

Project Title:

Rock, Paper, Scissors Game (Two Players)

1. Introduction

The Rock, Paper, Scissors game is a hand game commonly played by two players. In this project, we have implemented a simple console-based version of the game where two players take turns choosing between Rock, Paper, or Scissors. The game then determines the winner based on the following rules:

- Rock crushes Scissors (Rock > Scissors)
- Scissors cuts Paper (Scissors > Paper)
- Paper covers Rock (Paper > Rock)

2. Requirements

2.1 Functional Requirements

- The game should allow two players to enter their choices.
- The game should display the winner after each round.
- Players should be able to play multiple rounds without restarting the program.
- The game should handle invalid input and ask players to re-enter their choices when necessary.

2.2 Non-Functional Requirements

- The game should be easy to understand and user-friendly.
- The program should execute within a reasonable amount of time and have minimal latency.

2.3 Tools and Technologies

- Programming Language: **C**

- Development Environment: Any C compiler

3. Code Structure

3.1 Functions Used

1. **determineWinner()**

- o **Description:** This function compares the choices of the two players and determines the winner.
- o **Parameters:**
 - player1Choice (char)
 - player2Choice (char)
- o **Return Type:** None (prints the result to the console)

2. **main()**

- o **Description:** This is the main function where the game logic is implemented. It handles user input, validates the choices, calls the determineWinner() function, and controls the flow of the game.
- o **Parameters:** None
- o **Return Type:** int (exit code)

4.CODE

```
#include <stdio.h>
```

```
// Function to determine the winner
```

```
void determineWinner(char player1Choice, char player2Choice) {
```

```
    if (player1Choice == player2Choice) {
```

```
        printf("It's a tie! Both chose %c.\n", player1Choice);

    } else if ((player1Choice == 'R' && player2Choice == 'S') ||

               (player1Choice == 'S' && player2Choice == 'P') ||

               (player1Choice == 'P' && player2Choice == 'R')) {

        printf("Player 1 wins! %c beats %c.\n", player1Choice, player2Choice);

    } else {

        printf("Player 2 wins! %c beats %c.\n", player2Choice, player1Choice);

    }

}

int main() {

    char player1Choice, player2Choice;

    int playAgain = 1;

    printf("Welcome to the Rock, Paper, Scissors Game!\n");

    while (playAgain) {

        // Get Player 1's choice

        printf("\nPlayer 1, enter your choice (R for Rock, P for Paper, S for Scissors): ");

        scanf(" %c", &player1Choice); // Read Player 1's input

        // Ensure valid input

        if (player1Choice != 'R' && player1Choice != 'P' && player1Choice != 'S') {

            printf("Invalid input! Please enter 'R', 'P', or 'S'.\n");
```

```
        continue;
    }

    // Get Player 2's choice

    printf("\nPlayer 2, enter your choice (R for Rock, P for Paper, S for Scissors): ");

    scanf(" %c", &player2Choice); // Read Player 2's input


    // Ensure valid input

    if (player2Choice != 'R' && player2Choice != 'P' && player2Choice != 'S') {

        printf("Invalid input! Please enter 'R', 'P', or 'S'.\n");

        continue;
    }


    // Display choices

    printf("\nPlayer 1 chose: %c\n", player1Choice);

    printf("Player 2 chose: %c\n", player2Choice);


    // Determine the winner

    determineWinner(player1Choice, player2Choice);


    // Ask if they want to play again

    printf("\nDo you want to play again? (1 for Yes, 0 for No): ");
```

```
        scanf("%d", &playAgain);

    }

    printf("\nThanks for playing! Goodbye!\n");

    return 0;

}
```

Code link:

<https://onlinegdb.com/X3UJDkD3h>

Example Output:

Welcome to the Rock, Paper, Scissors Game!

Player 1, enter your choice (R for Rock, P for Paper, S for Scissors): R

Player 2, enter your choice (R for Rock, P for Paper, S for Scissors): S

Player 1 chose: R

Player 2 chose: S

Player 1 wins! R beats S.

Do you want to play again? (1 for Yes, 0 for No): 1

Player 1, enter your choice (R for Rock, P for Paper, S for Scissors): P

Player 2, enter your choice (R for Rock, P for Paper, S for Scissors): S

Player 1 chose: P

Player 2 chose: S

Player 2 wins! S beats P.

Do you want to play again? (1 for Yes, 0 for No): 0

Thanks for playing! Goodbye!

5.Code Explanation

5.1 Input Validation

The game prompts the players to enter their choice (R, P, or S). It ensures that the input is valid by checking whether the entered character is one of these three choices. If the input is invalid, it asks the player to re-enter their choice.

5.2 Winner Determination

The determine Winner() function compares the players' choices using conditional statements:

- If both players choose the same option, it's a tie.
- The function checks the winning conditions: Rock beats Scissors, Scissors beats Paper, and Paper beats Rock. It prints the appropriate winner message based on the comparison.

5.3 Replay Option

After each round, the game asks if the players want to continue playing. The players can choose to either play again by entering 1 or stop by entering 0.

6. Testing

6.1 Test Cases

Here are some example test cases for the Rock, Paper, Scissors game:

1. **Test Case 1:** Both players choose Rock
 - o **Input:** Player 1: R, Player 2: R
 - o **Output:** "It's a tie! Both chose R."
2. **Test Case 2:** Player 1 chooses Paper, Player 2 chooses Rock
 - o **Input:** Player 1: P, Player 2: R
 - o **Output:** "Player 1 wins! P beats R."
3. **Test Case 3:** Player 1 chooses Scissors, Player 2 chooses Paper
 - o **Input:** Player 1: S, Player 2: P
 - o **Output:** "Player 1 wins! S beats P."
4. **Test Case 4:** Player 1 chooses Rock, Player 2 chooses Scissors
 - o **Input:** Player 1: R, Player 2: S
 - o **Output:** "Player 1 wins! R beats S."

5. **Test Case 5:** Invalid input from Player 1

- o **Input:** Player 1: X, Player 2: P
- o **Output:** "Invalid input! Please enter 'R', 'P', or 'S'."

7. **User Guide**

7.1 **How to Play**

1. Upon running the program, Player 1 will be prompted to enter their choice (R, P, or S).
2. Player 2 will then be prompted to enter their choice.
3. After both players have entered their choices, the winner will be displayed, along with a message showing which option won.
4. Players will be asked if they want to play again. If yes, the game will continue; if no, the program will end.

Conclusion

The Rock, Paper, Scissors game is a simple yet effective way to demonstrate the basic use of conditionals, loops, and user interaction in C programming. This project provides an opportunity to learn about input validation, comparison operators, and handling multiple rounds in a game scenario.