

Xteam21 shouzhe li

1. **Title:** give your program a short descriptive and catchy name.

HuanLeDouDiZhu there is poker game called doudizhu in China

2. **Problem:** briefly describe a problem your program will help solve.

Players, Cards and Deal

The three-player game (from Baidu.com)

This game uses a 54-card pack including two jokers, red and black. The cards rank from high to low:

red joker, black joker, 2, A, K, Q, J, 10, 9, 8, 7, 6, 5, 4, 3.

Suits are irrelevant.

As in most Chinese games the cards are not really dealt, but taken from the deck by the players.

One of the players shuffles the cards, gives them to the player to his left to cut, and stacks them face down in the middle of the playing surface. One card is turned face up and inserted somewhere near the middle of the stack - this will determine who starts the auction. The dealer then draws the top card from the deck, looking at it but not showing it to the other players. The player to his right does the same, then the third player, then the dealer and so on counter-clockwise around the table until each player has 17 cards. The last three cards are left face down on the table until after the auction. It saves time if you look at your cards and sort them as you pick them up.

Auction

There is an auction to determine which player will be the landlord, and play alone against the other two. The possible bids are 1, 2 and 3. The player who drew the face up card in the "deal" is the first to bid. Each player in turn may either pass or bid higher than the highest bid so far. If everyone passes the hand is thrown in and there is a new deal. If there is a bid, the bidding continues counter-clockwise, each player passing or bidding higher than the previous bidder, until there are two consecutive players pass or someone bids 3, which ends the auction since it is the highest possible bid. The final and highest bidder is the landlord. This player now picks up the three face-down cards from the middle, for a total of 20 cards.

Play

The landlord plays first, and may play a single card or any legal combination. Each subsequent player in anticlockwise order must either pass (play no card) or beat the previous play by playing a higher combination of the **same number of cards** and **same type**. There are just two exceptions to this: a **rocket** can beat any combination, and a **bomb** can beat any combination except a higher bomb or rocket - see definitions below. The play continues around the table for as many circuits as necessary until two consecutive players pass. The played cards are then turned face down and put aside, and the person who played the last card(s) begins again, leading any card or legal combination.

In this game, there are thirteen types of combination that can be played:

1. **Single card** - ranking from three (low) up to red joker (high) as explained above
2. **Pair** - two cards of the same rank, from three (low) up to two (high)
3. **Triplet** - three cards of the same rank
4. **Triplet with an attached card** - a triplet with any single card added, for example 6-6-6-8. These rank according to the rank of the triplet - so for example 9-9-9-3 beats 8-8-8-A.
5. **Triplet with an attached pair** - a triplet with a pair added, like a full house in poker, the ranking being determined by the rank of the triplet - for example Q-Q-Q-6-6 beats 10-10-10-K-K.
6. **Sequence** - at least five cards of consecutive rank, from 3 up to ace - for example 8-9-10-J-Q. Twos and jokers cannot be used.
7. **Sequence of pairs** - at least three pairs of consecutive ranks, from 3 up to ace. Twos and jokers cannot be used. For example 10-10-J-J-Q-Q-K-K.
8. **Sequence of triplets** - at least two triplets of consecutive ranks from three up to ace. For example 4-4-4-5-5-5.
9. **Sequence of triplets with attached cards** - an extra card is added to each triplet. For example 7-7-7-8-8-8-3-6. The attached cards must be different from all the triplets and from each other. Although triplets of twos cannot be included, a two or a joker or one of each can be attached, but not both jokers.
10. **Sequence of triplets with attached pairs** - an extra pair is attached to each triplet. Only the triplets have to be in sequence - for example 8-8-8-9-9-9-4-4-J-J. The pairs must be different in rank from each other and from all the triplets. Although triplets of twos cannot be included, twos can be attached. Note that attached single cards and attached pairs cannot be mixed - for example 3-3-3-4-4-4-6-7-7 is not valid.
11. **Bomb** - four cards of the same rank. A bomb can beat everything except a rocket, and a higher ranked bomb can beat a lower ranked one.
12. **Rocket** - a pair of jokers. It is the highest combination and beats everything else, including bombs.
13. **Quadplex set** - there are two types: a quad with two single cards of different ranks attached, such as 6-6-6-6-8-9, or a quad with two pairs of different ranks attached, such as J-J-J-J-9-9-Q-Q. Twos and jokers can be attached, but you cannot use both jokers in one quadplex set. Quadplex sets are ranked according to the rank of the quad. Note that a quadplex set can only beat a lower quadplex set of the same type, and cannot beat any other type of combination. Also a quadplex set can be beaten by a bomb made of lower ranked cards.

Note that passing does not prevent you from playing on a future turn.

Example Player A (the landlord) leads 3-3-3-9 to get rid of some low cards, player B passes, player C plays 5-5-5-7, player A plays K-K-K-J and player B plays A-A-A-3. C and A pass, so B can start again with anything. He leads a single 4.

Note B could have played his aces on his the first turn, but preferred to pass to give his partner a chance to get rid of some cards. C will now play if possible, so as not to give the landlord (A) a free chance to lead again. Having beaten A's second play, B leads a low card to give C the choice of playing another unwanted card or putting the landlord under pressure by playing a high card.

Scoring// skip it ! I wont use this part

If the landlord runs out of cards first he has won, and each opponent pays him the amount of the bid - 1, 2 or 3 units - provided that no bomb or rocket was played. If one of the other two players runs out before the landlord, the landlord loses and must pay the amount of the bid **to each opponent**. For each occasion when any player played a bomb or rocket, the payment for the hand is doubled. So for example in a hand in which two bombs and a rocket were played, a player who bid 3 will win 24 points from each opponent for going out first, or pay 24 to each opponent if another player goes out first.

Note that since the opponents of the landlord stand to win or lose equally, they form a temporary partnership. When playing against the landlord it is just as profitable to help your partner to run out of cards first as to win yourself. Because of this the partners will usually not beat each other's cards, and the weaker partner will play to help the stronger partner.

3. **Primary stakeholder:** describe or explain who will be the primary user of the program and what the program will do or how the stakeholder will benefit from using this program. For example: will students use it, instructors, customers, vendors, kids (games-[G rating \(Links to an external site.\)](#) only), etc...

this is a game

4. **Graphical User interface:** Provide a set sketches or written description of (1 to 3) menus, forms, etc.. for the team to understand or see how the user will interact with the program and what output will be produced. Later in the course, we will learn how to make GUIs in Java. You can either draw a sketch by hand (and scan it) or you can use graphics software.

The rules of the game is in the problem section.

what we need to perform is the identity of farmers or landlord,

two AI players and cards in your hand

5. **Data:** What type of data is needed (to be read into your program from a file) for your program to produce results? What abstract data types will you use to store the inputted data to best provide the needed access for the program's operations? Providing sample data records in the format that you expect your program to handle is a great way to make your idea more clear.

variables:

two AI players

- contains variable identity**

- contains card array method**

- contains giveCard method**

 - givecards according to the rule normally from the smallest**

 - perhaps gonna use legalPlay method**

card array

- a ordered array with {red joker, black joker, 2, A, K, Q, J, 10, 9, 8, 7, 6, 5, 4, 3. }**

console(main method could be here)

- contains current cards array, current cards in the console**

- contains auction method**

 - play method (perhaps gonna use scanner here)**

MyPlay(perhaps implements AI players)

- contains a current card array and display to screen**