```java
import java.util.Scanner;

/**
 * This class holds the user interaction functionality of the card game and
 * contains the main method to start the game. It extends Board class
 * in order to access BOARD listArray and protected methods.
 */

public class ElevensGame extends Board
{
    private int selectCard() // Allows user to select card slot with appropriate validation in place
    {
        Scanner scan = new Scanner(System.in);
        AListArray<Integer> SELECTION = checkPossibleMoves(BOARD);
        int selection = 0;

        System.out.print("Please Enter a Slot Number for any Card (or any letter key for a hint): ");
        boolean isNum=false;
        while(!isNum) { // while loop validation only accepting when isNum is set to true
            if(scan.hasNextInt()) {
                    selection = scan.nextInt();
                if(selection >= 1 && selection <= getBoardLength())
                    isNum=true;
                else {
                    System.out.println("Invalid Entry. Not within range of slots in Board.");
                    System.out.print("\nPlease Enter a slot number for any card (or any letter key for
 a hint): ");
                }
            }
            else {
                if(SELECTION.getLength() < 3) {
                    System.out.println("Hint: " + getBoardEntry(SELECTION.getEntry(1)) + " & " +
                            getBoardEntry(SELECTION.getEntry(2)));
                }
                else {
                    System.out.println("Hint: " + getBoardEntry(SELECTION.getEntry(1)) + " & " +
                            getBoardEntry(SELECTION.getEntry(2)) + " & " +
                            getBoardEntry(SELECTION.getEntry(3)));
                }
                System.out.print("\nPlease Enter a slot number for any card: ");
                scan.next();
            }
        }   isNum = false;  //resetting isNum to false for next use

        return switch (selection) {
            case 2 -> 2;
            case 3 -> 3;
            case 4 -> 4;
            case 5 -> 5;
            case 6 -> 6;
            case 7 -> 7;
            case 8 -> 8;
            case 9 -> 9;
            default -> 1;
        };
    }

    private void demonstrationMode() // allows user to increment through a simulated game of Elevens
step by step
    {
        Scanner scan = new Scanner(System.in);
        while(getBoardLength() > 0) {
            String enterkey = "\nPress 'Enter' key to increment through Demonstration Mode.";
            System.out.print(enterkey);
            enterkey = scan.nextLine();
            System.out.print(enterkey);
            if(enterkey.equals("")) {
                displayBoard(BOARD);
                AListArray<Integer> SELECTION = checkPossibleMoves(BOARD);
                if (checkPossibleMoves(BOARD).isEmpty()) {
                    System.out.println("----STALEMATE----\n");
                    menu();
                } else if (checkPossibleMoves(BOARD).getLength() < 3) {
                    if (getBoardEntry(SELECTION.getEntry(1)).equalEleven(getBoardEntry(SELECTION.
getEntry(2))) == 1) {
                        System.out.println("----Removed: "+getBoardEntry(SELECTION.getEntry(1))+" & "
                                +getBoardEntry(SELECTION.getEntry(2))+"----");
                        replaceCards(SELECTION.getEntry(1), SELECTION.getEntry(2));
                    }
                } else {
                    if (getBoardEntry(SELECTION.getEntry(1)).equalJQK(getBoardEntry(SELECTION.getEntry(2)),
                            getBoardEntry(SELECTION.getEntry(3))) == 1) {
                        System.out.println("----Removed: " + getBoardEntry(SELECTION.getEntry(1)) + " & "
                                + getBoardEntry(SELECTION.getEntry(2))
                                + " & " + getBoardEntry(SELECTION.getEntry(3)) + "----");
                        replaceCards(SELECTION.getEntry(1), SELECTION.getEntry(2), SELECTION.getEntry(3));
                    }
                }
            }
        }
    }

    private void menu() // Elevens menu functionality
    {
        Scanner scan = new Scanner(System.in);
        System.out.print("Enter 'r' to replay your moves, 'p' to play again or any other key to exit : ");
        String selection = scan.next();
        if(selection.equals("r") || selection.equals("R")) {
            replaySteps();
            menu();
        }
        else if(selection.equals("p") || selection.equals("P")) {
            newBoard();
            game();
        }
        else {
            System.exit(0);
        }
    }

    public void game() // Elevens main game functionality
    {
        Scanner scan = new Scanner(System.in);
        System.out.println("\n-------------------------Welcome to Elevens --------------------------");
        System.out.println("Rules");
        System.out.println("Select two cards that add to eleven or a Jack, Queen, King combination.");
        System.out.println("If you can remove all the cards from the board and the deck you Win!\n");

        System.out.print("Enter 'p' to play or 'd' for demonstration mode: ");
        String selection = scan.next();
        if(selection.equals("d") || selection.equals("D")) {
            demonstrationMode();
        }
        int first, second, third;
        while(getBoardLength() > 0) {
            displayBoard(BOARD);
            if(checkPossibleMoves(BOARD).isEmpty()) {
                System.out.println("----STALEMATE----\n");
                menu();
            }
            if (getBoardEntry(first = selectCard()).getRankValue() <= 10) {
                System.out.println("\nPlease select a second card to add to Eleven");
                if (getBoardEntry(first).equalEleven(getBoardEntry(second = selectCard())) == 1) {
                    System.out.println("\n----Removed: "+getBoardEntry(first) + " & " + getBoardEntry(second) +"----");
                    replaceCards(first, second);
                } else {
                    System.out.println("\n" + getBoardEntry(first) + " & " + getBoardEntry(second) + " selected.");
                    System.out.println("----They don't add to Eleven. Please try again----");
                }
            } else if (getBoardEntry(first).getRankValue() >= 11) {
                System.out.println("\nPlease select two more different picture cards.");
                if (getBoardEntry(first).equalJQK(getBoardEntry(second = selectCard()), getBoardEntry(third = selectCard())) == 1) {
                    System.out.println("\n----Removed: "+getBoardEntry(first) + ", " + getBoardEntry(
second) +
                            ", " + getBoardEntry(third) +"----");
                    replaceCards(first, second, third);
                } else {
                    System.out.println("\n" +getBoardEntry(first) + ", " + getBoardEntry(second) + ", " +
                            getBoardEntry(third) +" selected.");
                    System.out.println("----Not a Jack, Queen and King. Please try again----");
                }
            }
        }
        System.out.println("\n----Congratulations! You've Won!----\n");
        menu();
    }

    public static void main(String[] args) // main to instantiate a new game of Elevens
    {
        ElevensGame elevens = new ElevensGame();
        elevens.game();
    }
}

```