KIET GROUP OF INSTITUTIONS

Introduction to AI

MSE-1

Problem-Simple Game AI for Rock-Paper-Scissors

Name-Sudhakar Kumar

Roll No.-202401100400190

• Introduction

The Rock-Paper-Scissors game is a classic hand game played between two participants. In this project, we implemented a simple text-based version using Python, where a user competes against the computer. The program allows multiple rounds of play and determines the winner based on standard game rules.

The main objectives of this project are:

- To implement basic game logic using Python functions.
- To use the random module for generating the computer's choice.
- To handle user input and provide appropriate feedback.

Methodology

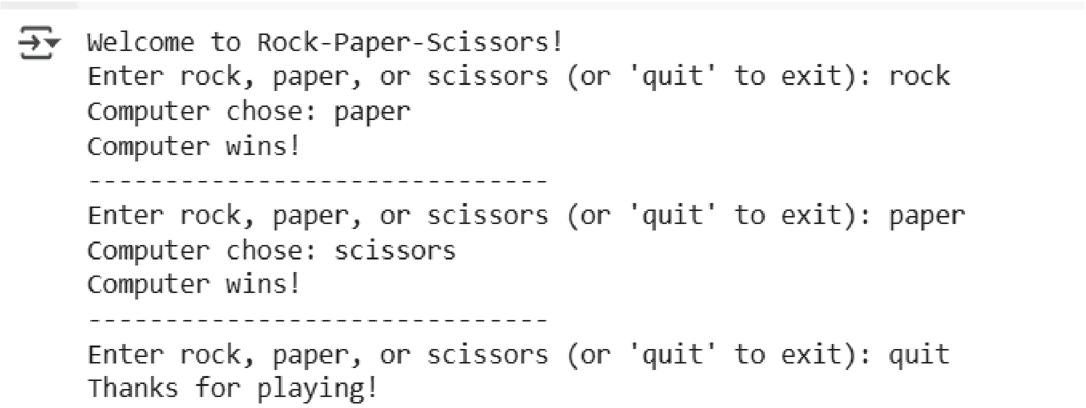
The project follows a structured approach:

1. Generating Computer's Move:
   - The random.choice() function is used to randomly select between "rock," "paper," or "scissors."
2. Determining the Winner:
   - A function compares the player's and computer's choices using the game's predefined rules.
3. User Interaction:
   - The user is prompted to enter their choice.
   - If the user enters "quit," the game ends.
   - If an invalid input is given, the user is asked to try again.
4. Loop for Multiple Rounds:
   - The game continues until the player decides to quit.

Code Typed

```
import random

# Function to randomly select the computer's move
def get_computer_choice():
    # The computer randomly chooses between 'rock', 'paper', or
'scissors'
    return random.choice(["rock", "paper", "scissors"])

# Function to determine the winner of the game based on player and
computer choices
def determine_winner(player, computer):
    # Check if the player's and computer's choices are the same
    if player == computer:
        return "It's a tie!"
    # Check if the player wins (rock beats scissors, scissors beats
paper, paper beats rock)
    elif (player == "rock" and computer == "scissors") or \
         (player == "scissors" and computer == "paper") or \
         (player == "paper" and computer == "rock"):
        return "You win!"
    # If the player doesn't win, the computer wins
    else:
        return "Computer wins!"

# Main function to run the game
def main():
    print("Welcome to Rock-Paper-Scissors!")
    # Start an infinite loop to play multiple rounds
    while True:
        # Ask the player for their choice and convert it to lowercase
        player_choice = input("Enter rock, paper, or scissors (or 'quit' to exit): ").lower()

        # Exit the game if the player types 'quit'
        if player_choice == "quit":
            print("Thanks for playing!")
            break

        # Validate that the player's choice is valid
        if player_choice not in ["rock", "paper", "scissors"]:
            print("Invalid choice. Try again.")
            continue

        # Get the computer's choice
        computer_choice = get_computer_choice()

        # Print the computer's choice
        print(f"Computer chose: {computer_choice}")

        # Determine the winner and print the result
        result = determine_winner(player_choice, computer_choice)
        print(result)

        # Print a separator line to make the output easier to read
        print("-" * 30)

# Run the main function when the script is executed
if __name__ == "__main__":
    main()
```

- # Output



```
Welcome to Rock-Paper-Scissors!
Enter rock, paper, or scissors (or 'quit' to exit): rock
Computer chose: paper
Computer wins!
------------------------------
Enter rock, paper, or scissors (or 'quit' to exit): paper
Computer chose: scissors
Computer wins!
------------------------------
Enter rock, paper, or scissors (or 'quit' to exit): quit
Thanks for playing!
```

Conclusion

- This project successfully implements a simple Rock-Paper-Scissors game using Python. It demonstrates:
- The use of conditional statements for game logic.
- User input handling and validation.
- Random selection using the random module.
- Further improvements could include adding a graphical user interface (GUI) using Tkinter or expanding the game with additional features like score tracking.