the game. I started framing a rudimentary concept of the software and worked my way up before going straight up in to coding.

I focused more on the GUI because it was my first time using one for an application or game, and I wanted to learn everything I could. I made the game entirely graphical while keeping the CLI in place for testing purposes.

The user can click on the cards and buttons to engage and play the game.

I completely updated the supplied deck of card with a Naruto themed deck.

I created two menus:

1. Start menu, which offers only two options: start and exit. It was also planned to include a score option that displayed the top ten people's scores, but due to time constraints, I was unable to add it.
2. Game Controls Menu: This is solely for CLI because a menu is unnecessary while using GUI. To play the game, simply click on the buttons and cards.

I also attempted to incorporate sound effects into the game. I researched it and came up with a solution. It compiled with no errors, but the sounds did not play. I kept it there so that I might work on it later.

Conclusion:

This assignment was the most demanding and challenging of any other project I worked on. However, working on it helped me develop a problem-solving approach as well as the capacity to do independent research. I had the most fun designing the graphical user interface. This experience will be extremely beneficial to me in the following years and projects.

Your assessment of the mark you would award yourself for the work (based on the assessment criteria detailed):

Ist: 70-80%:

I believe that the written components are professionally presented in both layout on the page and logical structure with a high grammatical standard. Each Task is implemented using appropriate technology and will at least completely fulfil the functional requirements.