```java
package Game;

import java.util.*;

public class Player {
    protected String name;
    protected List<Card> hand;
    protected int score = 0;
    protected List<Card> initialHand;

    public Player(String name) {
        this.name = name;
        this.hand = new ArrayList<>();
    }

    public void drawHand(Deck deck) {
        hand.clear();
        for (int i = 0; i < 5; i++) {
            hand.add(deck.draw());
        }
        initialHand = new ArrayList<>(hand);
    }

    public String getInitialHand() {
        return initialHand.toString();
    }

    public List<Card> getHand() {
        return hand;
    }

    public void playTurn(Deck deck, Scanner scanner) {
        System.out.println("\n" + name + "'s turn!");
        System.out.println("Your hand: ");
        for (int i = 0; i < hand.size(); i++) {
            System.out.println((i + 1) + ". " + hand.get(i));
        }

        List<Card> cardsToRemove = selectCardsToRemove(scanner);
        hand.removeAll(cardsToRemove);

        while (hand.size() < 3 && !deck.isEmpty()) {
            Card newCard = deck.draw();
            hand.add(newCard);
            System.out.println("Drew new card: " + newCard);
        }

        int roundScore = calculateScore(hand);
        score += roundScore;
        System.out.println(name + " scored: " + roundScore + " (Total: " + score + ")");
    }

    protected List<Card> selectCardsToRemove(Scanner scanner) {
        List<Card> selected = new ArrayList<>();

        while (selected.size() < 1) {
            System.out.print("Select first card to remove (1-" + hand.size() + "): ");
            try {
                int choice = scanner.nextInt() - 1;
                scanner.nextLine();
                if (choice >= 0 && choice < hand.size()) {
                    selected.add(hand.get(choice));
                } else {
                    System.out.println("□ Invalid selection. Please choose between 1-" +
hand.size());
                }
            } catch (InputMismatchException e) {
                System.out.println(" Please enter a number.");
                scanner.nextLine();
            }
        }

        while (selected.size() < 2) {
            System.out.print("Select second card to remove (1-" + hand.size() + ", cannot
choose " +
                            (hand.indexOf(selected.get(0)) + 1) + "): ");
            try {
                int choice = scanner.nextInt() - 1;
                scanner.nextLine();
                if (choice >= 0 && choice < hand.size() &&
!selected.contains(hand.get(choice))) {
                    selected.add(hand.get(choice));
                } else {
                    System.out.println(" Invalid selection. Choose a different card (1-" +
hand.size() + ")");
                }
            } catch (InputMismatchException e) {
                System.out.println(" Please enter a number.");
                scanner.nextLine();
            }
        }

        return selected;
    }

    protected int calculateScore(List<Card> hand) {
        int total = hand.stream().mapToInt(Card::getValue).sum();
        int score = Math.abs(15 - total);

        boolean allSameColor = hand.stream().allMatch(c -> c.isRed()) ||
                               hand.stream().noneMatch(c -> c.isRed());
        boolean allSameSuit = hand.stream().map(Card::getSuit).distinct().count() == 1;

        if (allSameColor) score -= 1;
        if (allSameSuit) score -= 2;

        return Math.max(0, score);
    }

    public String getName() { return name; }
    public int getScore() { return score; }
}
```