Idea: CardsGame: Blackjack (Twenty-One)

Net ID：kz2193　　Name: Kexin Zhou

Game description: Blackjack is one of the most widely played casino banking game in the world, it uses decks of 52 cards and descends from a global family of casino banking games known as Twenty-One. Blackjack players do not compete against each other. The game is a comparing card game where each player competes against the dealer.

1. Advanced concepts：

   UI: game page
   Network: multiple players could join and play the game at the same time on the network
   Multi-thread: each thread attaches to one player

2. Roles in the game: dealer(server) and player*3(client)

3. Rules of Game:

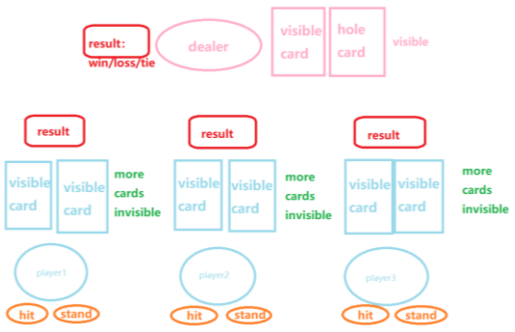
blackjack= A +J/Q/K
each player competes against the dealer
//specific rules of the game listed in another submitted file: rules of blackjack.pdf

4. Plan to realize

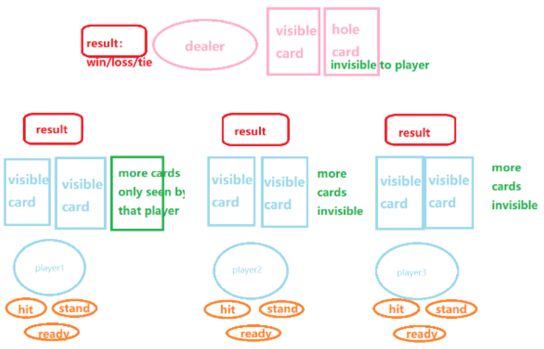
1) Design UI for players to take in the behaviors from players, Eg. Join, ready, hit, stand in the game, UI also reveal the current status of players
2) Use network to connect players and dealer and let them exchange message and play together
3) Use multithread to allow 3 players deal with one dealer one by one in a game
4) The behavior of dealer will be controlled by my program not a real person, who will randomly draw cards and send cards to other players
5) Players' decision in game will be processed in my program and give them feedback in result box on UI.

5. UI design: (about the structure)

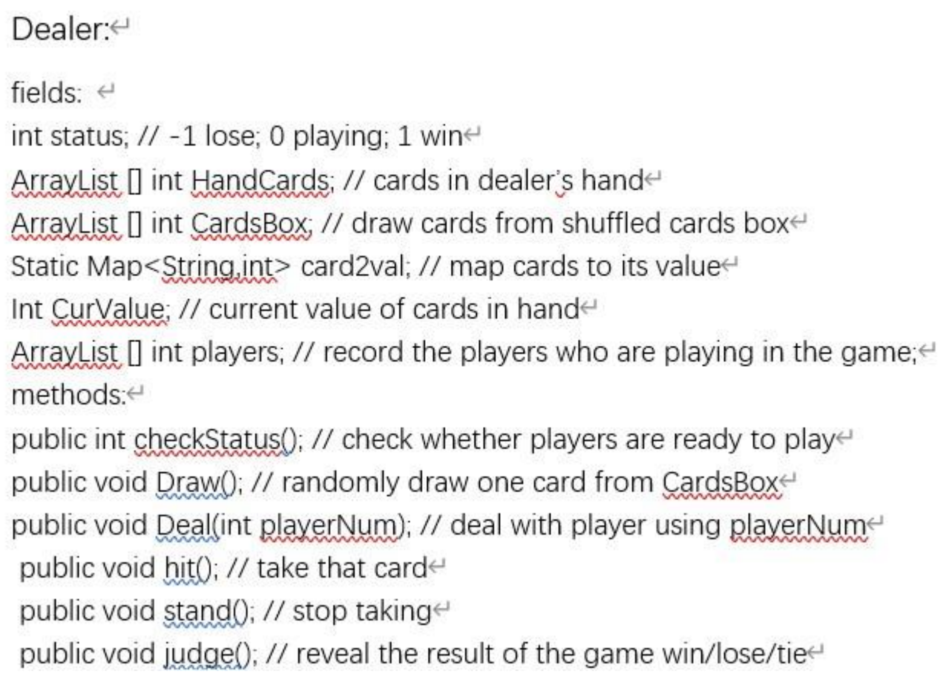
From dealer aspect:



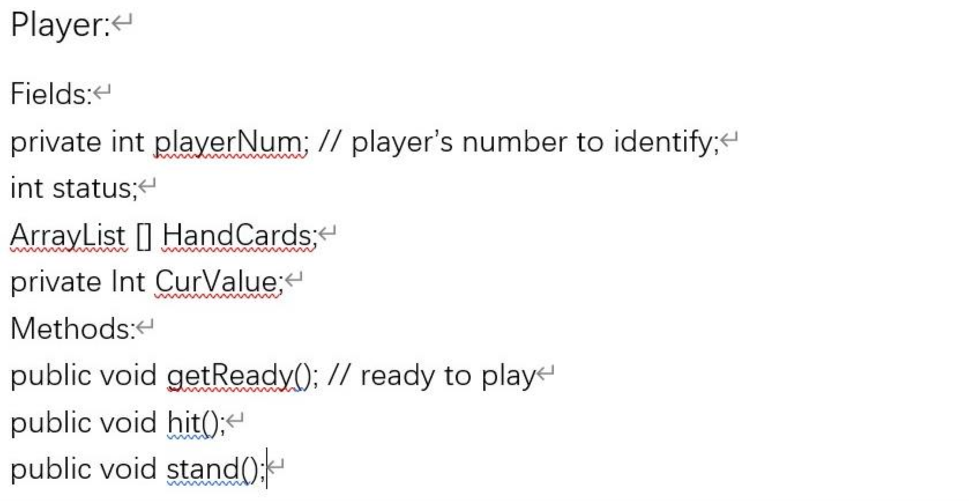
From player1 aspect:



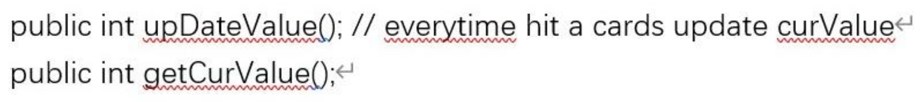
6. Class design(roughly)



```
Dealer:
fields:
int status; // -1 lose; 0 playing; 1 win
ArrayList [] int HandCards; // cards in dealer's hand
ArrayList [] int CardsBox; // draw cards from shuffled cards box
Static Map<String,int> card2val; // map cards to its value
Int CurValue; // current value of cards in hand
ArrayList [] int players; // record the players who are playing in the game;
methods:
public int checkStatus(); // check whether players are ready to play
public void Draw(); // randomly draw one card from CardsBox
public void Deal(int playerNum); // deal with player using playerNum
 public void hit(); // take that card
 public void stand(); // stop taking
 public void judge(); // reveal the result of the game win/lose/tie
```



```
Player:
Fields:
private int playerNum; // player's number to identify;
int status;
ArrayList [] HandCards;
private Int CurValue;
Methods:
public void getReady(); // ready to play
public void hit();
public void stand();
```



```
public int upDateValue(); // everytime hit a cards update curValue
public int getCurValue();
```