Cricket Game Project Explanation

Source Code :

Here is the source code to implement simple cricket game in Java.

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.Arrays;
import java.util.List;
import java.util.Random;

public class CricketGameSeries {
  private static BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
  private static Random rand = new Random();

  public String chooseTeam() throws IOException {
    List<String> teams = Arrays.asList("Australia", "England", "India", "South
Africa", "New Zealand", "Pakistan", "Bangladesh", "Srilanka", "West Indies");
    System.out.println("Choose team out of the following:");
    for (int i = 0; i < teams.size(); i++) {
      System.out.print(teams.get(i) + " ");
    }
    System.out.println();
    String playersTeam = br.readLine();
    System.out.println();
    if (!teams.contains(playersTeam)) {
      System.out.println("Invalid input... Program terminating.");
      return "xxx";
    } else
      return playersTeam;
  }

  public String chooseOpponent() throws IOException {
    List<String> teams = Arrays.asList("Australia", "England", "India", "South
Africa", "New Zealand", "Pakistan", "Bangladesh", "West Indies", "Srilanka");
    System.out.println("Choose opponent's team out of:");
    for (int i = 0; i < teams.size(); i++)
      System.out.print(teams.get(i) + " ");
```

```
    System.out.println();
    String opp_Team = br.readLine();
    System.out.println();
    if (!teams.contains(opp_Team)) {
        System.out.println("Invalid input... Program terminating.");
        return "xxx";
    } else
        return opp_Team;
}

public int chooseOver() throws IOException {
    List<Integer> overs = Arrays.asList(1, 3, 5, 10);
    int flag = 0, overschoice = 0;
    System.out.print("Choose no. of overs (1, 3, 5, or 10): ");
    while (flag == 0) {
        try {
            overschoice = Integer.parseInt(br.readLine());
            System.out.println();
        } catch (Exception e) {
            System.out.println("Invalid input. Choose no. of overs again.");
            continue;
        }
        flag++;
    }
    if (!overs.contains(overschoice)) {
        System.out.println("Invalid input... Program terminating.");
        return -1;
    } else
        return overschoice;
}

public String toss() throws IOException {
    List<String> tossArr = Arrays.asList("heads", "tails");
    int rnd = rand.nextInt(tossArr.size());
    String toss_Result = tossArr.get(rnd);
    System.out.print("Say Heads or Tails: ");
    String choice = br.readLine();
    if (!tossArr.contains(choice.toLowerCase())) {
        System.out.println("Invalid input... Program terminating.");
        return "xxx";
```

```
    } else {
      if (choice.equalsIgnoreCase(toss_Result)) {
        System.out.println("You have won the toss! Choose batting or bowling.");
        String choice2 = br.readLine();
        System.out.println();
        if (choice2.equalsIgnoreCase("batting") || choice2.equalsIgnoreCase("bowling") || choice2.equalsIgnoreCase("bat") || choice2.equalsIgnoreCase("bowl"))
          return choice2;
        else
          return "xxx";
      } else {
        System.out.println("Oops! Wrong choice...");
        System.out.println();
        String[] bat_bowl = {"batting", "bowling"};
        rnd = rand.nextInt(bat_bowl.length);
        String batbowl = bat_bowl[rnd];
        System.out.println("Your opponent has chosen " + batbowl);
        return (batbowl.equals("batting") ? "bowling" : "batting");
      }
    }
  }

  public int[] batting(int over, int target) throws IOException {
    int arr1[] = {0, 0, 0, 0, 0, 0};
    CricketGameSeries ob1 = new CricketGameSeries();
    int arr[] = new int[2];
    int limit = 6, user = 0;
    System.out.println("Allowed numbers: 1 to 6");

    int wickets = 0, runs = 0;
    for (int i = 1; i <= over; i++) {
      inner:
      for (int j = 1; j <= 6; j++) {
        if (runs > target) {
          arr[0] = runs;
          arr[1] = wickets;
          return arr;
        }
```

```
System.out.println("Get ready to bat! Take out a number:");
try {
   user = Integer.parseInt(br.readLine());
} catch (Exception e) {
   ob1.restart();
}
if (user < 1 || user > limit) {
   System.out.println("Invalid input. Reface the ball.");
   j--;
   continue inner;
}
System.out.println();
arr1[j - 1] = user;
System.out.println("Computer's turn:");
int random1 = rand.nextInt(7);
int random2 = rand.nextInt(7);
int rnd = 0;
if (j > 2 && (arr1[j - 3] == arr1[j - 2]))
   rnd = arr1[j - 2];
else if (j == random1 || j == random2)
   rnd = 6;
else {
   do {
      rnd = rand.nextInt(limit + 1);
   } while (rnd == 0);
}
System.out.println(rnd);
if (user == rnd) {
   wickets++;
   System.out.println();
   System.out.println("Oops! A wicket gone....");
   System.out.println("Score = " + runs + " / " + wickets);
   System.out.println();
   if (wickets == 3) {
      System.out.println("Your team is all out!");
      System.out.println("_____");
      arr[0] = runs;
      arr[1] = 10;
      return arr;
   }
```

```
            } else {
                runs += user;
                System.out.println();
                System.out.println("Score = " + runs + " / " + wickets);
            }
            if (j == 6) {
                System.out.println("Current Run Rate = " + runs / (double) i);
                if ((target != 400) && (i != over))
                    System.out.println("Required Run Rate = " + (target - runs) /
(double) (over - i));
            }
            System.out.println();
        }
    }
    arr[0] = runs;
    arr[1] = wickets;
    return arr;
  }

  public int[] bowling(int over, int target) throws IOException {
    int[] arr = new int[2];
    int limit = 6, user = 0;
    System.out.println("Allowed numbers: 1-6");
    System.out.println();
    CricketGameSeries ob2 = new CricketGameSeries();
    int wickets = 0, runs = 0;
    int[] invalid = {1, 4};
    for (int i = 1; i <= over; i++) {
      inner:
      for (int j = 1; j <= 6; j++) {
        if (runs > target) {
          arr[0] = runs;
          arr[1] = wickets;
          return arr;
        }
        System.out.println("Get ready to bowl. Take out a number:");
        try {
          user = Integer.parseInt(br.readLine());
        } catch (Exception e) {
          ob2.restart();
```

```
            }
            if (user < 1 || user > limit) {
               int r = rand.nextInt(invalid.length);
               if (invalid[r] == 1) {
                  System.out.println("Wide! 1 run added to the opponent's score.");
                  runs += 1;
               } else {
                  System.out.println("Byes! 4 runs added to the opponent's score.");
                  runs += 4;
               }
               j--;
               continue inner;
            }
            System.out.println("Computer's turn: ");
            int rand1 = rand.nextInt(7), rand2 = rand.nextInt(7), r2 = 0;
            if (j == rand1)
               r2 = 6;
            else if (j == rand2)
               r2 = 5;
            else {
               do {
                  r2 = rand.nextInt(limit);
               } while (r2 == 0);
            }
            System.out.println(r2);
            if (user == r2) {
               wickets++;
               System.out.println("Beautiful delivery! Batsman dismissed as a wicket
goes down!");
               System.out.println("Score = " + runs + " / " + wickets);
               System.out.println();
               if (wickets == 3) {
                  System.out.println("Opponent team all out!");
                  System.out.println("_____");
                  arr[0] = runs;
                  arr[1] = wickets;
                  return arr;
               }
            } else {
               runs += r2;
```

```
            System.out.println("Score = " + runs + " / " + wickets);
            System.out.println();
          }
          if (j == 6) {
            System.out.println("Current Run Rate = " + runs / (double) i);
            if ((target != 400) && (i != over))
              System.out.println("Required Run Rate = " + (target - runs) /
(double) (over - i));
          }
          System.out.println();
        }
      }
      arr[0] = runs;
      arr[1] = wickets;
      return arr;
    }

    public static void restart() throws IOException {
      System.out.println("Press 0 to exit or 1 to restart the game.");
      int n = Integer.parseInt(br.readLine());
      System.out.println();
      if (n == 0)
        System.exit(0);
      else
        main(null);
    }

    public static void main(String[] args) throws IOException {
      System.out.println("WELCOME TO CRICKET !!!");
      System.out.println("_____");
      System.out.println("RULES OF THE GAME:");
      System.out.println("1.) There are 6 numbers allowed, from 1 to 6.");
      System.out.println("2.) A wicket falls from the batting side if the bowler and
batsman both take out the same number.");
      System.out.println("3.) You and the computer both have 3 wickets in total.");
      System.out.println("3.) Please take care of cases (exactly the same case as
provided to you in choices) else the program restarts.");
      System.out.println("_____");

      CricketGameSeries ob = new CricketGameSeries();
```

```
        String error = "xxx";
        String playersTeam = ob.chooseTeam();
        int arr1[] = new int[2], arr2[] = new int[2];

        if (playersTeam.equals(error)) {
            System.out.println("Program ended.");
            ob.restart();
        } else {
            String opp = ob.chooseOpponent();

            if (opp.equals(error)) {
                System.out.println("Program ended.");
                ob.restart();
            } else {
                if (playersTeam.equalsIgnoreCase(opp)) {
                    System.out.println("You can't choose the same team as the opponent.");
                    ob.restart();
                }

                System.out.print("Enter the number of matches you want to play (odd number only): ");
                int numMatches = Integer.parseInt(br.readLine());

                if (numMatches % 2 == 0) {
                    System.out.println("Number of matches should be odd. Program terminating.");
                    ob.restart();
                }

                int totalMatches = numMatches;
                int playerScore = 0, opponentScore = 0;

                for (int match = 1; match <= numMatches; match++) {
                    System.out.println("************************ MATCH " + match + " ************************");

                    int over = ob.chooseOver();

                    if (over == -1) {
```

```
                System.out.println("Program ended.");
                ob.restart();
            } else {
                String toss = ob.toss();

                if (toss.equals(error)) {
                    System.out.println("Program ended.");
                    ob.restart();
                } else if (toss.equalsIgnoreCase("batting") ||
toss.equalsIgnoreCase("bat")) {
                    arr1 = ob.batting(over, 400);
                    System.out.println(opp + " needs " + (arr1[0] + 1) + " runs to
win.");
                    arr2 = ob.bowling(over, arr1[0]);

                    if (arr2[0] > arr1[0]) {
                        System.out.println("You lost! " + opp + " won by " + (10 -
arr2[1]) + " wickets.");
                        opponentScore++;
                    } else if (arr2[0] < arr1[0]) {
                        System.out.println("Congrats! You won! " + playersTeam + "
won by " + (arr1[0] - arr2[0]) + " runs.");
                        playerScore++;
                    } else {
                        System.out.println("Game drawn!");
                    }
                } else if (toss.equalsIgnoreCase("bowling") ||
toss.equalsIgnoreCase("bowl")) {
                    arr1 = ob.bowling(over, 400);
                    System.out.println("You need " + (arr1[0] + 1) + " runs to win.");
                    arr2 = ob.batting(over, arr1[0]);

                    if (arr2[0] > arr1[0]) {
                        System.out.println("Congrats! You (" + playersTeam + ") won by
" + (10 - arr2[1]) + " wickets!");
                        playerScore++;
                    } else if (arr2[0] < arr1[0]) {
                        System.out.println("You lost! " + opp + " won by " + (arr1[0] -
arr2[0]) + " runs.");
                        opponentScore++;
```

```
                } else {
                    System.out.println("Game Drawn!");
                }
            } else {
                System.out.println("Error.");
            }

            System.out.println("************************ END OF MATCH "
+ match + " ************************");
            System.out.println();
        }
    }

    System.out.println("************************ SERIES RESULT
************************");
    System.out.println("Total Matches Played: " + totalMatches);
    System.out.println(playersTeam + " Score: " + playerScore);
    System.out.println(opp + " Score: " + opponentScore);

    if (playerScore > opponentScore) {
        System.out.println(playersTeam + " won the series!");
    } else if (opponentScore > playerScore) {
        System.out.println(opp + " won the series!");
    } else {
        System.out.println("The series is drawn!");
    }
  }
 }
}
```

Program Explanation : The "Cricket Game Series" program is a console-based cricket game simulation written in Java. Let's break down the code and explain its various components :

1. **Class Declaration :**
   - The program starts with the declaration of a class named **CricketGameSeries**.
2. **Member Variables :**
   - The class contains two static variables, **br** (BufferedReader) and **rand** (Random), for handling input and generating random numbers.

3. **Team Selection Methods :**
    - **chooseTeam()** and **chooseOpponent()** methods allow the user to select their team and opponent from predefined lists of international cricket teams. Invalid inputs are handled, and the program terminates if necessary.
4. **Overs Selection Method :**
    - **chooseOver()** method prompts the user to choose the number of overs for the match, and the input is validated. If the input is invalid, the program terminates.
5. **Toss Method :**
    - The **toss()** method simulates a coin toss. The user is prompted to choose "Heads" or "Tails." Based on the toss result, the user may choose to bat or bowl. If the user wins the toss, they can choose; otherwise, the computer randomly decides.
6. **Batting and Bowling Methods :**
    - **batting(int over, int target)** and **bowling(int over, int target)** methods simulate the batting and bowling innings, respectively.
    - Users and the computer take turns choosing numbers (1 to 6) for each ball.
    - Wickets are determined if the user and computer select the same number.
    - Runs are accumulated, and the innings continue until the specified number of overs is completed or all wickets are taken.
7. **Restart Method :**
    - The **restart()** method allows the user to restart the game or exit by entering 0 or 1, respectively.
8. **Main Method :**
    - The **main()** method is the entry point of the program.
    - It provides an introduction to the game and its rules.
    - The user selects their team, opponent, and the number of matches in the series.
    - Matches are played, and the series result is displayed at the end.
9. **Series Outcome & Result :**
    - The program keeps track of the scores in each match and determines the winner of each match based on runs or wickets.
    - The overall series winner is decided by comparing the total scores of the player and the computer.
10. **Error Handling :**
    - The program includes error handling for invalid inputs, ensuring a smoother user experience.

- If an invalid input is detected, appropriate error messages are displayed, and the program may terminate or prompt the user to restart.

11. Random Number Generation :

- The **Random** class is used to generate random numbers, simulating aspects like the toss and computer's choices during the game.

12. Overall Features :

- The program provides a simple and interactive text-based cricket game experience, allowing users to play a series of matches against a computer opponent.

Note : While the code structure is functional, there are some areas for improvement, such as code modularization, comments for better readability, and the potential for enhanced user interactions. Additionally, the game logic is relatively simple, and the focus is on educational aspects rather than sophisticated game mechanics or graphical user interfaces.