# Blackjack

by Berke Altıparmak, Can İvit, Mehmet Naci Keskin, Yiğit Kılıçoğlu
AP Programming 2019-2020
Second Semester
Midterm 1

**Work Distribution (Who did what)**

As a group we decided that the first thing we should do is to create a complete and flawless console game. Then we would transfer our robust game code to a nice graphical user interface. Finally we would bring networking in to add multiplayer functionality over TCP/IP to our game. Since the complete development of the game would involve lots of steps and work of different programmers (us), we knew we must follow a highly object oriented approach to make this process as efficient as possible.

Mehmet Naci Keskin: He wrote the Card, Hand, Deck, and Player classes. He made sure the methods of these fundamental classes work as intended and are easy to use because other group members use these classes to develop the actual game.

Berke Altıparmak: He read the rules of Blackjack over and over again to write the ConsoleGame class that holds the base algorithm of the game. Since he thinks a game is not perfect without a great introduction, he also wrote the IntroFrame class that attracts users with graphical transitions.

Yiğit Kılıçoğlu: He knew a Blackjack game needs to have nice jazz music playing smoothly in the background. That's why he wrote the MusicPlayer class that can play a sound file continuously. He found suitable chip, card, and background images for the graphical user interface of the game. Then he provided example codes on how to scale and place these images inside JLabels, how to place those Jlabels on anywhere on a JPanel, and how to add mouse listeners to those JLabels to get user input.

Can Ivit: He wrote the MainGamePanel, the graphical user interface version of the game, by combining the example codes Yiğit provided and the base game algorithm in ConsoleGame class.

**Operating "Manual"**

1) Open "Main.java", compile and run it.
2) Input the number of players and usernames.
3) The players will then make their bets by clicking on the chips.
4) Cards will be distributed to every player. The game starts. Players make moves until every player and the dealer stand or bust.

**Classes and their explanations:**

| Class Name | Description |
|---|---|
| Card.java | Creates a standard playing card. Stores its type, suit and whether it is face up. |
| Deck.java | Creates standard 52-card decks with four suits: clubs, diamonds, hearts and spades. Allows the simultaneous creation of decks. |
| Hand.java | Stores the cards of a player. Calculates the value of the hand as well. |
| Player.java | Stores the name, hand, money and bet of a player. |
| GameRules.java | Stores constants related to the board that the user selects in the beginning. The deck count and the minimum & maximum bet amounts |
| BankPanel.java | Lets the user make bets. Shows the money of the player and lets the user select chips one by one. |
| PlayPanel.java | Users give the "hit", "stand" and "double" command via this panel. |
| MainGamePanel.java | Displays the blackjack board and hosts the game. Cards and chips are put on this panel. The bank panel and the playing panel are utilized by this class. |
| MusicPlayer.java | Continuously plays jazz music. |
| IntroFrame.java | Displays the intro animations, including the "Want A Punch? Studios" logo and the blackjack wallpaper. |
| MainGameFrame.java | Combines the MainGamePanel with MusicPlayer to create a game with background music. |

**Resources (Citations)**

MusicPlayer.java
https://www.geeksforgeeks.org/play-audio-file-using-java/
IntroFrame.java
https://docs.oracle.com/javafx/2/get_started/jfxpub-get_started.htm
https://examples.javacodegeeks.com/desktop-java/javafx/javafx-animation-example/

**Imported Libraries**

javax.swing.JPanel;
javax.swing.JLabel;
javax.swing.ImageIcon;
javax.swing.SwingConstants;
javax.swing.JOptionPane;

```
javax.imageio.ImageIO;
java.awt.image.BufferedImage;
java.awt.Image;

java.awt.Graphics;
java.awt.Graphics2D;
java.awt.BorderLayout;
java.awt.GridLayout;
java.awt.Color;
java.awt.Font;

java.awt.event.MouseListener;
java.awt.event.MouseEvent;
java.awt.event.ComponentListener;
java.awt.event.ComponentEvent;
java.awt.event.ActionListener;

java.io.File;
java.io.IOException;

java.util.HashMap;
java.util.Stack;
java.util.ArrayList;
java.util.Scanner;
java.util.Random;

javafx.stage.Stage;
javafx.stage.Modality;
javafx.scene.Scene;
javafx.scene.Group;
javafx.scene.layout.HBox;
javafx.scene.paint.Color;
javafx.geometry.Insets;
javafx.geometry.Pos;
javafx.animation.Timeline;
javafx.event.ActionEvent;
javafx.event.EventHandler;
javafx.beans.InvalidationListener;
javafx.beans.Observable;
javafx.scene.shape.Path;
javafx.scene.shape.MoveTo;
javafx.scene.shape.HLineTo;
javafx.scene.shape.VLineTo;
javafx.util.Duration;
javafx.scene.image.Image;
javafx.scene.image.ImageView;
javafx.application.Application;
javafx.scene.layout.Pane;
javax.sound.sampled.AudioInputStream;
javax.sound.sampled.AudioSystem;
javax.sound.sampled.Clip;
javax.sound.sampled.LineUnavailableException;
javax.sound.sampled.UnsupportedAudioFileException;
```