src\BlackjackPanel1.java

```
1  import java.awt.*;
2  import javax.swing.*;
3   import java.awt.event.KeyEvent;
4   import java.awt.event.KeyListener;
5   import java.util.ArrayList;
6   import java.util.List;
7
8   public class BlackjackPanel1 extends JPanel implements KeyListener{
9
10     String state;
11     String winner;
12     char userInput;
13     Deck deck;
14     BlackjackPlayer player1;
15     BlackjackPlayer dealer;
16     final int CARD_WIDTH = 128;
17     final int CARD_HEIGHT = 186;
18     final int FRAME_RATE = 100;
19     private List<Card> discard;
20     final int NEED_SHUFFLE = 15;
21     final int EXACT_SCORE = 21;
22
23     public BlackjackPanel1(){
24         setPreferredSize(new Dimension(800,800));
25         setBackground(new Color(2,108,26));
26         addKeyListener(this);
27         state = "READY";
28         winner = " ";
29         userInput = '-';
30         deck = new Deck();
31         player1 = new BlackjackPlayer("Soumya");
32         dealer = new BlackjackPlayer("Disha");
33         deck.shuffle();
34         discard = new ArrayList<Card>();
35     }
36
37     @Override
38     public void paintComponent(Graphics g) {
39         super.paintComponent(g);
40         g.setColor(Color.WHITE);
41         g.setFont(new Font("Britannic Bold", Font.BOLD, 40));
42         g.drawString("BLACKJACK!", 300, 40);
43         int playerX = 0;
44         for(int i = 0; i < player1.seeCards().size(); i++) {
45             playerX += 50;
46             Card c = player1.seeCards().get(i);
47             g.drawImage(c.getFace(), playerX, 50, CARD_WIDTH, CARD_HEIGHT, null);
48         }
```

```
49          int dealerX = 0;
50          for(int i = 0; i < dealer.seeCards().size(); i++) {
51              dealerX += 50;
52              Card c = dealer.seeCards().get(i);
53              g.drawImage(c.getFace(), dealerX, 300, CARD_WIDTH, CARD_HEIGHT, null);
54          }
55          if(state.equals("READY")) {
56              g.drawString("Press d to deal", 300, 350);
57          }
58          if(state.equals("SHOW")) {
59              g.drawString(winner, 250, 290);
60
61          }
62      }
63
64      public void run() {
65          while(true) {
66              update();
67              repaint();
68              delay(FRAME_RATE);
69          }
70      }
71
72      private void update() {
73          if (state.equals("READY")){
74              if(userInput == 'd') {
75                  userInput = '-';
76                  state = "SHUFFLE AND DEAL";
77              }
78          } else if (state.equals("SHUFFLE AND DEAL")){
79              player1.take(deck.deal());
80              player1.take(deck.deal());
81              dealer.take(deck.deal());
82              state = "P1_TURN";
83          } else if (state.equals("P1_TURN")){
84              if(player1.getScore() > EXACT_SCORE) {
85                  state = "SHOW";
86              }
87              if(userInput == 'h') {
88                  userInput = '-';
89                  player1.take(deck.deal());
90              }
91              if(userInput == 's') {
92                  userInput = '-';
93                  state = "DEALER TURN";
94              }
95          } else if (state.equals("DEALER TURN")){
96              dealer.take(deck.deal());
97                  while(dealer.getScore() <= 16) {
98                  dealer.take(deck.deal());
```

```
            }
            if(dealer.getScore() > EXACT_SCORE || userInput == 's') {
                state = "SHOW";
            }
        } else if(state.equals("SHOW")) {
            if (player1.getScore() > EXACT_SCORE) {
                winner = "DEALER WON";
            } else if (dealer.getScore() > EXACT_SCORE) {
                winner = "PLAYER WON";
            } else if (player1.getScore() == dealer.getScore()) {
                winner = "THERE WAS A TIE";
            } else if (player1.getScore() == EXACT_SCORE) {
                winner = "PLAYER WON";
            } else if (dealer.getScore() == EXACT_SCORE) {
                winner = "DEALER WON";
            } else if (player1.getScore() > dealer.getScore()) {
                winner = "PLAYER WON";
            } else {
                winner = "DEALER WON";
            }
            if(userInput == 'c') {
                userInput = '-';
                state = "CLEAR";
            }
        } else if(state.equals("CLEAR")) {
            discard.addAll(player1.fold());
            discard.addAll(dealer.fold());
            if(deck.getCardsLeft() < NEED_SHUFFLE) {
                deck.addCardsBack(discard);
                discard.clear();
                deck.shuffle();
            }
            state = "READY";
        } else {
            throw new RuntimeException("Unkown state: " + state);
        }
    }

    public void keyPressed(KeyEvent e) {

    }
    public void keyReleased(KeyEvent e) {

    }
    public void keyTyped(KeyEvent e) {
        userInput = e.getKeyChar();
    }

    private void delay(int milliseconds) {
        try {
```

```
149         Thread.sleep(milliseconds);
150         } catch (Exception e) {
151         System.out.println(e);
152         }
153     }
154 }
155
156
157
```