Rock-Paper-Scissors Game Report

Purpose: This report details the implementation of a Rock-Paper-Scissors game using Python.

Prepared by: Diya Maheshwari

University Roll Number: 202401100300108

Date: 11/03/2025

2. Introduction

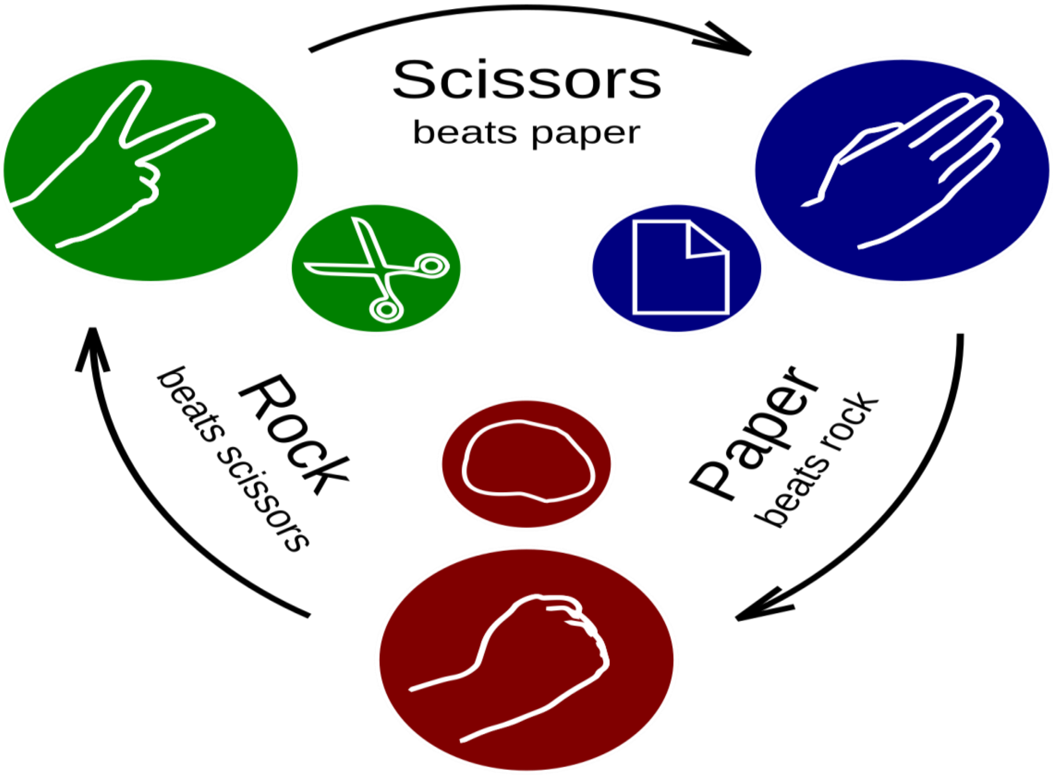
The Rock-Paper-Scissors game is a simple yet popular game where players select one of the three choices: rock, paper, or scissors. The game follows the standard rules:

- Rock beats Scissors

- Scissors beat Paper

- Paper beats Rock

In this project, a Python-based command-line version of the game has been developed. The user plays against the computer, with an option to choose between Best of 3 or Best of 5 rounds. The program ensures fair gameplay with a randomized computer choice, real-time score tracking, and final result announcement.



3. Methodology

Development Approach:

- Programming Language: Python 3.x

- Structure: Modular functions for improved readability and maintainability.

- Input Validation: Ensures users enter valid choices (rock, paper, scissors) and game modes (3 or 5 rounds).

- Randomized Selection: Uses Python's `random.choice()` to generate the computer's move.

- Game Logic: Determines winners based on Rock-Paper-Scissors rules.

- User Experience Enhancements: Includes score tracking and result announcements.

Implementation Steps:

1. User Prompt: Ask the user to select Best of 3 or Best of 5 mode.

2. Game Execution:

   - User inputs choice.

   - Computer generates a random move.

   - Determine round winner and update scores.

3. Final Result:

   - Announce winner based on final scores.

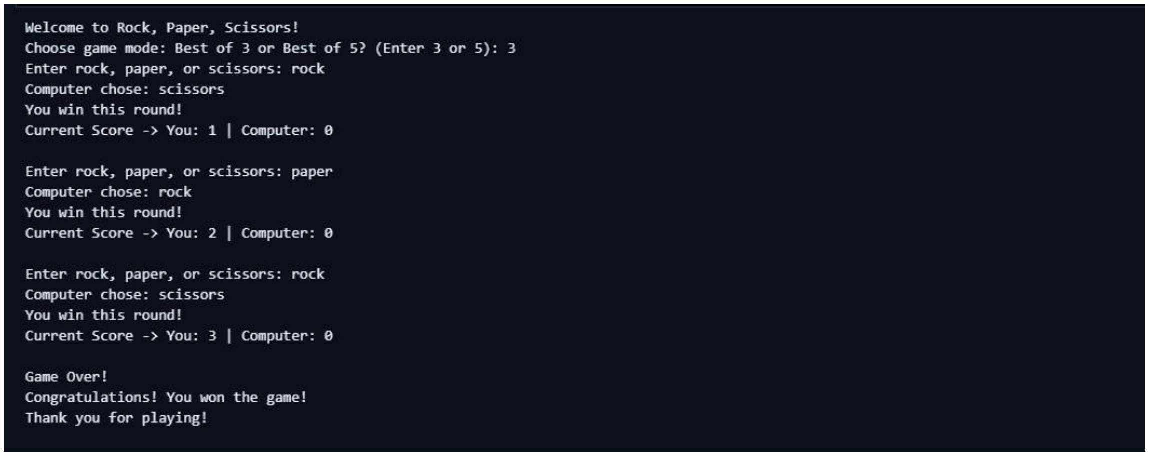
   - Display a motivational message.

4. Code Typed

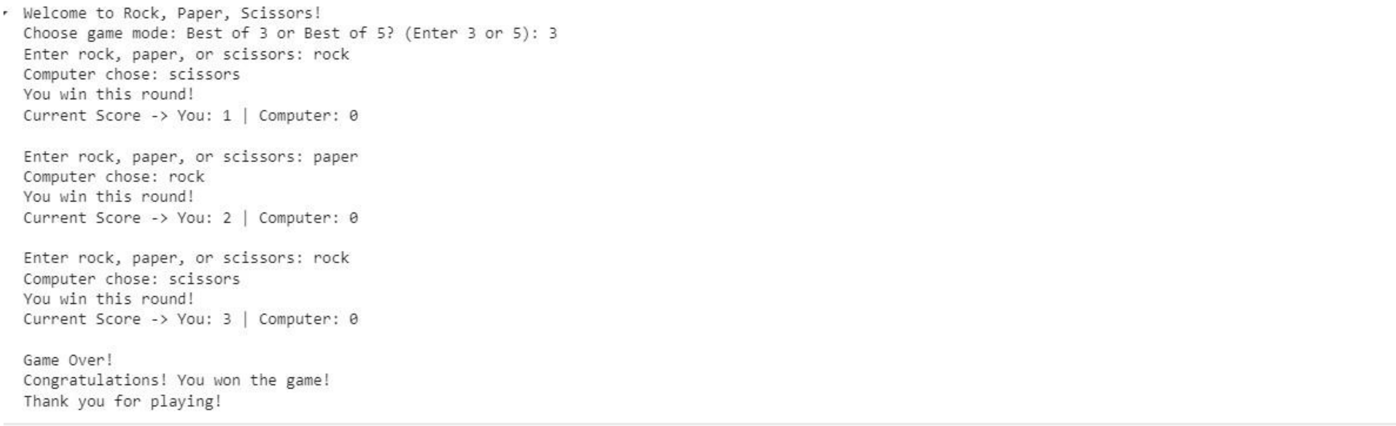
```
import random  # Importing the random module for computer's choice
def get_user_choice():
    """Prompts the user for input and ensures it is a valid choice."""
    choices = ['rock', 'paper', 'scissors']
    user_choice = input("Enter rock, paper, or scissors: ").lower()
    while user_choice not in choices:
        print("Invalid choice. Please enter 'rock', 'paper', or 'scissors'.")
        user_choice = input("Enter rock, paper, or scissors: ").lower()
    return user_choice


def get_computer_choice():
    """Randomly selects and returns the computer's choice."""
    return random.choice(['rock', 'paper', 'scissors'])


def determine_winner(user, computer):
    """Determines the winner based on standard Rock-Paper-Scissors rules."""
    if user == computer:
        return "tie"
    elif (user == 'rock' and computer == 'scissors') or \
        (user == 'paper' and computer == 'rock') or \
        (user == 'scissors' and computer == 'paper'):
        return "user"
    else:
        return "computer"
```

```
def get_game_mode():
    """Allows the user to select the game mode: Best of 3 or Best of 5 rounds."""
    while True:
        try:
            rounds = int(input("Choose game mode: Best of 3 or Best of 5? (Enter 3 or 5): "))
            if rounds in [3, 5]:
                return rounds
            else:
                print("Invalid input. Please enter 3 or 5.")
        except ValueError:
            print("Invalid input. Please enter a numerical value (3 or 5).")

def play_game():
    """Manages the Rock-Paper-Scissors game with user-selected settings."""
    print("Welcome to Rock, Paper, Scissors!")
    rounds = get_game_mode()  # User selects game mode
    user_score = 0
    computer_score = 0

    for _ in range(rounds):
        user_choice = get_user_choice()  # Get user input
        computer_choice = get_computer_choice()  # Get computer's choice
        print(f"Computer chose: {computer_choice}")
        winner = determine_winner(user_choice, computer_choice)  # Determine winner
        if winner == "user":
            print("You win this round!")
            user_score += 1
        elif winner == "computer":
            print("Computer wins this round!")
            computer_score += 1
        else:
            print("This round is a tie!")

        print(f"Current Score -> You: {user_score} | Computer: {computer_score}\n")

    print("Game Over!")
    if user_score > computer_score:
        print("Congratulations! You won the game!")
    elif computer_score > user_score:
        print("Better luck next time! The computer wins the game.")
    else:
        print("It's a draw!")

    print("Thank you for playing!")
    print("\"The only way to do great work is to love what you do.\" - Steve Jobs")

# Execute the game only if the script is run directly
if _name__== "_main_":
    play_game()
```

5. Screenshots Output



```
Welcome to Rock, Paper, Scissors!
Choose game mode: Best of 3 or Best of 5? (Enter 3 or 5): 3
Enter rock, paper, or scissors: rock
Computer chose: scissors
You win this round!
Current Score -> You: 1 | Computer: 0

Enter rock, paper, or scissors: paper
Computer chose: rock
You win this round!
Current Score -> You: 2 | Computer: 0

Enter rock, paper, or scissors: rock
Computer chose: scissors
You win this round!
Current Score -> You: 3 | Computer: 0

Game Over!
Congratulations! You won the game!
Thank you for playing!
```



```
Welcome to Rock, Paper, Scissors!
Choose game mode: Best of 3 or Best of 5? (Enter 3 or 5): 3
Enter rock, paper, or scissors: rock
Computer chose: scissors
You win this round!
Current Score -> You: 1 | Computer: 0

Enter rock, paper, or scissors: paper
Computer chose: rock
You win this round!
Current Score -> You: 2 | Computer: 0

Enter rock, paper, or scissors: rock
Computer chose: scissors
You win this round!
Current Score -> You: 3 | Computer: 0

Game Over!
Congratulations! You won the game!
Thank you for playing!
```

Conclusion

The Rock-Paper-Scissors game was successfully implemented using Python. The program follows structured programming principles, ensuring modularity, input validation, and a user-friendly experience. This implementation serves as a basic yet effective demonstration of decision-making, randomization, and user interaction in Python.

Future Enhancements

- Implement a GUI version using Tkinter or PyQt.

- Add a multiplayer mode.

- Introduce an AI-based strategy for the computer opponent.

References

- Python Official Documentation: https://docs.python.org/3/

- Random Module Documentation: https://docs.python.org/3/library/random.html