Nicholas Vadivelu, Frederick Ngo, Danyal Abbasi

President Card Game

User Guide

Getting Started

President is a card game that involves 2 or more players. The cards are dealt at random, and players are allowed to see their own cards, but may not see the cards of their opponents. The first player to play their cards is the person who has the 3 of Diamonds. The players then continue to play in a cycle, typically clockwise. You may play between 1 to 4 cards per turn if you are the first to play for that round, but they must all be the same rank (for example, they must all be Kings). If you are not first in that round, you must play the same number of cards as the person who initially started the round, and your cards must be of a higher rank. If they are the same rank, they must be of a higher suit (the order is as follows: Diamonds, Clubs, Hearts, Spades). You may choose to pass if you are unable to or do not want to play any cards. Once you passed, however, you may not play again until the round is over. The round ends when all but one player has passed, which in turn means all the cards are disposed in the "Trash", and the person who last played opens the next round. The game is over when all the players but one have disposed all the cards in their hand. The objective is to get rid of all your cards first. The winner is declared President, second is declared Vice President, third is declared Neutral, and the loser is declared Bum.

To start the program, double click the "President.jar" file included in the zip file. You will be presented with a welcome screen. You must click anywhere on the blue area of the window or the "Start Game" button to start the game. The window will turn into a playing field with four players: three CPUs and you. The player with the 3 of Diamonds will play first, and the order is clockwise. Information about all of the players is displayed in a small red box connected to that player's cards. It will show their name, if they have won, if they have passed, or if it is their turn (represented by a blue star). When it is your turn (shown by the blue star), you may select cards by clicking them. The cards are sorted in order of rank then suit for you. You may choose to deselect cards by clicking them again. Please note that you must click the **left side of the card** (with the exception of the final card, which is completely responsive). This is important to prevent misclicks. You can play the selected cards by clicking the "Play" button found at the bottom of the window. You may pass if by clicking the "Pass" button found to the right of the "Play" button, however, you are not allowed to pass if you the first to play in a given round. You can find information about what is happening in text form to the left of the "Play" button.

Features

1. Advanced AI that is the appropriate difficulty level for most players.
2. Mouse Listener that allows you to interact with your cards via clicking.
3. Welcome screen you are presented with that gives you an introduction and instructions to the program.
4. Buttons that allows you to start the game, play the cards you have selected, and pass.
5. Message Panel that pop up to notify you if you choose invalid cards.
6. Message Panel that pop up to list the rankings
7. Information on the bottom of the screen pertaining to action on the field.

8. Option Pane that pops up to allow you to play a new game.
9. Graphical interface that shows the game visually.
10. Rankings that are stored.
11. All the basic functionality of the President Card Game.
12. Ability to choose backgrounds and card backs

Limitations

1. No nukes or poker hands.
2. No card exchange on games after the first game.
3. Only left edge of cards are responsive, except for the final card (to prevent misclicks)

Bugs

1. The program will not accept if you pass when you start a round (rule in president)
2. The program does not refresh the display as intended when starting a new round

Developers Guide

Overview

This digital version of the popular card game President was designed using various classes, methods, GUI components, and other objects. This program was carefully designed in order to be able to easily modify the code. Most of the variables and methods are self documenting, and have comments explaining their functionality when they are declared. All of the primary functionality can be found in the PresidentCardGame constructor. It utilizes the player class to create an array of four players: 3 computers and 1 human. The following code is what runs the primary game, then the code after the while(!gameOver) deals with what happens at the end of the game. An instance of the GUI is created in the PresidentCardGame constructor to deal with the interface, and the GUI has the current instance of the PresidentCardGame as an argument. To deal with the hand of the players and the cards themselves, there exists a 5 Deck instances (one for each player's hand + the initial deck) and a total of 52 cards (distributed between the hands).

Section for each feature

1. The AI operates as follows
    a. When it is not the first of the round it will play the lowest possible cards. If it cannot play anything, it will pass. The AI will also refrain from playing cards that are too high too early in the game, although they are programmed to be fairly aggressive.
        i. The AI will NOT break doubles, triples, or quadruples early in the game, it will save them until later in the game. Although sometimes this leads to its downfall, usually this tactic is very effective.
    b. When the AI is first to play in a round:
        i. First, it plays the longest set of cards it has, meaning it will prioritize quadruples, then triples, then doubles, then finally singles. If it has multiple quadruples for example, it will play the lower one.
            1. Early in the game, however, the AI will refrain from playing cards that are too high, but again, it is quite aggressive.
    c. If the AI cannot play anything, it will return that same cards that were played by the previous players. This indicates that the AI has passed.
2. The mouse listener is setup to analyze where on the screen you click. By using a for loop that checks the position of each of the cards in your hand, the program is able to identify which card you have selected. The deselection works in the same fashion.
3. The welcome screen has a button on the bottom and has a mouselistener listening to the entire board.
4. The buttons have an action listener on them, which allows the user to provide input through them. The start game button is the same button as the play button with different text, and will change into the Play button when the user starts the game. The play button will access the boolean array that indicates which cards you have selected, which the choose method uses as your selection. The pass button returns the cards that were previously played, which indicates to the main class that you have passed.

5. The message that pops up for invalid cards is a JOptionPane. This will pop up if the user's cards are not of higher rank/suit, does not match the length of the previous cards played, and/or are not consistent within themselves.
6. The panel that pops up to show the rankings is JOptionPane. This uses the rankings data field in each player to determine the order.
7. The information at the bottom of the field matches exactly what is going on the board, and is controlled by a JLabel.
8. The option pane that gives you an opportunity to play another game controls the condition on a loop. By clicking yes or no, you are setting the condition to true or false respectively, which will restart on end the game respectively.
9. The graphical interface is found solely in the GUI class, and makes use of various components, interfaces and classes including JPanels, JLabels, MouseListener, ActionListener, and JButtons.
10. The rankings, as mentioned earlier, are stored in each player as a data field.
11. The basic rules for president are fully implemented in the PresidentCardGame class in the constructor, in the validate method found in the Human Class, and the choose method found in the Computer Class. The latter two ensure that the cards are valid, based on the rules explained above. The logic in the constructor ensures that the appropriate players play at the appropriate times, and can include changing the order and skipping over plays due to passing/winning.
12. The card backs are updated using JComboBoxes paired with item listeners found in the GUI class. There are string variables in the main PresidentCardGame class also storing the names of the selected backs. By passing the cardBack name into the constructor of each card, or by calling the setCardBack(String name) method, the program is able to change the card backs with ease. Similarly, the GUI can change the background of the display using the name shown in the combobox.

Section for each class

PresidentCardGame

- This class controls all the primary logic of the program. The logic in the constructor ensures that the appropriate players play at the appropriate times, and can include changing the order and skipping over plays due to passing/winning. This also controls sends information to the GUI class to effectively display the information. The PresidentCardGame class also changes the players' status, such as passing, winning, rankings, etc.

GUI

- This class extends JFrame and controls all the display aspects of the game. The constructor initializes all the components, along with storing references to the main objects from the PresidentCardGame class, including the players.

Player

- This abstract class stores information about all the players such as whether they passed, won, is their turn, their standing, name, etc along with the getters and setters for each. This class has the abstract method choose that is implemented by both its subclasses, as it is a crucial method to allow players to choose cards.

Computer

- This class extends the Player class and is the AI of the program, and the thought process was outlined above. The computer also names itself based on its position relative to the other players.

Human

- This class extends the Player class and represents the human player. It receives the user's choice of cards and also validates that they are acceptable. Code to simulate text input is also included for debugging purposes.

Card

- This class holds all the information for the cards in the game, including their image, rank, suit, value, whether they are faceup, and more. This class also has several getter and setter methods, along with methods that can flip cards, rotate cards, compare cards, and more.

Deck

- This class holds a set of cards, and is used in the program to create the initial deck and used for each player's hands. This class is very flexible with many overloaded methods for convenience, such as the deal method. Along with the getters and setters, this class can deal cards, search for cards, sort the cards, return cards, and flip cards.

Suggestions for Improvement

There are several improvements that can be made, and would have been made by the primary development team if there was adequate funding, resources, and time.

- Implement an exchange system for players to exchange cards based on who is president, vice president, neutral, and bum.
- Implement keylisteners to allows for keyboard input
- Create animations that make the cards move on the board for a more immersive experience
    - We did not implement this feature due to the speed restrictions on school computers, as our game became laggier than we would like it to be.
- Include background music to create a more pleasant experience.
    - We did not implement this feature due to the speed restrictions on school computers, as our game became laggier than we would like it to be.
- Implement multiplayer features so users can play with their friends.
- Implement the use of Nukes and Poker Hands
- Possibly force the first player to always play 3 of Diamonds, however, this is only required in some versions of President
- Implement a feature that allows users to watch the computers play against each other