PROJECT 2

A GAME OF SIMPLE ROULETTE

CSCI 145
JAVA LANGUAGE

MAI PHAM

DEVELOPMENT ENVIRONMENT
ECLIPSE

TABLE OF CONTENTS

Mai Pham
CSCI 145 - Project 2
Prof. T. Vo

PROJECT NOTE

Objective – Project 2 is developed an application that uses classes and objects to create a simple roulette game. This roulette game has 12 positions (from number 0 to 10 and 00). A player gets to bet on black (even number), red (odd number), or an actual number. If a bet on color and win, the player wins double the amount of bet. If a bet on actual number and win, the player wins 10 times the amount of bet. However, if the wheel spin on 0 or 00, which is color green, the player automatically loses. The game should hold two players, and if one quit, the other player should still be able to continuous playing. At the end of the game, it should conclude each player's amount of money win/lose.

Summary – The project come with 3 files (Roulette.java, Player.java, and Wheel.java) that already have the skeleton of the game. Therefore, there isn't much work on designing the project. I just need to enhance the program base on the requirements and instructions. Majority of this project is spent on testing the program. I tested the provided files before I started the project and continuously testing it throughout the coding period. Once finished, I had to keep testing to make sure my program does all the mathematic correctly, able to handle all requirements, and doesn't crash anywhere in different circumstance.

Extra credits – There are two extra credits in this project. The first one is to keep track of the money win/lose by the house. This extra credit is fairly simple because in the main project, we have similar situation where we need to keep track of the money win/lose per player. The second extra credit is being able to speed up the testing process using alterative way than input through keyboard. To input by file using Scanner class, the coding need to be change. Therefore, I picked the method of redirection technique. This technique only requires to create a text file and input that file in the run configuration.

Conclusion – As a conclusion, I am able to get my project finished and successfully created a roulette game. I'm also able to do both extra credits and applied them to my main project. From this project, I learned how to use 3 classes together instead of just 2 classes like how we usually do for class's assignments. I also learned how static methods work differently compare to the class contain main method or constructor method. For static classes, there is no need to create an object. The 3rd thing I learned from this project is redirection technique to speed up my testing process.

PROJECT Q & A

Question #6:
Note that some issues, like the bet options, are presented by the Wheel class, while others, like determining the number to bet, are handled by the Player class. Why is that?
Answer:
Divide methods/issues into different classes is mainly based on personal choices and reference. The roulette game could still possibly work with just one class. However, dividing into multiple classes is better as the application get bigger and bigger. This allows the application to be more readable, allow reusable, easier testing, better code maintaining, and better project structure. When it comes to divide classes, it focuses on organization and logic concept. Therefore, it makes more sense to keep the Wheel class focus on spinning, determine the number and color, etc. Meanwhile, the Player class is focus on recording per player's bets and their money. It is not technical wrong if we do some change to it but it wouldn't seem reasonable to other people when they need to look through our code or reuse it for other applications.

Question #13
How could we modify the program to facilitate the testing of this issues while still demonstrating confidence in the program's logic?
Answer:
It is true that the chances of betting on a number and winning that bet are relatively small due to the fact that the roulette wheel has 12 positions. In order to test the program to ensure it works correctly, I change the constant variable MAX_NUM to 3. This way, the chances of betting on the winning number is much higher.

Question
Which class needs to be modified to accommodate multiple games? Explain.
Answers:
I think in order to accommodate multiple games, I would modify the Roulette class. Because, I would still keep the Wheel class handle the spinning and determine the wining/losing amount while the Player class as individual player recording their money. This leave with the Roulette class to handle multiple games with multiple players. So, the Roulette class just need to call the Wheel class multiple time as there are multiple players playing spontaneously.

OUTPUT FILE

TEST CASE 1: Validate bet amounts, bet options, and bet numbers.

```
A Simple Roulette Game
By Mai Pham

Welcome to a simple version of roulette game.
You can place a bet on black, red, or a number.
A color bet is paid 2 the bet amount.
A number bet is paid 10 the bet amount.
You can bet on a number from 1 to 10.
Gamble responsibly.  Have fun and good luck!

Money available for Jane: 100
How much to bet: -5                              //validate bet amounts
The amount is invalid.
Please enter the amount again: 109
The amount is invalid.
Please enter the amount again: 30
Betting Options:
    1. Bet on black (even numbers)
    2. Bet on red (odd numbers)
    3. Bet on a number between 1 and 10
Please enter a betting option: 0                 //validate bet options
The betting option is invalid.
Please enter betting option again: 4
The betting option is invalid.
Please enter betting option again: 1

Money available for Jerel: 100
How much to bet: 50
Betting Options:
    1. Bet on black (even numbers)
    2. Bet on red (odd numbers)
    3. Bet on a number between 1 and 10
Please enter a betting option: 3
Please enter a number: 0                         //validate bet numbers
The number is invalid.
Please enter a number again: 13
The number is invalid.
Please enter a number again: 8

The number is: 3
The color is: Red

The amount Jane win is: 0
Play again [y/n]? n

The amount Jerel win is: 0
Play again [y/n]? n

The total amount winning/losing by Jane is: -30
The total amount winning/losing by Jerel is: -50
The total amount winning/losing by the house is: 80
```

```
Game over!  Thanks for playing.
```

TEST CASE 2: Show one player can still play when the other play already quit, player automatically gets $100 when current money ran out and player wishes to continue, and players automatically lose on 0 and 00.

```
A Simple Roulette Game
By Mai Pham

Welcome to a simple version of roulette game.
You can place a bet on black, red, or a number.
A color bet is paid 2 the bet amount.
A number bet is paid 10 the bet amount.
You can bet on a number from 1 to 10.
Gamble responsibly.  Have fun and good luck!

Money available for Jane: 100
How much to bet: 30
Betting Options:
    1. Bet on black (even numbers)
    2. Bet on red (odd numbers)
    3. Bet on a number between 1 and 10
Please enter a betting option: 3
Please enter a number: 2

Money available for Jerel: 100
How much to bet: 25
Betting Options:
    1. Bet on black (even numbers)
    2. Bet on red (odd numbers)
    3. Bet on a number between 1 and 10
Please enter a betting option: 1

The number is: 0                                    //player lose on 0 and 00
The color is: Green

The amount Jane win is: 0
Play again [y/n]? n

The amount Jerel win is: 0
Play again [y/n]? y

Money available for Jerel: 75                       //the game continuous with 1 player
How much to bet: 75
Betting Options:
    1. Bet on black (even numbers)
    2. Bet on red (odd numbers)
    3. Bet on a number between 1 and 10
Please enter a betting option: 1

The number is: 1
The color is: Red

The amount Jerel win is: 0
Play again [y/n]? y

Money available for Jerel: 100                      //obtain $100 when balance is 0
```

```
How much to bet: 50
Betting Options:
    1. Bet on black (even numbers)
    2. Bet on red (odd numbers)
    3. Bet on a number between 1 and 10
Please enter a betting option: 2

The number is: 6
The color is: Black

The amount Jerel win is: 0
Play again [y/n]? n

The total amount winning/losing by Jane is: -30              //amount Jane lose
The total amount winning/losing by Jerel is: -150            //amount Jerel lose
The total amount winning/losing by the house is: 180         //amount house win

Game over!  Thanks for playing.
```

TEST CASE 3: Show correct win/lose amount.

```
A Simple Roulette Game
By Mai Pham

Welcome to a simple version of roulette game.
You can place a bet on black, red, or a number.
A color bet is paid 2 the bet amount.
A number bet is paid 10 the bet amount.
You can bet on a number from 1 to 10.
Gamble responsibly.  Have fun and good luck!

Money available for Jane: 100
How much to bet: 15
Betting Options:
    1. Bet on black (even numbers)
    2. Bet on red (odd numbers)
    3. Bet on a number between 1 and 10
Please enter a betting option: 3
Please enter a number: 2

Money available for Jerel: 100
How much to bet: 20
Betting Options:
    1. Bet on black (even numbers)
    2. Bet on red (odd numbers)
    3. Bet on a number between 1 and 10
Please enter a betting option: 3
Please enter a number: 5

The number is: 4
The color is: Black

The amount Jane win is: 0
Play again [y/n]? y

The amount Jerel win is: 0
Play again [y/n]? y
```

```
Money available for Jane: 85
How much to bet: 30
Betting Options:
    1. Bet on black (even numbers)
    2. Bet on red (odd numbers)
    3. Bet on a number between 1 and 10
Please enter a betting option: 3
Please enter a number: 6                        //payoff 10 times on number

Money available for Jerel: 80
How much to bet: 40
Betting Options:
    1. Bet on black (even numbers)
    2. Bet on red (odd numbers)
    3. Bet on a number between 1 and 10
Please enter a betting option: 2

The number is: 6
The color is: Black

The amount Jane win is: 300                     //bet 30, win 300
Play again [y/n]? y

The amount Jerel win is: 0
Play again [y/n]? y

Money available for Jane: 355
How much to bet: 55
Betting Options:
    1. Bet on black (even numbers)
    2. Bet on red (odd numbers)
    3. Bet on a number between 1 and 10
Please enter a betting option: 1                //payoff 2 times on color

Money available for Jerel: 40
How much to bet: 10
Betting Options:
    1. Bet on black (even numbers)
    2. Bet on red (odd numbers)
    3. Bet on a number between 1 and 10
Please enter a betting option: 2

The number is: 6
The color is: Black

The amount Jane win is: 110                     //bet 55, win 110
Play again [y/n]? y

The amount Jerel win is: 0
Play again [y/n]? n

Money available for Jane: 410
How much to bet: 3
Betting Options:
    1. Bet on black (even numbers)
    2. Bet on red (odd numbers)
```

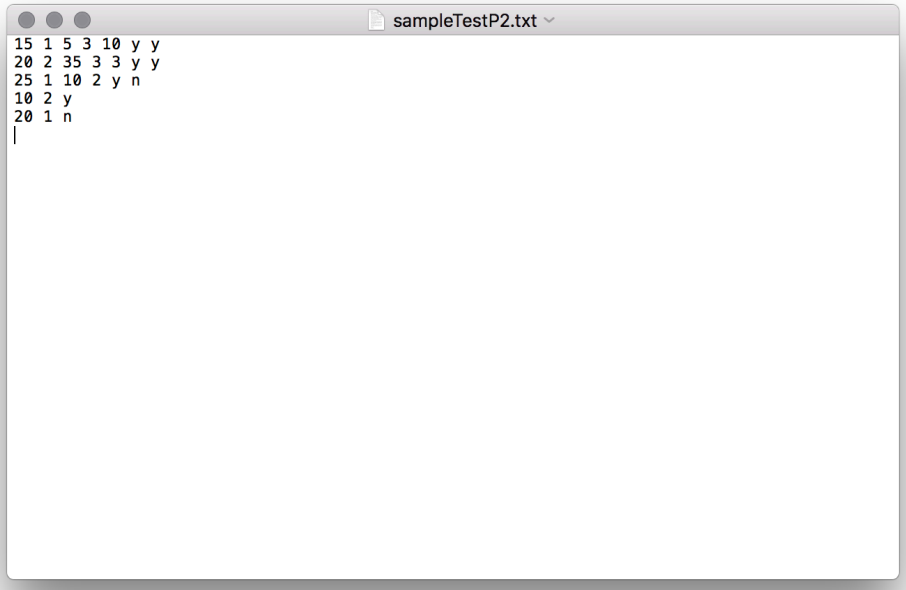
```
    3. Bet on a number between 1 and 10
Please enter a betting option: 3
Please enter a number: 9

The number is: 1
The color is: Red

The amount Jane win is: 0
Play again [y/n]? n

The total amount winning/losing by Jane is: 307                //amount Jane wins
The total amount winning/losing by Jerel is: -70               //amount Jerel lose
The total amount winning/losing by the house is: -237          //amount house lose

Game over!  Thanks for playing.
```

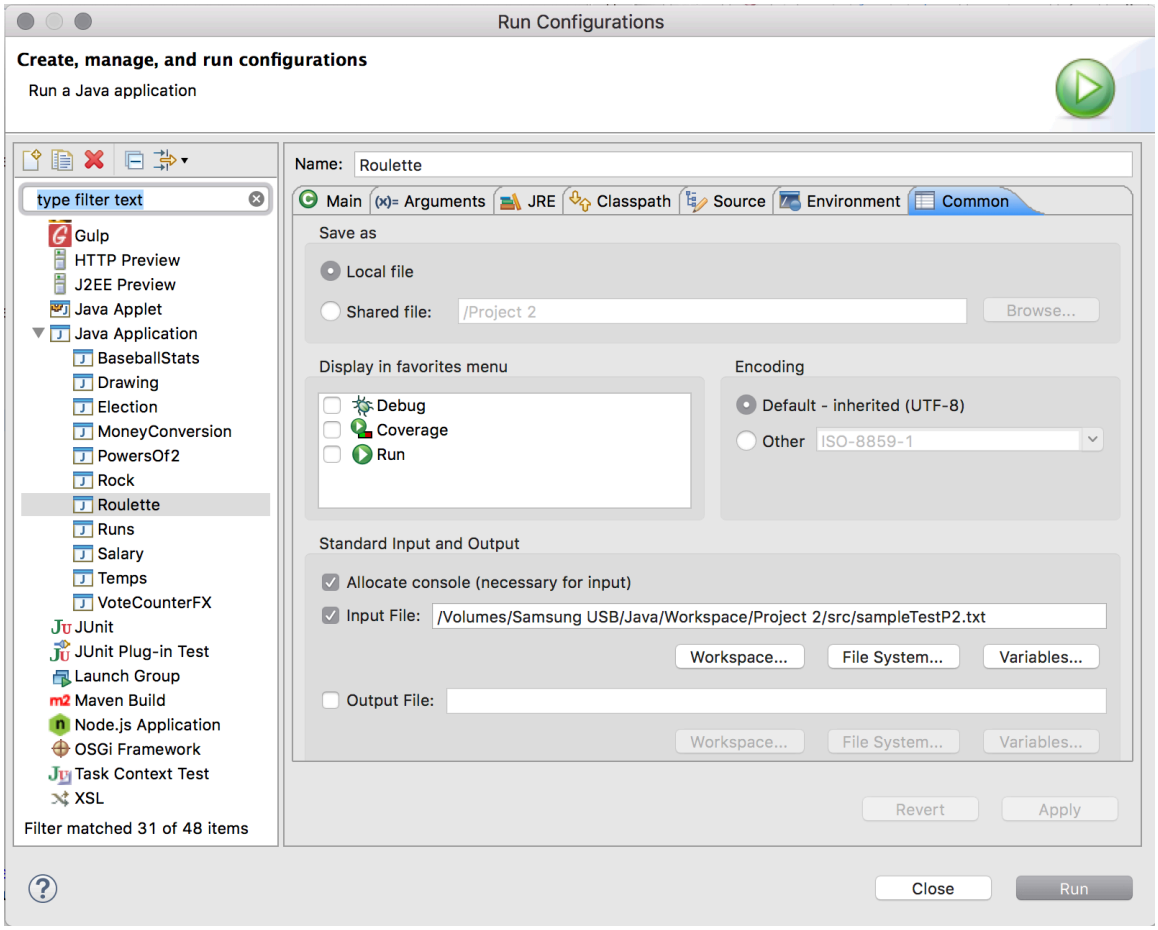
TEST CASE 4: Extra credit output

```
A Simple Roulette Game
By Mai Pham

Welcome to a simple version of roulette game.
You can place a bet on black, red, or a number.
A color bet is paid 2 the bet amount.
A number bet is paid 10 the bet amount.
You can bet on a number from 1 to 10.
Gamble responsibly.  Have fun and good luck!

Money available for Jane: 100
How much to bet: Betting Options:
    1. Bet on black (even numbers)
    2. Bet on red (odd numbers)
    3. Bet on a number between 1 and 10
Please enter a betting option:
Money available for Jerel: 100
How much to bet: Betting Options:
    1. Bet on black (even numbers)
    2. Bet on red (odd numbers)
    3. Bet on a number between 1 and 10
Please enter a betting option: Please enter a number:
The number is: 4
The color is: Black

The amount Jane win is: 30
Play again [y/n]?
The amount Jerel win is: 0
Play again [y/n]?
Money available for Jane: 115
How much to bet: Betting Options:
    1. Bet on black (even numbers)
    2. Bet on red (odd numbers)
    3. Bet on a number between 1 and 10
Please enter a betting option:
Money available for Jerel: 95
How much to bet: Betting Options:
    1. Bet on black (even numbers)
    2. Bet on red (odd numbers)
```

```
    3. Bet on a number between 1 and 10
Please enter a betting option: Please enter a number:
The number is: 00
The color is: Green

The amount Jane win is: 0
Play again [y/n]?
The amount Jerel win is: 0
Play again [y/n]?
Money available for Jane: 95
How much to bet: Betting Options:
    1. Bet on black (even numbers)
    2. Bet on red (odd numbers)
    3. Bet on a number between 1 and 10
Please enter a betting option:
Money available for Jerel: 60
How much to bet: Betting Options:
    1. Bet on black (even numbers)
    2. Bet on red (odd numbers)
    3. Bet on a number between 1 and 10
Please enter a betting option:
The number is: 4
The color is: Black

The amount Jane win is: 50
Play again [y/n]?
The amount Jerel win is: 0
Play again [y/n]?
Money available for Jane: 120
How much to bet: Betting Options:
    1. Bet on black (even numbers)
    2. Bet on red (odd numbers)
    3. Bet on a number between 1 and 10
Please enter a betting option:
The number is: 4
The color is: Black

The amount Jane win is: 0
Play again [y/n]?
Money available for Jane: 110
How much to bet: Betting Options:
    1. Bet on black (even numbers)
    2. Bet on red (odd numbers)
    3. Bet on a number between 1 and 10
Please enter a betting option:
The number is: 00
The color is: Green

The amount Jane win is: 0
Play again [y/n]?
The total amount winning/losing by Jane is: -10
The total amount winning/losing by Jerel is: -50
The total amount winning/losing by the house is: 60

Game over!  Thanks for playing.
```

INPUT TEXT FILE





SOURCE CODE

PLAYER

```
// Class Player for CSCI 145 Project 2 Fall 17
// Modified by: Mai Pham

import java.util.*;

//**************************************************************************
//   Class Player represents one roulette player.
//**************************************************************************
class Player
{

    private static final int RELOAD_AMOUNT = 100;
    private int bet, money;
    private String name;
    private int betType, number;
    private int winLose, housePay;


    //=========================================================================
    //  The Player constructor sets up  name and initial available money.
    //=========================================================================
    public Player (String playerName, int initialMoney)
    {
       name = playerName;
       money = initialMoney;
    } // constructor Player


    //=========================================================================
    //  Returns this player's name.
    //=========================================================================
    public String getName()
    {
       return name;
    }  // method getName


    //=========================================================================
    //  Returns this player's current available money.
    //=========================================================================
    public int getMoney()
    {
       return money;
    }  // method getMoney


    //=========================================================================
    //  Prompts the user and reads betting information.
    //=========================================================================
    public void makeBet(Scanner scan)
    {
       System.out.print("How much to bet: ");
       bet = scan.nextInt();

       while (bet < 1 || bet > money)
       {
           System.out.print("The amount is invalid. \nPlease enter the amount again: ");
           bet = scan.nextInt();
       }
```

```
        money = money - bet;
        housePay += bet;
        winLose -= bet;

        Wheel.betOptions();
        System.out.print("Please enter a betting option: ");
        betType = scan.nextInt();
        while (betType < 1 || betType > 3)
        {
            System.out.print("The betting option is invalid. \nPlease enter betting option
again: ");
            betType = scan.nextInt();
        }

        if (betType == Wheel.NUMBER)
        {
            System.out.print("Please enter a number: ");
            number = scan.nextInt();
            while (number < Wheel.MIN_NUM || number > Wheel.MAX_NUM)
            {
                System.out.print("The number is invalid. \nPlease enter a number again: ");
                number = scan.nextInt();
            }
        }
    } // method makeBet


    public void payment()
    {
        int winAmount;
        winAmount = Wheel.payoff(bet, betType, number);
        money += winAmount;
        housePay -= winAmount;
        winLose += winAmount;
        System.out.println("The amount " + name + " win is: " + winAmount);
    }


    public void displayStatus()
    {
        System.out.println("The total amount winning/losing by " + name + " is: " +
winLose);
    }


    public int byTheHouse()
    {
        return housePay;
    }


    //=================================================================
    //  Determines if the player wants to play again.
    //=================================================================
    public boolean playAgain(Scanner scan)
    {
        String answer;
        System.out.print ("Play again [y/n]? ");
        answer = scan.next();
        if (answer.equalsIgnoreCase("Y"))
            if (money == 0)
                money += RELOAD_AMOUNT;
        return (answer.equals("y") || answer.equals("Y"));
    }  // method playAgain
}
```

WHEEL

```
// Class Wheel for CSCI 145 Project 2 Fall 17
// Modified by: Mai Pham

//*********************************************************************
//   Class Wheel represents a roulette wheel and its operations.  Its
//   data and methods are static because there is only one wheel.
//*********************************************************************
class Wheel
{
    // public name constants -- accessible to others
    public final static int BLACK     =  0;             // even numbers
    public final static int RED       =  1;             // odd numbers
    public final static int GREEN     =  2;             // 00 OR 0
    public final static int NUMBER    =  3;             // number bet
    public final static int MIN_NUM   =  1;             // smallest number to bet
    public final static int MAX_NUM   = 10;             // largest number to bet

    // private name constants -- internal use only
    private final static int MAX_POSITIONS = 12;        // number of positions on wheel
    private final static int NUMBER_PAYOFF = 10;        // payoff for number bet
    private final static int COLOR_PAYOFF  = 2;         // payoff for color bet

    // private variables -- internal use only
    private static int ballPosition;                    // 00, 0, 1 .. 10
    private static int color;                           // GREEN, RED, OR BLACK


    //=====================================================================
    //  Presents welcome message
    //=====================================================================
    public static void welcomeMessage()
    {
       System.out.println("Welcome to a simple version of roulette game.");
       System.out.println("You can place a bet on black, red, or a number.");
       System.out.println("A color bet is paid " + COLOR_PAYOFF + " the bet amount.");
       System.out.println("A number bet is paid " + NUMBER_PAYOFF + " the bet amount.");
       System.out.println("You can bet on a number from " + MIN_NUM + " to " + MAX_NUM +
"."");
       System.out.println("Gamble responsibly.  Have fun and good luck!\n");
    }


    //=====================================================================
    //  Presents betting options
    //=====================================================================
    public static void betOptions()
    {
       System.out.println("Betting Options:");
       System.out.println("    1. Bet on black (even numbers)");
       System.out.println("    2. Bet on red (odd numbers)");
       System.out.println("    3. Bet on a number between " + MIN_NUM + " and " + MAX_NUM);
    }
```

```
    public static void spin()
    {
       ballPosition = (int)(Math.random()*(MAX_POSITIONS));

       if (ballPosition == 0 || ballPosition == (MAX_NUM+1))
       {
           if (ballPosition == (MAX_NUM+1))
               System.out.println("The number is: 00");
           else
               System.out.println("The number is: " + ballPosition);
           color = GREEN;
           System.out.println("The color is: Green");
       }
       else
       {
           System.out.println("The number is: " + ballPosition);
           if (ballPosition % 2 == BLACK)
           {
               color = BLACK;
               System.out.println("The color is: Black");
           }
           else if (ballPosition % 2 == RED)
           {
               color = RED;
               System.out.println("The color is: Red");
           }
       }
    }


    public static int payoff(int bet, int betType, int number)
    {
       int payoff = 0;

       if (betType == NUMBER && number == ballPosition)
           payoff = bet * NUMBER_PAYOFF;
       else if (betType == 1 && color == BLACK || betType == 2 && color == RED)
           payoff = bet * COLOR_PAYOFF;
       return payoff;
    }
}
```

ROULETTE

```
// Class Roullete for CSCI 145 Project 2 Fall 17
// Modified by: Mai Pham

import java.util.*;

//*************************************************************************
//   Class Roulette contains the main driver for a roulette betting game.
//*************************************************************************
class Roulette
{
    //=========================================================================
    //  Contains the main processing loop for the roulette game.
    //=========================================================================
    public static void main (String[] args)
```

```
    {
        Scanner scan = new Scanner(System.in);
        Player player1 = new Player ("Jane", 100);   // $100 to start for Jane
        Player player2 = new Player ("Jerel", 100);   // $100 to start for Jerel
        boolean done1 = false;
        boolean done2 = false;

        System.out.println ("A Simple Roulette Game");
        System.out.println ("By Mai Pham\n");
        Wheel.welcomeMessage();

        while (!done1 || !done2)
        {
            if (!done1)
            {
                System.out.println ("Money available for " + player1.getName()
                                            + ": " + player1.getMoney());
                player1.makeBet(scan);
                System.out.println();
            }
            if (!done2)
            {
                System.out.println ("Money available for " + player2.getName()
                                            + ": " + player2.getMoney());
                player2.makeBet(scan);
                System.out.println();
            }

            Wheel.spin();
            System.out.println();

            if (!done1)
            {
                player1.payment();
                done1 = !player1.playAgain(scan);
                System.out.println();
            }
            if (!done2)
             {
                player2.payment();
                done2 = !player2.playAgain(scan);
                System.out.println();
              }
        }
        player1.displayStatus();
        player2.displayStatus();

        System.out.println("The total amount winning/losing by the house is: "
                                            + (player1.byTheHouse() + player2.byTheHouse()));

        System.out.println ("\nGame over!  Thanks for playing.");
        scan.close();
    }
}
```