

PlzHaveMercy Documentation

Created by

Nalin Baipluthong 6330271121
Parin Opasphatikul 6330308821

**2110215 Programming Methodology
Semester 1 Year 2021
Chulalongkorn University**

PoorPresident

Introduction

PoorPresident is inspired by the famous 4-player strategy card game named "President". The objective of the game is to play cards in hand until find 3 winners with their hands empty.

Rules

Card Points

card points are divided into 2 parts depending on

1. The card value

The card values are ranked high to low in this order:

2 > A > K > Q > J > 10 > 9 > 8 > 7 > 6 > 5 > 4 > 3

2. The card suit

The four suits are ranked high to low in this order:

the suit of spades(♠), the suit of hearts(♥), the suit of diamonds(♦), and the suit of clubs(♣).

Cards are to be ranked first from higher to lower value, then shall the suit be considered.

How to play the cards

Card play in the game of PoorPresident is sorted into 2 categories which are

- a.) Odd cards (playing a single or three cards)
- b.) Even cards (playing two or four cards)

When a category of cards is chosen for the game, players have to play with the chosen category until the table is reset (all players pass) then starting a new round. The first player of the round determines the category of cards to be played in that round.

The card play is formatted as such:

- Playing a single card: This is when a player plays one card. The next player is to play a single card from a superior point or play three cards.
- Playing two cards: This is when a player plays two cards. The next player is to play two cards from a superior point or play four cards.
- Playing three cards: This is when a player plays three cards. The next player is to play only three cards from a superior point.

- Playing four cards: This is when a player plays four cards. The next player is to play only four cards from a superior point.

Every format of card play that the player chooses to play. The set of cards that they choose must have the same value.

How to play PoorPresident

- 1) Distribute a deck of cards to 4 players. The real player, who is playing the Poorpresident, is at the bottom right and the other 3 players are bots. Each player will receive 13 cards which can see the number of player cards at the top of their profile.



- 2) The player with 3 clubs(♣) starts the game by putting the 3 clubs(♣) into the desk (the player can start the game with a single card, two, three, or four cards—but the 3 clubs(♣) must be included in the play).



- 3) Players take turns playing clockwise.
- 4) Players are to play the cards with superior points than the cards on the desk. (Play button of the real player is on center right)
- 5) Players with no card to play or players who wish not to discard their cards in that round must make a pass (the pass button of the real

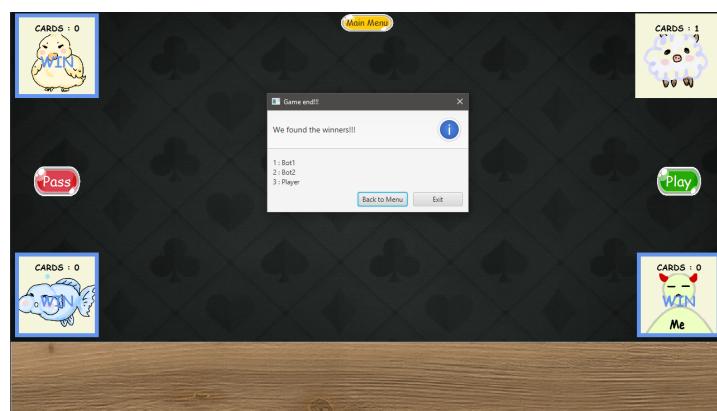
player is on center left), and can play again after the desk table is cleared for another round.



- 6) A round ends when players do not have cards that are superior to the cards on the desk. Reset the cards from the desk. The last player to play starts the round. The game continues as such until players are left with their hands empty.



- 7) The game ends when 3 players are left with their hands empty.



Example

- There are 3 statuses of player :



Playing



Pass



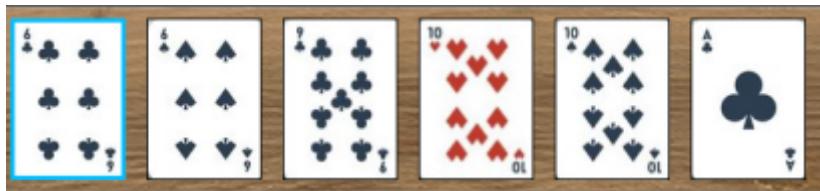
Win

Playing: when it's your turn to play.

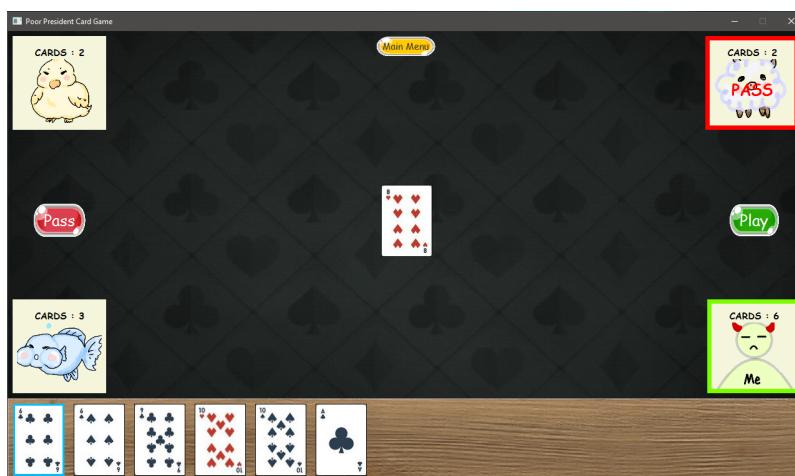
Pass: when you pass you can't play the card in that round again.

Win: You play all cards in your hand

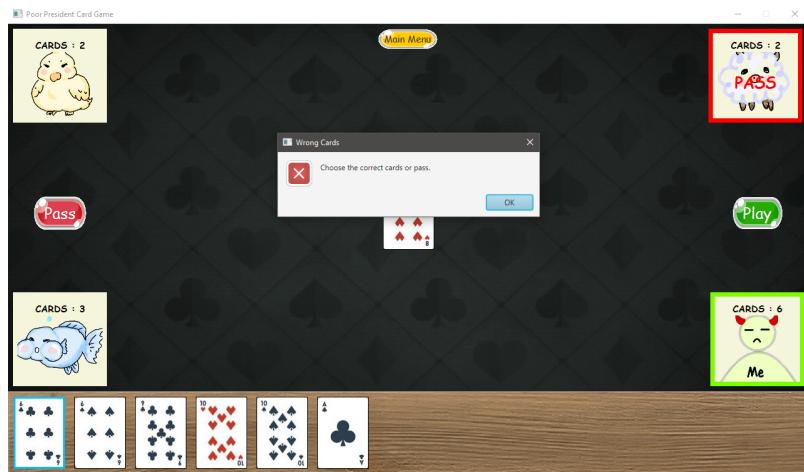
- When you click the card on your turn it will highlight the blue border that means you choose that card to play. You can re-click to unselect it.



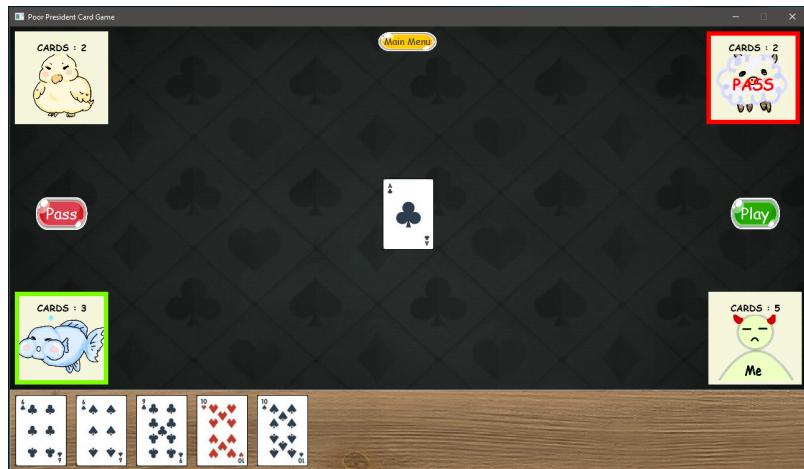
- When you click play, if the cards you choose are not correct according to the rules, the error will occur.



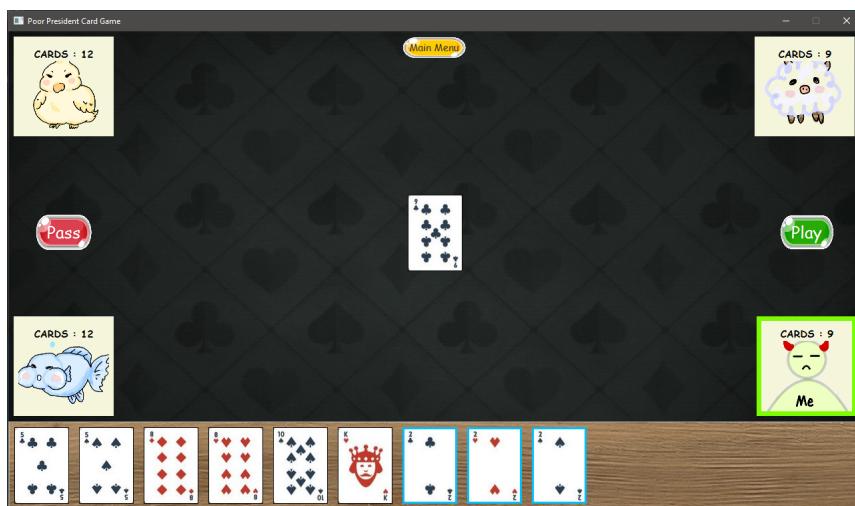
- In this case, 6 is less than 8.



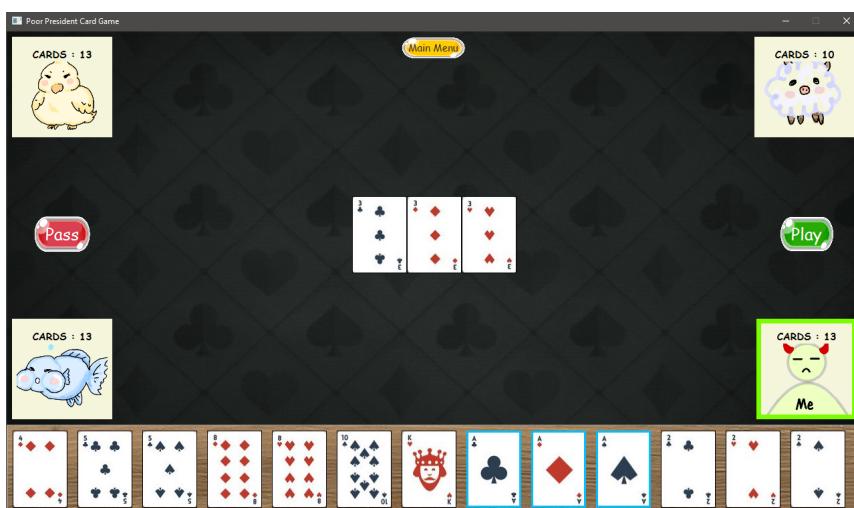
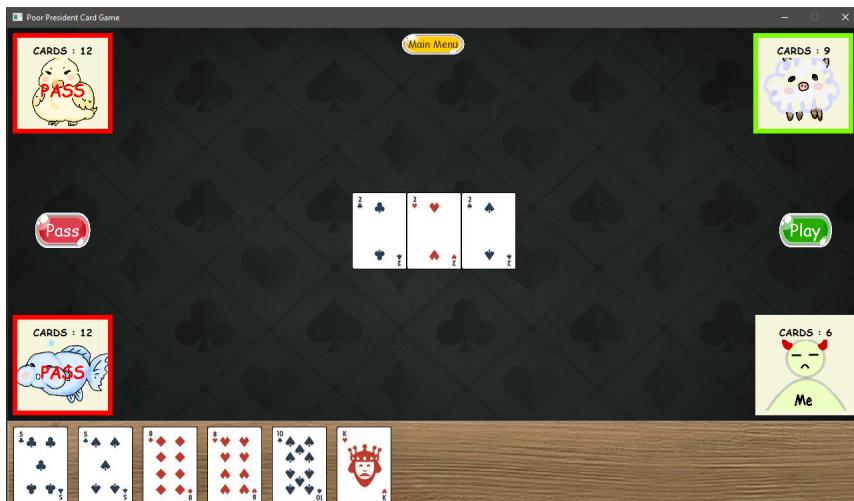
- If you change to Ace you can play it and the card will be removed from your hand and placed on the table.



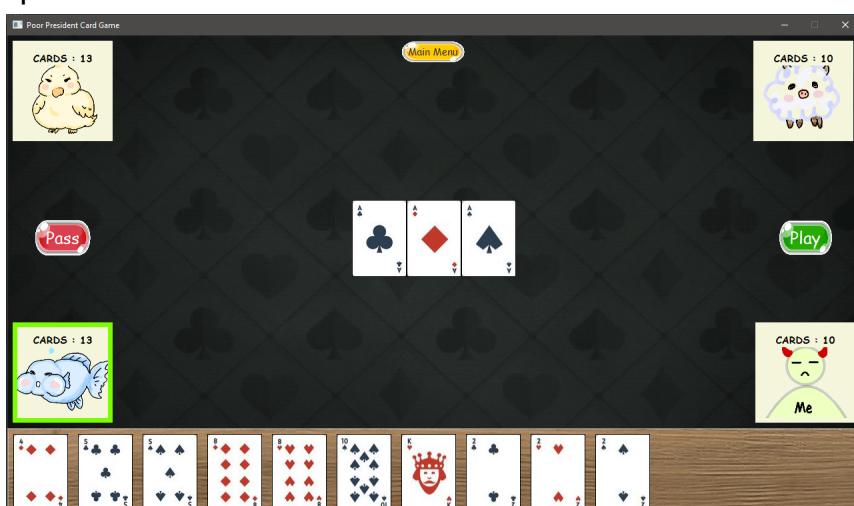
- Playing 3 cards with the same value to beat a single card.



- Playing 3 cards that are superior points to 3 cards on the table.



- The Ace point is more than 3.

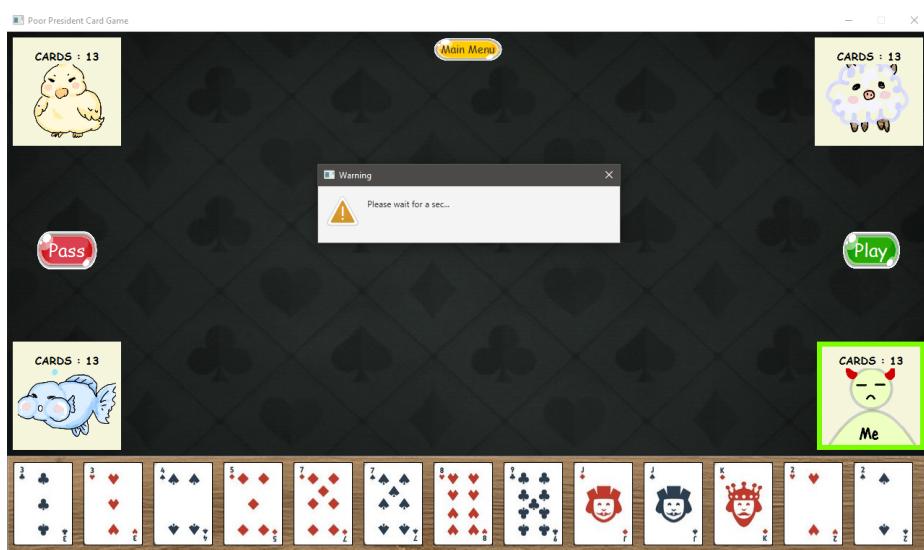


- In case playing 2 cards and 4 cards are almost the same as above just change single to two and 3 cards to 4 cards.

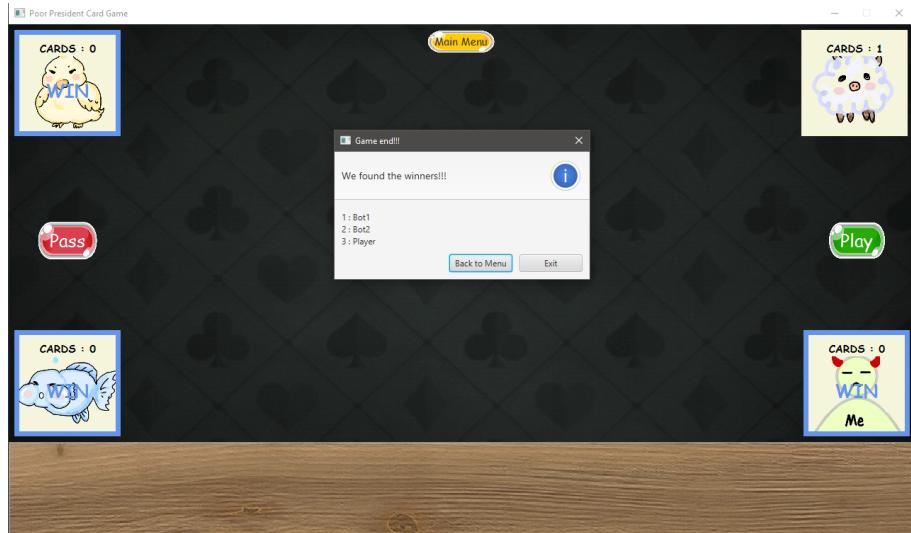
- You can click the menu button to go back to the menu but the game won't be saved for you.



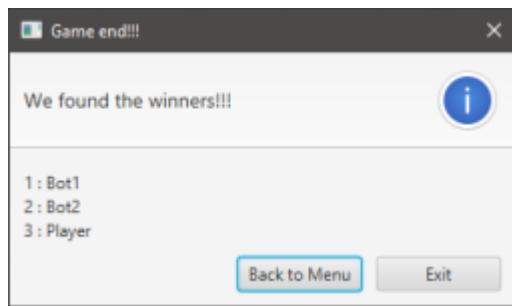
- If you click the Yes button It will clear the table and restart the game.



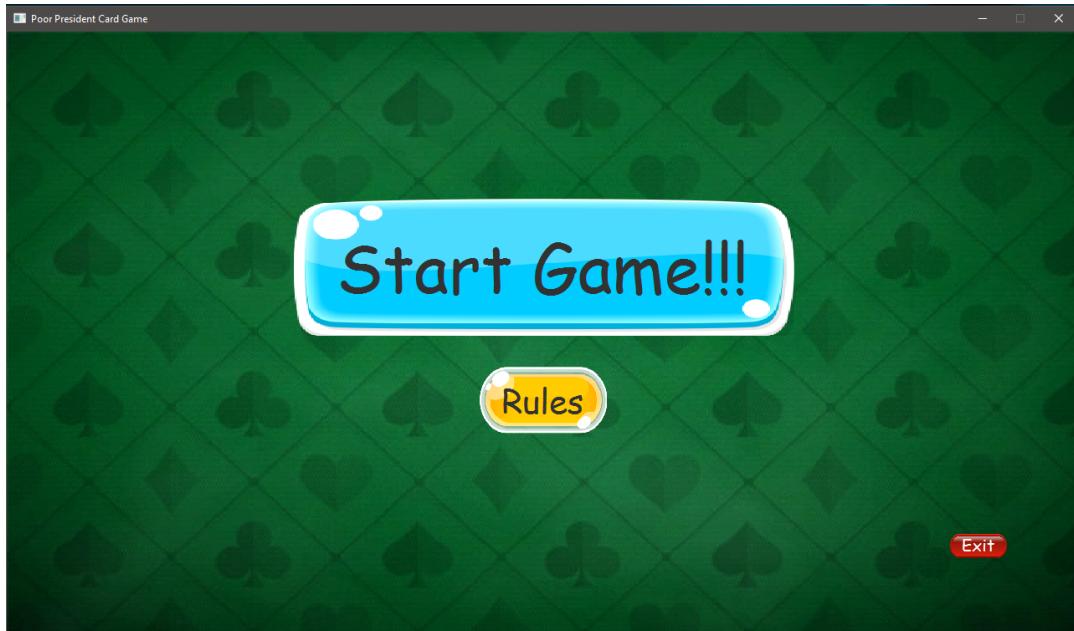
- If there are 3 winners the game will end and show the rank of the winners.



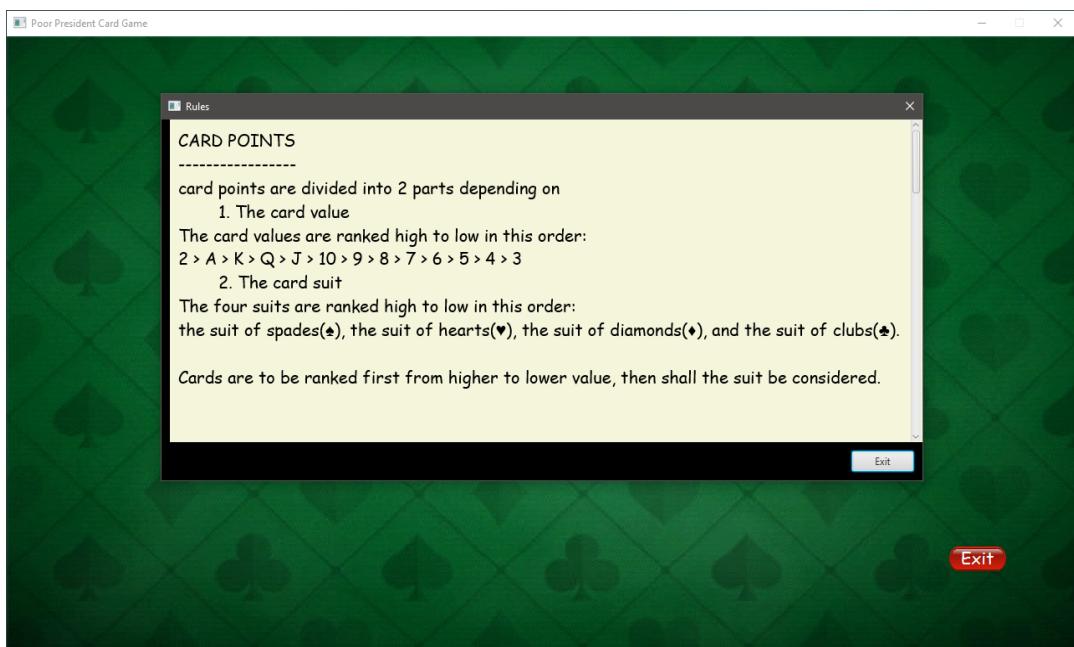
- You can click the Back to Menu button to go back to the menu or the Exit button to exit the game.



MainMenu scene

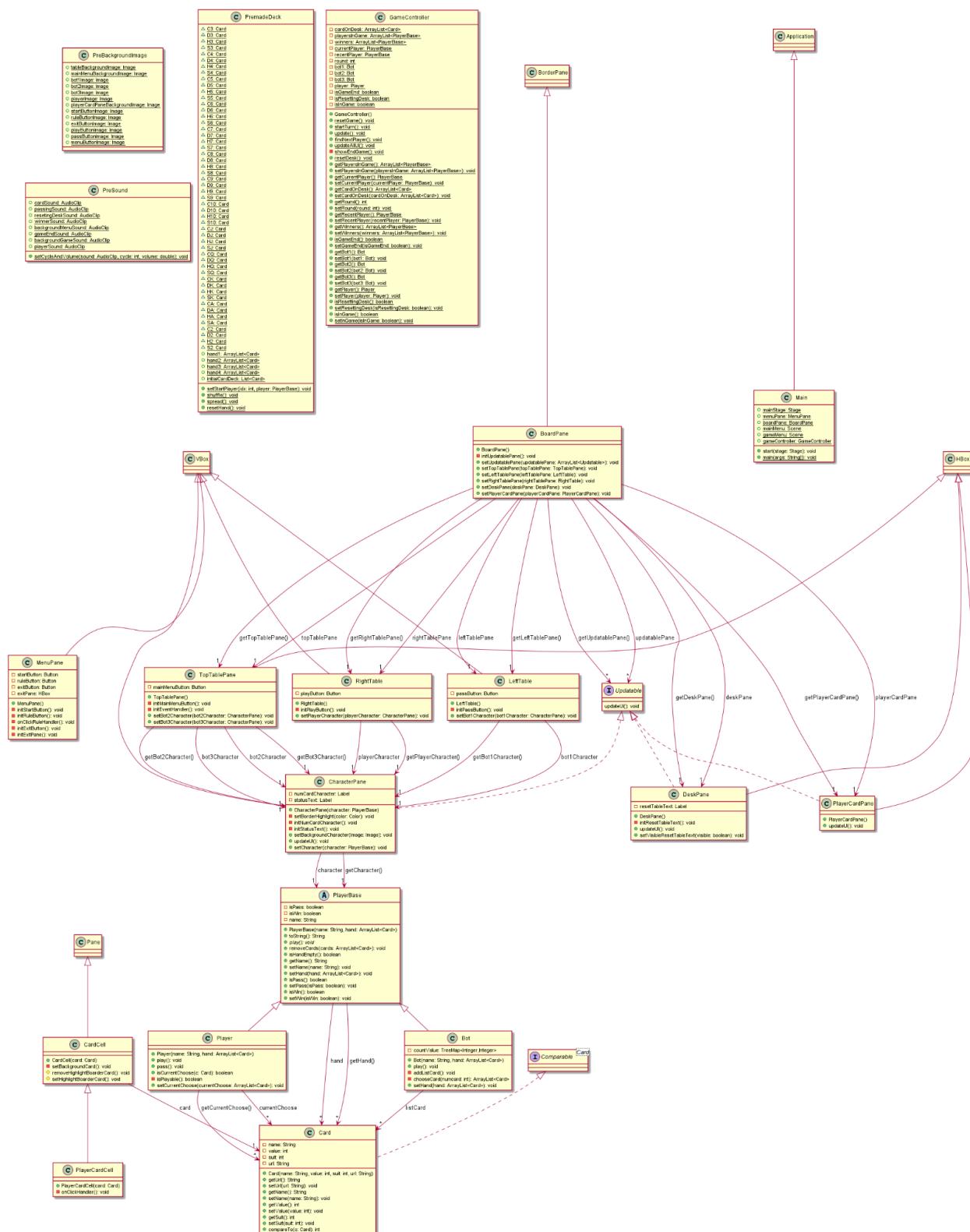


You can click start game to start the game.



The rules button will tell you how to play and the exit button to exit the game.

Class diagram



To see a high-quality picture click on the link below:

https://drive.google.com/file/d/1yalnB36SP6SoiY_Tz7MeSd2mO2IY6uu/view?usp=sharing

1. Package card

1.1 class Card implements Comparable<Card>

1.1.1 Fields

- String name	Name of the card
- int value	Value of the card
- int suit	Suit of the card
- String url	URL to the image of the card

1.1.2 Constructor

+ Card(String name, int value, int suit, String url)	-Initialize Fields
------------------------------------------------------	--------------------

1.1.3 Methods

+ int compareTo(Card o)	Compare the value of the card first if the value is equal to the other card then compare the suit instead. This method return 0 if two cards are the same card, return 1 if more than the other card and return -1 if less than the other card.
+ setter and getter for url, name, value, suit	

2. Package deck

2.1 class PremadeDeck

2.1.1 Fields

# Card C3, D3, H3, S3, C4, D4, H4, S4, C5, D5, H5, S5, C6, D6, H6, S6, C7, D7, H7, S7, C8, D8, H8, S8, C9, D9, H9, S9, C10, D10,	Initialize every card in deck
----------------------------------------------------------------------------------------------------------------------------------	-------------------------------

<u>H10, S10, CJ, DJ, HJ, SJ, CQ, DQ, HQ, SQ, CK, DK, HK, SK, CA, DA, HA, SA, C2, D2, H2, S2</u>	
<u>+ ArrayList<Card> hand1, hand2, hand3, hand4</u>	Initialize hand (ArrayList<Card>) of players with null
<u>+ List<Card> initialCardDeck</u>	List contain every card

2.1.2 Methods

<u>+ void setStartPlayer(int idx, PlayerBase player)</u>	Set start player to who has 3 Club Card use in spread()
<u>+ void shuffle()</u>	Shuffle card in initialCardDeck
<u>+ void spread()</u>	Add a card from the initialCardDeck deck to players hands and call setStartPlayer to find the start player of the game
<u>+ void resetHand()</u>	Reset card of all players hand

3. Package sound

3.1 class PreSound

3.1.1 Fields

<u>+ AudioClip cardSound</u>	Sound effect when playing the card
<u>+ AudioClip passingSound</u>	Sound effect when passing
<u>+ AudioClip resettingDeskSound</u>	Sound effect when resetting desk
<u>+ AudioClip winnerSound</u>	Sound effect when the hand is empty
<u>+ AudioClip backgroundMenuSound</u>	Background sound in Main Menu
<u>+ AudioClip gameEndSound</u>	Sound effect when 3 players win
<u>+ AudioClip backgroundGameSound</u>	Background sound in Game

+ AudioClip playerSound	Sound effect when It's your turn
-------------------------	----------------------------------

3.1.2 Methods

+ void setCycleAndVolume(AudioClip sound, int cycle, double volume)	-Set cycle of sound -Set volume
---------------------------------------------------------------------------	------------------------------------

4. Package image

4.1 class PreBackgroundImage

4.1.1 Fields

+ Image tableBackgroundImage	Background table in game
+ Image mainMenuBackgroundImage	Background in Main Menu
+ Image bot1Image	Image background of Bot1
+ Image bot2Image	Image background of Bot2
+ Image bot3Image	Image background of Bot3
+ Image playerImage	Image background of Player
+ Image playerCardPaneBackgroundImage	Background in PlayerCardPane
+ Image startButtonImage	Background for Start Button
+ Image exitButtonImage	Background for Exit Button
+ Image playButtonImage	Background for Play Button
+ Image passButtonImage	Background for Pass Button
+ Image menuButtonImage	Background for Menu Button and Rules Button

5. Package player

5.1 abstract class PlayerBase

5.1.1 Fields

- boolean isPass	Check pass status of the player
- boolean isWin	Check win status of the player
- String name	Name of player
- ArrayList<Card> hand	ArrayList of cards in player's hand

5.1.2 Constructor

+ PlayerBase(String name, ArrayList<Card> hand)	-Intialize Fields
----------------------------------------------------	-------------------

5.1.3 Methods

+ String toString()	Return player's name and name of cards in hand
+ void play()	Abstract method for be used in Bot and Player class when play card
+ void removeCards(ArrayList<Card> cards)	Remove cards in hand that equal cards in ArrayList
+ boolean isEmpty()	Check if the hand is empty
+ setter and getter for name, hand, isPass and isWin	

5.2 class Player extends PlayerBase

5.2.1 Fields

- ArrayList<Card> currentChoose	ArrayList of cards that player chooses to play
---------------------------------	------------------------------------------------

5.2.2 Constructor

+ Player(String name, ArrayList<Card> hand)	-Intialize Fields
------------------------------------------------	-------------------

5.2.3 Methods

+ void play()	Check playable() if playable is true: -play cardSound -call method GameController.setCardOnDesk(currentChoose); and GameController.setRecentPlayer(this); -create thread and put GameController.update(); in thread then start -call removeCards(currentChoose); for remove card in hand -clear currentChoose else: -show alert "Wrong Card"
+ void pass()	Check cardOnDesk in GameController if it is empty: -show alert "You need to play some cards." else: -play passingSound -set player's isPass to be true -clear currentChoose -create thread and put GameController.update(); in thread then start
+ boolean isCurrentChoose(Card card)	If card is in currentChoose return true else return false
- boolean playable()	Check condition that currentChoose which are picked by the player can play in the game if currentChoose is correct return true else return false
+ setter and getter for hand	

5.3 class Bot extends PlayerBase

5.3.1 Fields

- TreeMap<Integer, Integer>	TreeMap for contain the number of
-----------------------------	-----------------------------------

countValue	each value card that are in bot hand
- ArrayList<Card> listCard	ArrayList of card that be add for bot to play each turn

5.3.2 Constructor

+ Bot(String name, ArrayList<Card> hand)	-Intialize Fields
---------------------------------------------	-------------------

5.3.3 Methods

+ void play()	Call addListCard() to choose card to play if listCard is still empty: -play passingSound -set bot's isPass to be true else: -play cardSound -call method GameController.setCardOnDesk(list Card); and GameController.setRecentPlayer(this); -create thread and put GameController.update(); in thread then start -call removeCards(listCard); for remove card in hand -clear listCard
- void addListCard()	Add card to listCard depends on card on desk and call chooseCard(numcard) to choose which number of cards should bot play
- ArrayList<Card> chooseCard(int numcard)	Return ArrayList of card with the same amount of numcard for bot to play on desk
+ void setHand(ArrayList<Card> hand)	-Set hand -Set countValue

6. Package gui.game

6.1 interface Updatable

6.1.1 Methods

+ void updateUI()	Update UI
-------------------	-----------

6.2 class CardCell extends Pane

6.2.1 Fields

# Card card	The card that is used in CardCell
-------------	-----------------------------------

6.2.2 Constructor

+ CardCell(Card card)	-Initialize Fields -Set width and height -Set padding -Call removeHighlightBorderCard() -Call setBackgroundCard()
-----------------------	-------------------------------------------------------------------------------------------------------------------------------

6.2.3 Methods

- void setBackgroundCard()	Set background image according to card in CardCell
# void removeHighlightBorderCard()	Set Border to Black
# void setHighlightBorderCard()	Set Border to Blue

6.3 class PlayerCardCell extends CardCell

6.3.1 Constructor

+ PlayerCardCell(Card card)	-Initialize Fields -Add event when mouse click handle if It is player turn and It is not resetting desk: call onClickHandler()
-----------------------------	-----------------------------------------------------------------------------------------------------------------------------------

6.3.2 Methods

- void onClickHandler()	If this card is not in currentChoose of player: -remove the card from currentChoose of player -call removeHighlightBorderCard(); else: -add the card to currentChoose of player -call setHighlightBoderCard()
-------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

6.4 class CharacterPane extends VBox implements Updatable

6.4.1 Fields

- Label numCardCharacter	Label number of cards in character hand
- PlayerBase character	The character that is used in CharacterPane
- Label statusText	Show status of the character

6.4.2 Constructor

+ CharacterPane(PlayerBase character)	-Initialize Fields -Set width and height -Set padding -Set spacing -Set alignment -Set Border and Background color -Call initNumCardCharacter -Call initStatusText() -Call updateUI() -Add numCardCharacter and statusText to children of CharacterPane
---------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

6.4.3 Methods

- void setBorderHighlight(Color color)	Set Border Color to color
- void initNumCardCharacter()	-Initialize numCardCharacter -Set TextFill -Set Background color -Set Font and alignment
- void initStatusText()	-Initialize statusText -Set Text to "PASS" -Set Color and Font -Set visible false
+ void setBackgroundCharacter(Image image)	Set background image
+ void updateUI()	-Update numCardCharacter -Update status and border color If character is win: Set statusText to "WIN" Change border color to blue If it character turn: Set statusText to invisible Change border color to green if character is pass: Set statusText to visible Change border color to red else: Set border color to beige Set statusText to invisible
+ setter and getter for character	

6.5 class TopTablePane extends HBox

6.5.1 Fields

- Button mainMenuButton	Main Menu Button
- CharacterPane bot2Character	Bot2 Character
- CharacterPane bot3Character	Bot3 Character

6.5.2 Constructor

+ TopTablePane()	-Initialize Fields -Set padding -Set spacing -Call initMainMenuButton() -Call initEventHandler() -Set background character bot2,bot3 -Add to children of TopTablePane
------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

6.5.3 Methods

- void initMainMenuButton()	-Initialize the Main menu button -Set Font -Set alignment and background
- void initEventHandler()	-Add event when clicking Main menu button, it will alert pop up to confirm to back to menu
+ setter and getter for bot2Character and bot3Character	

6.6 class LeftTable extends VBox

4.6.1 Fields

- CharacterPane bot1Character	Bot 1 Character
- Button passButton	Pass Button

6.6.2 Constructor

+ LeftTable()	-Initialize Fields -Set padding -Set spacing -Set alignment -Call initPassButton() -Set background character bot1 -Add to children of LeftTable
---------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------

6.6.3 Methods

- void initPassButton()	-Initialize Pass button -Set Font, Text -Set background -Add event mouse click to pass the turn if it is your turn
+ setter and getter for bot1Character	

6.7 class RightTable extends VBox

6.7.1 Fields

- CharacterPane playerCharacter	Player Character
- Button playButton	Play Button

6.7.2 Constructor

+ RightTable()	-Initialize Fields -Set padding -Set spacing -Set alignment -Call initPlayButton() -Set background character player -Add to children of RightTable
----------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------

6.7.3 Methods

- void initPlayButton()	-Initialize Play button -Set Font, Text -Set background -Add event mouse click to play the card in current choose if it is your turn
+ setter and getter for playerCharacter	

6.8 class DeskPane extends HBox implements Updatable

6.8.1 Fields

- Label resetTableText	Resetting Table Text
------------------------	----------------------

6.8.2 Constructor

+ DeskPane()	-Initialize Fields -Set Padding -Set alignment -Call initResetTableText()
--------------	------------------------------------------------------------------------------------

6.8.3 Methods

- void initResetTableText()	-Initialize Rest Table text -Set color -Set Font and alignment -Set visible to false
+ void updateUI()	-Clear all children and Add Card on Desk from GameController Class to these children
+ void setVisibleResetTableText(boolean visible)	-Clear all children and add resetTableText to these children -Set visible resetTableText

6.9 class PlayerCardPane extends HBox implements Updatable

6.9.1 Constructor

+ PlayerCardPane()	-Set alignment -Set padding -Set width and height -Set border and background image -Call updateUI()
--------------------	-----------------------------------------------------------------------------------------------------------------

6.9.2 Methods

+ void updateUI()	-Clear all children then recreate PlayerCardCell of all cards in the player hand and add to PlayerCardPane children
-------------------	---------------------------------------------------------------------------------------------------------------------

6.10 class BoardPane extends BorderPane

6.10.1 Fields

- TopTablePane topTablePane	Top table pane
- LeftTable leftTablePane	Left table pane
- RightTable rightTablePane	Right table pane
- DeskPane deskPane	Desk pane
- PlayerCardPane playerCardPane	Player card pane
- ArrayList<Updatable> updatablePane	All updatable pane

6.10.2 Constructor

+ BoardPane()	-Initialize Fields -Set width and height -Set alignment every pane -Call initUpdatablePane() -Set background image
---------------	--------------------------------------------------------------------------------------------------------------------------------

6.10.3 Methods

- void initUpdatablePane()	-Initialize updatablePane and add all pane that implement Updatable interface
+ setter and getter for updatablePane, topTablePane, leftTablePane, rightTablePane, deskPane and playerCardPane	

7. Package gui.menu

7.1 class MenuPane extends VBox

7.1.1 Fields

- Button startButton	Start game button
- Button ruleButton	Rule button
- Button exitButton	Exit game button
- HBox exitPane	Exit pane

7.1.2 Constructor

+ MenuPane()	<ul style="list-style-type: none">-Initialize Fields-Set width and height-Set padding-Set spacing-Set Background image-Call initStartButton()-Call initRuleButton()-Call initExitButton()-Call initExitPane()-Set margin-Add startButton, ruleButton, and exitPane to children of MenuPane
--------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

7.1.3 Methods

- void initStartButton()	<ul style="list-style-type: none">-Initialize the Start game button-Add event when clicking the button then start the game and change the background sound
- void initRuleButton()	<ul style="list-style-type: none">-Initialize Rule Button-Add event when clicking the button call onClickRuleHandler
- void onClickRuleHandler()	<ul style="list-style-type: none">-Pop up an alert with a scroll pane to show the rule of the game
- void initExitButton()	<ul style="list-style-type: none">-Initialize Exit Button-Add event when clicking the button closes the game

- void initExitPane()	-Initialize the HBox pane -Set alignment -Add exitButton into it
-----------------------	------------------------------------------------------------------------

8. Package logic

8.1 class GameController

8.1.1 Fields

- ArrayList<Card> cardOnDesk	Contain card on desk which play by recent player
- ArrayList<PlayerBase> playersInGame	Contain 4 player: bot1, bot2, bot3 and player
- ArrayList<PlayerBase> winners	Contain winners in the game sorted from first to last
- PlayerBase currentPlayer	The current player in the game
- PlayerBase recentPlayer	The recent player who owns the cards on the desk
- int round	Round in game, count up +1 when resetting the desk
- Bot bot1	The first bot in game
- Bot bot2	The second bot in game
- Bot bot3	The third bot in game
- Player player	The real player in game
- boolean isGameEnd	Check if end game
- boolean isResettingDesk	Check if desk is being reset
- boolean isInGame	Check player is in game or on main menu

8.1.2 Constructor

+ GameController()	-Initialize Fields -Call method
--------------------	------------------------------------

	PremadeDeck.shuffle(); and PremadeDeck.spread(); to shuffle and spread deck for players -Set isGameEnd to false -Set round to 1
--	---------------------------------------------------------------------------------------------------------------------------------------

8.1.3 Methods

public static void resetGame()	-Set isInGame to false -Call PremadeDeck.resetHand(); to clear 4 hands in PremadeDeck
public static void startTurn()	If currentPlayer is a bot: -let's bot play by call currentPlayer.play(); else: -play playerSound
public static void update()	-Update winner -Update next player by call findNextPlayer(); -Call updateAllUI() -Create thread and put GameController.startTurn(); in thread then start
public static void findNextPlayer()	Find next player from current player and set current player to be next player instead
public static void updateAllUI()	Call Main.boardPane.getUpdatablePane() and update all updatable pane if game end: -call showEndGame();
private static void showEndGame()	Show alert which has list of winners in it
public static void resetDesk()	-Clear cardOnDesk -Set pass all playersInGame to false
+ setter and getter for playersInGame, currentPlayer,	

cardOnDesk, round, recentPlayer, winners, isGameEnd, bot1, bot2, bot3, player, isResettingDesk and isInGame	
-------------------------------------------------------------------------------------------------------------	--

9. Package main

9.1 class Main extends Application

9.1.1 Fields

<u>+ Stage mainStage</u>	Main stage
<u>+ MenuPane menuPane</u>	Menu pane
<u>+ BoardPane boardPane</u>	Board game pane
<u>+ Scene mainMenu, gameMenu</u>	Scene in Menu and in Game
<u>+ GameController gameController</u>	Gamecontroller

9.1.2 Methods

<u>+ void start(Stage stage)</u>	-Set background sound -Initialize Fields -Set scene -Set title game -Set resizable to false -Show the stage
<u>+ void main(String[] args)</u>	-Call Application.launch(args)