

Simple Game AI for Rock-Paper-Scissors

Problem Statement:

The goal of this project is to develop an AI-based opponent for the classic Rock-Paper-Scissors (RPS) game. The AI should be able to make intelligent choices, analyze past moves, and enhance its decision-making over time. The implementation will explore different approaches, from random selection to predictive strategies using pattern recognition.

Name: [Amit Yadav]

Roll No.: [202401100300034]

Course: [Artificial Intelligence]

Institution: [Kiet group of institution]

Introduction

Rock-Paper-Scissors (RPS) is a simple game with three possible choices: rock, paper, and scissors. The game follows these rules:

- Rock beats Scissors.
- Scissors beat Paper.
- Paper beats Rock.

While the game may seem purely random, human players often follow patterns that can be exploited using AI. This project aims to implement a simple AI that can either randomly pick a move or analyze past patterns to make better choices.

Importance of Game AI:

1. **Understanding AI Decision Making:** Simple AI implementations help in understanding core decision-making techniques.
2. **Pattern Recognition:** The AI can analyze user inputs over multiple rounds.
3. **Reinforcement Learning Possibilities:** Future improvements can involve machine learning to enhance AI choices.

This report details the methodology, implementation, and results of the Rock-Paper-Scissors AI model.

Methodology

The project follows these structured steps to develop and test the AI:

1. **Game Rule Implementation:** The basic logic of RPS is coded in Python.
 2. **User Input Handling:** The user provides their choice via input prompts.
 3. **AI Decision Making:** The AI can select options using:
 - **Random Selection:** The simplest approach where the AI randomly selects Rock, Paper, or Scissors.
 - **Pattern Recognition (Advanced Enhancement):** The AI tracks user inputs to predict their next move.
 4. **Determine Winner:** The program compares the choices and declares the winner.
 5. **Result Display:** The outcome is displayed after each round.
 6. **Enhancements (Optional):** Future versions may use reinforcement learning to improve AI performance.
-

Code

```
import random
```

```
def get_ai_move():
```

```
    return random.choice(['rock', 'paper', 'scissors'])
```

```
def determine_winner(player_move, ai_move):
```

```
    if player_move == ai_move:
```

```
        return "It's a tie!"
```

```
    elif (player_move == 'rock' and ai_move == 'scissors') or \
```

```
         (player_move == 'scissors' and ai_move == 'paper') or \
```

```
         (player_move == 'paper' and ai_move == 'rock'):
```

```
        return "You win!"
```

```
    else:
```

```
        return "AI wins!"
```

```
def play_game():
```

```
print("Welcome to Rock-Paper-Scissors!")

player_move = input("Enter your move (rock, paper, or scissors): ").lower()

if player_move not in ['rock', 'paper', 'scissors']:
    print("Invalid move! Please enter 'rock', 'paper', or 'scissors'.")
    return

ai_move = get_ai_move()
print(f"AI chose: {ai_move}")
result = determine_winner(player_move, ai_move)
print(result)

play_game()
```

Output/Result

Example Run:

Enter rock, paper, or scissors: rock

AI chose: scissors

You win!

(Screenshots of actual output should be attached here.)

Findings:

- The AI successfully plays the game based on random choice.
 - Results are fair and follow the game rules.
 - Future versions can implement pattern recognition to predict user choices.
-

References/Credits

- Python Random Module: <https://docs.python.org/3/library/random.html>

- Game Theory Basics: https://en.wikipedia.org/wiki/Rock_paper_scissors
- AI Decision Making Concepts: <https://towardsdatascience.com/reinforcement-learning-for-rock-paper-scissors-9b06f51f2c41>

This report outlines the development of a simple AI for Rock-Paper-Scissors using Python. Future enhancements may include machine learning models to predict user behavior and improve AI performance.

End of Report