Specification of Dou Dizhu

Author: 李耀坤(49611200)

Time: 2019/5/13

version:1.0

Introduction to Dou Dizhu

Fight the Landlord (Dou DiZhu) is a poker game primarily for three players, but also playable by four. In each hand one player, the “landlord”, plays alone and the others form a team. The landlord’s aim is to be the first to play out all his cards in valid combinations, and the team wins if any one of them manages to play all their cards before the landlord. The game is said to have originated in Hubei province but is now popular all over China, and is also extensively played on line.

Dou dizhu is played among three people with one pack of cards, including the two differentiated jokers. The game starts with players bidding for the "landlord" (地主) position. Those who lose the bid or don't bid enter the game as the "peasants" (农民) team competing against the landlord. The objective of the game is to be the first player to have no cards left.

Environment

- Win10
- MATLAB R2019a

  Our system is implemented on Matlab R2019a and the picture below gives detail of the version of matlab/OS/Java. Please pay attention to that version lower than 9.6.0 may lead to unexpected result. We recommend to use our version as the plantform to development.

```
>> ver
-----------------------------------------------------------------------------------------------------
MATLAB 版本: 9.6.0.1072779 (R2019a)
MATLAB 许可证编号: 40504308
操作系统: Microsoft Windows 10 家庭中文版 Version 10.0 (Build 17134)
Java 版本: Java 1.8.0_181-b13 with Oracle Corporation Java HotSpot(TM) 64-Bit Server VM mixed mode
-----------------------------------------------------------------------------------------------------
MATLAB                                                  版本 9.6            (R2019a)
Simulink                                                版本 9.3            (R2019a)
Control System Toolbox                                  版本 10.6           (R2019a)
DSP System Toolbox                                      版本 9.8            (R2019a)
Data Acquisition Toolbox                                版本 4.0            (R2019a)
Image Processing Toolbox                                版本 10.4           (R2019a)
Instrument Control Toolbox                              版本 4.0            (R2019a)
Optimization Toolbox                                    版本 8.3            (R2019a)
Signal Processing Toolbox                               版本 8.2            (R2019a)
Simulink Control Design                                 版本 5.3            (R2019a)
Statistics and Machine Learning Toolbox                 版本 11.5           (R2019a)
Symbolic Math Toolbox                                   版本 8.3            (R2019a)
```

Specification

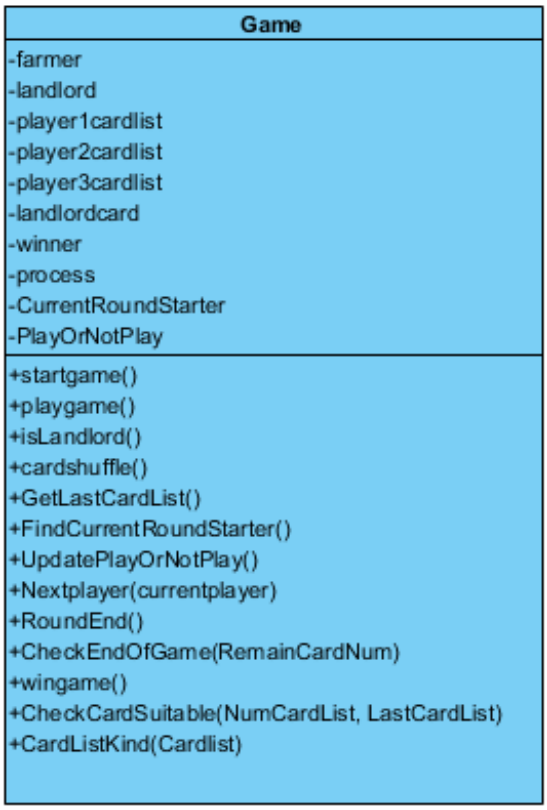
The UI

main UI

the player UI

The implementation of main classes

Game Class



The properties

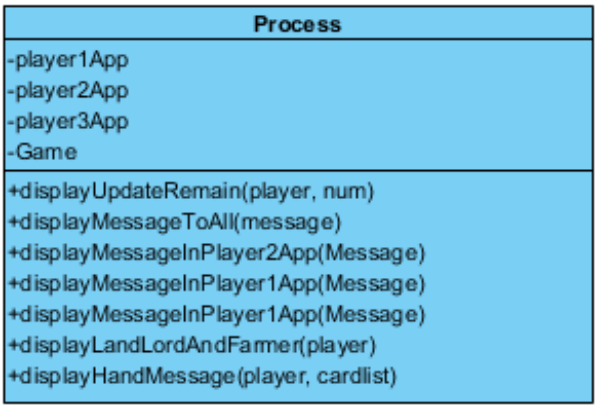
- $farmer$ / $landlord$: record which player is farmer / landlord
- $player1cardlist$ / $player2cardlist$ / $player3cardlist$: record the card of each player
- $landlordcard$: As we all known, the landlord has three more cards, and this variable records these three cards
- $winner$: the winner of this game
- $process$: a instance of Class Process
- $CurrentRoundStarter$: the starter of a round
- $PlayOrNotPlay$: it is a variable that's used to record the state of a round.

The method

- $startgame()$ is used to decide who will be the landlord. The algorithm of deciding who is the landlord is implemented in this function. startgame is the first phase of the game.
- $playgame()$: the process after $startgame$. After choosing the landlord, we begin to play. The basic framework of this function is a while loop whose condition is whether some player has no cards left. If the condition is satified, we jump to the next phase. Otherwise the players will go on to play more rounds. Basically, an iteration of the while loop is a round. The $CurrentRoundStarter$ plays the first hand, then the next player, then the one after. The variable $PlayOrNotPlay$ will record the state of a round, we judge whether the round is end according to this variable.

- $wingame()$: the process after $playgame$
- The remaining methods are all assistant functions which are used in $startgame()$, $playgame()$ or $wingame()$

Process Class



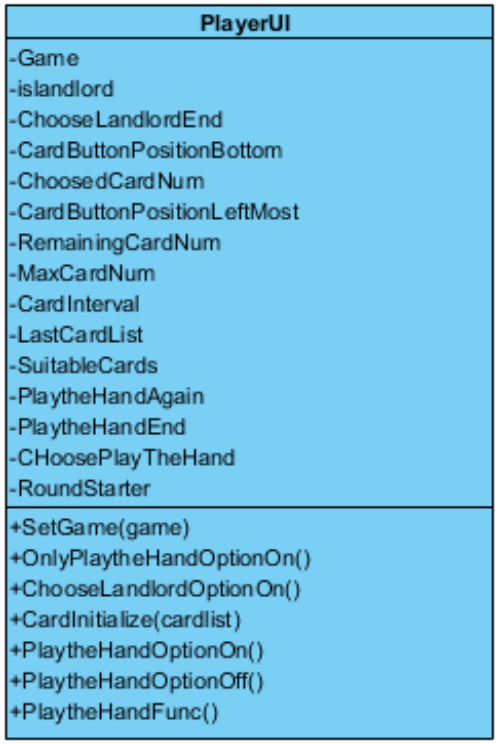
The properties

- $player1App$ / $player2App$ / $player3App$: the UI of the three players
- $Game$: an instance of class Game.

The methods

- $displayUpdateRemain()$: used to send and display how many cards the other two players is left
- $displayMessageToAll()$: send message to all players and display them on the UI.
- $displayMessageInPlayer1App()$ / $displayMessageInPlayer2App()$ / $displayMessageInPlayer3App()$: these three functions are used to send message to show in the UI
- $displayLandLordAndFamer()$: send message of landlord and farmer. We need to display who is the landlord on each player's UI and this function can display this information.
- $displayHandMessage()$: player need to be reminded the hand of last player so he can decide which hand to play.

PlayerUI Class



1. We need to get information of the cards player choose to play from the playerUI, therefore some of the functions and variables are used to record the card the player has selected to play.
2. On the other hand, we can only get the infromation of whether the player has finished the process of choosing cards from the playerUI, therfore some variables are used to record the state. And then these variables will be used in the Process Class.

Some Algorithm and Implementation Detail

The numerical value of every card

Char of the card	corresponding numerical value
3	3
4	4
5	5
6	6
7	7
8	8

Char of the card	corresponding numerical value
9	9
10	10
J	11
Q	12
K	13
A	14
2	15
小王	16
大王	17

Who is the landlord?

All players first review and appraise their own cards without showing their cards to the other players. The system will randomly choose one player then the player need to decide if he wants to be the landlord. After that the remaining two player take turn to decide if they want or not. The players who choose yes will be after again as order they are asked in the former round if more than one player wants to be the landlord.

This can ben implemented through basic loop and if-else statement.

How to judge whether a hand is appropriate?

In order to judge whether the hand is appropriate, we first need to give the kind of the hand.

kind	Description	number to represent the kind	example
solo	single card	1	3
pair	tow matching cards of equal rank	2	3-3
Trio	Three-of-a-kind:Three individual cards of the same rank	3	3-3-3
	Three cards of the same rank with a solo as the kicker	4	3-3-3-4

kind	Description	number to represent the kind	example
	Five consecutive individual cards	5	3-4-5-6-7
	Three consecutive pairs	6	3-3-4-4-5-5
Airplane	Two consecutive trios with each carries a distinct individual card as the kicker	7	3-3-3-4-4-4-5-6
Bomb	Four-of-a-kind, with a kicker. Four cards of the same rank without the kicker is called a bomb, which defies category rules, even beats four with a kicker.	10	3-3-3-3
	大王+小王	100	大王-小王

To judge which kind the hand is, we first sort the hand according to the numerical value. For example, if the hand is 10-10-10-J-J-J-Q-K, the corresponding numerical value of this hand is 10-10-10-11-11-11-12-13 (You can get this according to the graph above) we then sort this hand according to the numerical value list. Then get the length of that hand L, according to the length of the hand L, we are able to know the possible kind of this hand. Let take another example, the hand is 5-5-5-5, the length is 4 then we know that it can only be kind 10 (bomb) or kind 4 (three cards of the same rank with a solo as the kicker). Knowing the possible kinds we then test whether the hand is a valid one for every possible kind. If it is ,then it must be that kind. Otherwise, this hand is not valid, which means we should reject it.

For the first player of every round, he do not need to consider the hand of last player, so if his hand is valid, he can play it. However for players who play after the first one, the kind of their hand have to be consistent with last hand. We need to check it.

During the process of judging the kind, we will record the Key Card of the hand. Let use example to illustrate it. Two hand is 7-7-7-8-8-8-9-10 and 9-9-9-10-10-10-J-Q, to compare which is larger we actually only need to compare 7 with 9 if we are certain their type are the same. Here 7/9 is the Key Card of 7-7-7-8-8-8-9-10/9-9-9-10-10-10-J-Q. To compare two hands, we just comapre the Key Card of them.

How to judge whether a round is end ?

As we have mentioned before, we use $PlayOrNotPlay$ to record the state of a round and use this variable to judge whether the round is end. And it works as below:

It is a 3×1 vector. Any player choosing to play the hand(出牌) will reset the vector to [1;1;1]. if some player chooses not to play the hand, we will reset the vector according to the state of this vector. If the vector == [1;1;1], which means last player has choosed to play(出牌), we then reset the vector to [1;0;1]. If the vector is [1;0;0], which means last player selects to not to play, then the vector will be reseted to [1;0;0].

Our algorithm will check the vector after any player has choosed to play or not play. If the vector == [1;0;0] which means the end of the round.