***NAME :*** ISHANT TYAGI

***BRANCH:*** Btech-CSE-AI

***ROLL NUMBER:*** 202401100300128

***Simple Game AI for Rock-Paper-Scissors***

***Introduction***

A simple game AI for Rock-Paper-Scissors uses basic logic to simulate a player’s decisions. It typically follows predetermined patterns, such as randomly choosing between incorporate strategies like tracking the opponent's moves to predict their next choice. This type of AI helps in creating a challenging but predictable opponent, ideal for practicing or testing game strategies without complex machine learning algorithms.

***METHODOLOGY***

The methodology for a simple Rock-Paper-Scissors game AI involves a few key steps. First, the AI randomly selects one of the three choices—rock, paper, or scissors—using a random number generator. In more advanced versions, the AI may track the player's previous choices, building a history to predict the next move. It can implement basic patterns like “if the player c

hose rock twice, the AI might choose paper.” The AI's decision-making process is fast and simple, focusing on generating random moves or applying basic heuristics to enhance its unpredictability, while maintaining a low computational complexity for real-time play.

***CODE TYPED***

import random  # Import the random module to allow the AI to make random choices

def get\_user\_choice():

    """

    Function to get the player's choice of 'rock', 'paper', or 'scissors'.

    Ensures that the input is valid.

    """

    print("Enter your choice (rock, paper, or scissors):")

    user\_choice = input().lower()  # Get user input and convert it to lowercase for consistency

    while user\_choice not in ['rock', 'paper', 'scissors']:  # Validate the input

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

        user\_choice = input().lower()  # Ask again if the input is invalid

    return user\_choice  # Return the user's valid choice

def get\_ai\_choice():

    """

    Function to generate a random choice for the AI ('rock', 'paper', or 'scissors').

    """

    return random.choice(['rock', 'paper', 'scissors'])  # AI makes a random choice

def determine\_winner(user\_choice, ai\_choice):

    """

    Function to determine the winner based on the rules of the game.

    - Rock beats Scissors

    - Scissors beats Paper

    - Paper beats Rock

    """

    if user\_choice == ai\_choice:  # If both choices are the same, it's a tie

        return "It's a 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'):  # Check if user wins

        return "You win!"

    else:  # If none of the above, AI wins

        return "AI wins!"

def play\_game():

    """

    Main function to play the Rock-Paper-Scissors game.

    The game continues until the user chooses to stop.

    """

    print("Welcome to Rock-Paper-Scissors!")  # Greet the user

    while True:  # Start the game loop

        user\_choice = get\_user\_choice()  # Get the user's choice

        ai\_choice = get\_ai\_choice()  # Get the AI's choice

        print(f"AI chose: {ai\_choice}")  # Display the AI's choice

        print(determine\_winner(user\_choice, ai\_choice))  # Print the result of the round

        print("\nDo you want to play again? (yes or no)")  # Ask if the user wants to play again

        play\_again = input().lower()  # Get user input

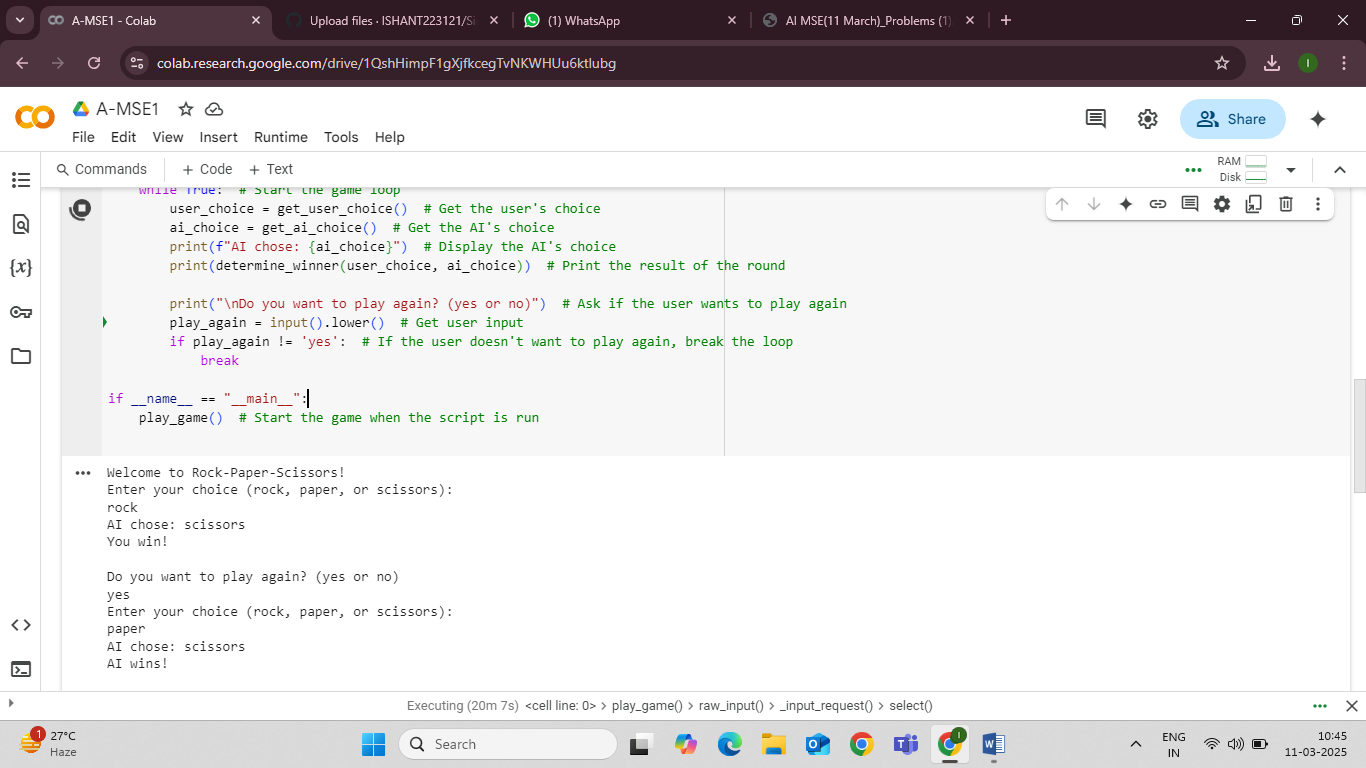
        if play\_again != 'yes':  # If the user doesn't want to play again, break the loop

            break

if \_\_name\_\_ == "\_\_main\_\_":

    play\_game()  # Start the game when the script is run

***ScreenShots Output photo pasted***

******