Project ScreenShots

This is my shuffle method. It finds a random int within range of the arraylist indices, then swaps the indices.

```
//shuffle methods

public static void shuffleList(ArrayList<Card> a) {
      int n = a.getSize();
      Random random = new Random();
      random.nextInt();
      for (int i = 0; i < n; i++) {
            int change = i + random.nextInt( bound: n - i);
            swap(a, i, change);
      }
      System.out.println("shuffled: " + protoDeck + "\n");
}

private static void swap(ArrayList<Card> a, int i, int change) {
      //  int helper = a.get(i);
      Card helper = a.get(i);  // needs to return int
      a.set(i, a.get(change));
      a.set(change,(helper));   //set accepts index, card
}
```

It produces this:

```
shuffled:  FOUR of SPADES
 | NINE of HEARTS | SIX of DIAMONDS | FIVE of CLUBS | FIVE of DIAMONDS
 | FOUR of HEARTS | SEVEN of SPADES | JACK of CLUBS | FOUR of CLUBS
 | EIGHT of SPADES | ACE of HEARTS | ACE of SPADES | TWO of CLUBS
 | ACE of DIAMONDS | TWO of HEARTS | SIX of HEARTS | TEN of CLUBS
 | QUEEN of DIAMONDS | THREE of CLUBS | SEVEN of DIAMONDS | NINE of DIAMONDS
 | TEN of DIAMONDS | TWO of SPADES | FIVE of SPADES | EIGHT of HEARTS
 | JACK of DIAMONDS | KING of DIAMONDS | SIX of CLUBS | KING of SPADES
 | FOUR of DIAMONDS | KING of CLUBS | EIGHT of CLUBS | JACK of HEARTS
 | TWO of DIAMONDS | SIX of SPADES | THREE of HEARTS | EIGHT of DIAMONDS
 | THREE of DIAMONDS | JACK of SPADES | SEVEN of HEARTS | TEN of SPADES
 | SEVEN of CLUBS | QUEEN of SPADES | QUEEN of HEARTS | ACE of CLUBS
 | KING of HEARTS | QUEEN of CLUBS | TEN of HEARTS | NINE of CLUBS
 | FIVE of HEARTS | NINE of SPADES | THREE of SPADES |
```

Then we begin the game.

```
Let's play a game.
Player 2 has been randomly selected to go first.
-----Round 1:

Player1's hand:  FOUR of SPADES   |   NINE of HEARTS   |   SIX of DIAMONDS   |   FIVE of CLUBS   |   FIVE of DIAMONDS   |   FOUR o
Player2's hand:  ACE of DIAMONDS   |   TWO of HEARTS   |   SIX of HEARTS   |   TEN of CLUBS   |   QUEEN of DIAMONDS   |   THREE of
Player3's hand:  KING of DIAMONDS   |   SIX of CLUBS   |   KING of SPADES   |   FOUR of DIAMONDS   |   KING of CLUBS   |   EIGHT o
Player4's hand:  SEVEN of HEARTS   |   TEN of SPADES   |   SEVEN of CLUBS   |   QUEEN of SPADES   |   QUEEN of HEARTS   |   ACE of

table 1:    |   FOUR of SPADES   |   ACE of DIAMONDS   |   KING of DIAMONDS   |   SEVEN of HEARTS

Winner: KING of DIAMONDS
```

Player is randomly selected to go first.
Everyone puts down their top cards on the table
Winner is determined by this complicated if statement I wrote in the menuClient class and this equalsTo() method I included in the Card class

```
public static Card findWinner(Card card1, Card card2, Card card3, Card card4) {
      Card winningCard = null;
      //4 > 3
      if (Card.equalsTo(card1, card2)) { //card 1 > 2
            if (Card.equalsTo(card1, card3)) { //card 1 > 3
                  if (Card.equalsTo(card1, card4)) {//card 1 > all
                        return winningCard = card1;
                  } else return winningCard = card4; //card 4 > all
            } else if (Card.equalsTo(card3, card4)) { // card 3 > all
                  return winningCard = card3;
            } else return winningCard = card4;
      }else if (Card.equalsTo(card2, card3)) { //card 2 > 3
            if (Card.equalsTo(card2, card4)){ //card 2 > 4
                  return winningCard = card2;
            } else return winningCard = card4;
      } else if (Card.equalsTo(card3, card4)) { //card 3 is greater
            return winningCard = card3;
      } else return winningCard = card4;

}
```

```
public static Boolean equalsTo(Card p, Card o) {
      if (p.getCardValue().equals(o.getCardValue())){  //cards are the same number
            if(p.cardSuit.getCardSuit() > (o.cardSuit.getCardSuit())) {  //compare cardSuits p>o
                  return true;
            } else return false;  //o > p
      } else if (p.cardValue.getCardValue() > (o.cardValue.getCardValue())) { // p>o card number is
            return true;
      } else return false;  //o > p
}



public String toString() { return cardValue + " of " + cardSuit /* + "(" + cardValue() + ")"*/; }
```

My code only goes to round 5 because I ran out of time, but everything works as it's supposed to. At the end of each round the winner keeps all the cards, they're added to the end of their deck and then it continues like that.
Winners:

```
Let's play a game.
Player 2 has been randomly selected to go first.
-----Round 1:

Player1's hand:  FOUR of SPADES   |   NINE of HEARTS   |   SIX of DIAMONDS   |   FIVE of CLUBS   |   FIVE of DIAMONDS   |
Player2's hand:  ACE of DIAMONDS   |   TWO of HEARTS   |   SIX of HEARTS   |   TEN of CLUBS   |   QUEEN of DIAMONDS   |   TH
Player3's hand:  KING of DIAMONDS   |   SIX of CLUBS   |   KING of SPADES   |   FOUR of DIAMONDS   |   KING of CLUBS   |
Player4's hand:  SEVEN of HEARTS   |   TEN of SPADES   |   SEVEN of CLUBS   |   QUEEN of SPADES   |   QUEEN of HEARTS   |

table 1:    |   FOUR of SPADES   |   ACE of DIAMONDS   |   KING of DIAMONDS   |   SEVEN of HEARTS

Winner: KING of DIAMONDS

-----Round 2:

Player1's hand:  NINE of HEARTS   |   FIVE of CLUBS   |   FIVE of DIAMONDS   |   FOUR of HEARTS   |   SEVEN of SPADES   |
Player2's hand:  TWO of HEARTS   |   TEN of CLUBS   |   QUEEN of DIAMONDS   |   THREE of CLUBS   |   SEVEN of DIAMONDS   |
Player3's hand:  SIX of CLUBS   |   FOUR of DIAMONDS   |   KING of CLUBS   |   EIGHT of CLUBS   |   JACK of HEARTS   |   TWO
Player4's hand:  TEN of SPADES   |   QUEEN of SPADES   |   QUEEN of HEARTS   |   ACE of CLUBS   |   KING of HEARTS   |   QUE

table 2:    |   NINE of HEARTS   |   TWO of HEARTS   |   SIX of CLUBS   |   TEN of SPADES

Winner: TEN of SPADES

-----Round 3:
```

Added to their queues


ACE of SPADES | TWO of CLUBS
| EIGHT of HEARTS | JACK of DIAMONDS
| THREE of DIAMONDS | JACK of SPADES | FOUR of SPADES | ACE of DIAMONDS | KING of DIAMONDS | SEVEN of HEARTS
NE of SPADES | THREE of SPADES
TWO of CLUBS
S | JACK of DIAMONDS
DS | JACK of SPADES | FOUR of SPADES | ACE of DIAMONDS | KING of DIAMONDS | SEVEN of HEARTS
HREE of SPADES | NINE of HEARTS | TWO of HEARTS | SIX of CLUBS | TEN of SPADES