

# **PROBLEM STATEMENT**

## Simple Game AI for Rock-Paper-Scissors

**Name:** VANSH AGARWAL

**Roll No:** 202401100300273

**Course:** Introduction to A.I.

**Institution:** KIET Group of Institutions

# Introduction

Rock-Paper-Scissors is a well-known hand game usually played between two players, where each player simultaneously forms one of three shapes with an outstretched hand. The game follows simple rules:

- Rock beats Scissors
- Scissors beats Paper
- Paper beats Rock

In this project, we have implemented an AI-powered Rock-Paper-Scissors game where the AI learns from user moves and adapts its strategy in real-time. Unlike traditional random-based AI, this version improves as the game progresses by analyzing the player's

move history and adjusting its choices accordingly.

# Methodology

To develop this AI, we followed the following steps:

1. **Game Logic Implementation:** The core logic of Rock-Paper-Scissors was implemented in Python.
2. **AI Decision-Making:**
  - AI initially chooses moves randomly.

- As the game progresses, AI tracks the user's move history.
- AI uses probability-based decision-making to counter the most frequently used user moves.

3. **User Interaction:** The program takes user input and processes AI's move in real-time.

4. **Result Calculation:** The system determines the winner based on the standard game rules and updates the scores.

5. **Data Storage for AI Learning:** The AI maintains a count of the user's past choices to improve future decisions.

6. **Final Score Display:** When the user exits, the final scores are displayed.

# Code

The following is the complete Python code for the project:

```
import random
```

```
class RockPaperScissorsAI:
```

```
    """Rock-Paper-Scissors game with AI that  
    learns from previous results."""
```

```
def __init__(self):  
    """Initialize scores and AI learning  
memory."""  
  
    self.user_score = 0  
  
    self.computer_score = 0  
  
    self.choices = ["rock", "paper", "scissors"]  
  
    self.ai_memory = {"rock": 0, "paper": 0,  
"scissors": 0}  
  
def get_ai_choice(self):  
    """  
  
    AI chooses based on past user choices.  
  
    The more a user picks a move, the more AI  
tries to counter it.  
  
    """  
  
    total_moves = sum(self.ai_memory.values())
```

```
if total_moves == 0:
    return random.choice(self.choices) #
First round is random

# Calculate probabilities based on past
choices

weights = [
    self.ai_memory["scissors"], # AI chooses
rock if user plays scissors more
    self.ai_memory["rock"], # AI chooses
paper if user plays rock more
    self.ai_memory["paper"] # AI chooses
scissors if user plays paper more
]

return random.choices(self.choices,
weights=weights)[0]
```

```
def determine_winner(self, user_choice,
ai_choice):

    """Determines the winner based on
    Rock-Paper-Scissors rules."""

    if user_choice == ai_choice:

        return "tie"

    elif (user_choice == "rock" and ai_choice ==
"scissors") or \

        (user_choice == "scissors" and ai_choice
== "paper") or \

        (user_choice == "paper" and ai_choice
== "rock"):

        return "user"

    else:

        return "computer"

def play_game(self):
```



**""""Runs the Rock-Paper-Scissors game in a loop until the user exits.""""**

**print("Welcome to the Adaptive AI  
Rock-Paper-Scissors Game!")**

**while True:**

**user\_choice = input("\nEnter rock, paper,  
or scissors (or 'exit' to quit): ").lower()**

**if user\_choice == "exit":**

**self.display\_final\_score()**

**break**

**if user\_choice not in self.choices:**

**print("Invalid choice! Please enter rock,  
paper, or scissors.")**

**continue**

```
ai_choice = self.get_ai_choice()
```

```
print(f"AI chose: {ai_choice}")
```

```
winner =
```

```
self.determine_winner(user_choice, ai_choice)
```

```
if winner == "tie":
```

```
    print("It's a tie!")
```

```
elif winner == "user":
```

```
    print("You win this round!")
```

```
    self.user_score += 1
```

```
else:
```

```
    print("AI wins this round!")
```

```
    self.computer_score += 1
```

```
# Store user choice in AI memory to  
improve next decisions
```

```
self.ai_memory[user_choice] += 1
```

```
print(f"Score -> You: {self.user_score} |  
AI: {self.computer_score}")
```

```
def display_final_score(self):  
    """Displays the final score when the user  
    exits the game."""  
    print("\nFinal Score:")  
    print(f"User: {self.user_score} | AI:  
{self.computer_score}")  
    print("Thanks for playing!")
```

```
# Run the game
```

```
if __name__ == "__main__":  
    game = RockPaperScissorsAI()  
    game.play_game()
```

**Output/Results:** Below is an example output of the program execution.

```
Welcome to the Adaptive AI Rock-Paper-Scissors Game!

Enter rock, paper, or scissors (or 'exit' to quit): ROCK
AI chose: scissors
You win this round!
Score -> You: 1 | AI: 0

Enter rock, paper, or scissors (or 'exit' to quit): ROCK
AI chose: paper
AI wins this round!
Score -> You: 1 | AI: 1

Enter rock, paper, or scissors (or 'exit' to quit): EXIT

Final Score:
User: 1 | AI: 1
Thanks for playing!
```

# **References/Credits**

- **Python Official Documentation:**  
<https://docs.python.org/3/>
  - **Random Module Documentation:**  
<https://docs.python.org/3/library/random.html>
  - **AI Learning Concept Based on Probability Distribution from past user inputs.**
-