

|  |
| --- |
| **RAJALAKSHMI INSTITUTE OF TECHNOLOGY** |
| (An Autonomous Institution, Affiliated to Anna University, Chennai) |

**DEPARTMENT OF CSE (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)**

**ACADEMIC YEAR 2025 - 2026**

**SEMESTER III**

**ARTIFICIAL INTELLIGENCE LABORATORY**

**MINI PROJECT REPORT**

|  |  |
| --- | --- |
| **REGISTER NUMBER** | 2117240030010 |
| **NAME** | ARCHANA DV |
| **PROJECT TITLE** | Randomized AI Decision Algorithm for Rock–Paper–Scissors" |
| **DATE OF SUBMISSION** |  |
| **FACULTY IN-CHARGE** | **Mrs. M. Divya** |

**Signature of Faculty In-charge**

**INTRODUCTION**

* The project is based on the classic **Rock-Paper-Scissors** game between a user and an AI opponent.
* It demonstrates the use of **Artificial Intelligence** to make logical and randomized decisions.
* The AI uses a **randomized algorithm** to choose its moves, ensuring fairness and unpredictability.
* The game is developed using **Python** with a **Tkinter GUI**, providing an interactive user experience.
* It applies basic AI concepts such as **decision-making** and **probability-based logic**.
* The project aims to show how simple AI techniques can be used for **fun and engaging gameplay**.
* It also helps in understanding **human-computer interaction** and random behavior modeling.

**PROBLEM STATEMENT**

Artificial Intelligence (AI) enables systems to exhibit intelligent behavior by analyzing patterns, making decisions, and adapting to various scenarios.  
This project focuses on developing a **Randomized AI Decision Algorithm for Rock-Paper-Scissors** using Python and Tkinter, where the AI uses a combination of randomness and simple strategic logic to compete against a human player.  
The system integrates probability-based decision-making and adaptive responses to ensure that the AI’s moves remain unpredictable while maintaining a fair challenge.  
Unlike purely rule-based or learning-based models, the AI here employs **pseudo-random logic** and **pattern observation** to simulate human-like spontaneity and strategic guessing.  
A **Graphical User Interface (GUI)** built with Tkinter enhances user interaction by providing real-time visual feedback for each round and displaying results dynamically.  
Overall, the system demonstrates how **randomized algorithms and basic AI decision-making** can be combined to create an interactive, engaging, and intelligent game-playing experience.

**GOAL**

* o simulate an interactive **Rock-Paper-Scissors** game between the player and the AI.
* To demonstrate how **AI can make random yet fair decisions** in a simple game environment.
* To display real-time **results and scores** after each round (Win, Lose, or Draw).
* To provide a **user-friendly GUI** for smooth gameplay using buttons and messages.
* To show the AI’s ability to **analyze and respond instantly** to user inputs.
* To encourage user engagement through features like **Restart, Fullscreen, and Exit** options.

**THEORETICAL BACKGROUND:**

* Artificial Intelligence (AI) focuses on enabling machines to simulate human reasoning and decision-making. In traditional AI systems, logic-based reasoning and rule-based inference are used to make deterministic decisions. However, in dynamic and unpredictable environments like games, introducing **randomization** enhances realism and challenge.
* The **Rock-Paper-Scissors** game provides an excellent framework to demonstrate AI decision-making through a **randomized yet strategic approach**. The AI algorithm in this project uses **pseudo-random number generation** combined with **conditional logic** to select moves, ensuring variability and fairness. This technique allows the system to mimic human unpredictability while maintaining logical consistency.
* Unlike machine learning models, which rely on data-driven learning, this project employs **non-learning, probabilistic reasoning**, where decisions are based on **random selection and adaptive response**. This makes the algorithm **deterministic in structure but non-deterministic in behavior**, providing both transparency and engagement.
* A **Tkinter-based Graphical User Interface (GUI)** serves as the interaction medium between the user and the AI. It visually displays the moves, results, and scores, creating an interactive and enjoyable gameplay experience.
* Overall, the theoretical foundation lies in combining **AI decision logic**, **probability theory**, and **randomization techniques** to create an intelligent, fair, and user-friendly Rock-Paper-Scissors game..

**ALGORITHM EXPLANATION:**

1. **Input Choice:** The user selects Rock, Paper, or Scissors through the GUI.
2. **AI Decision:** The AI randomly chooses one option (Rock, Paper, or Scissors) using a randomization function.
3. **Comparison Logic:** The system compares the user’s choice and the AI’s choice based on game rules:
   * Rock beats Scissors
   * Scissors beats Paper
   * Paper beats Rock
4. **Result Determination:** The program decides whether the result is a *Win*, *Lose*, or *Draw*.
5. **Score Update:** The player’s and AI’s scores are updated accordingly.
6. **Output Display:** The result and updated scores are displayed on the GUI with text and color feedback.
7. **Repeat Option:** The user can restart or continue playing to test different outcomes.

**IMPLEMENTATION AND CODE**

**import tkinter as tk**

**import random**

**# --- Game Setup ---**

**choices = ["Rock", "Paper", "Scissors"]**

**emojis = {"Rock": "🪨", "Paper": "📄", "Scissors": "✂️"}**

**player\_score = 0**

**ai\_score = 0**

**fullscreen = False # default windowed mode**

**# --- Functions ---**

**def play(player\_choice):**

**global player\_score, ai\_score**

**ai\_choice = random.choice(choices)**

**if player\_choice == ai\_choice:**

**result = "It's a Tie 🤝"**

**color = "#FFD54F"**

**elif (player\_choice == "Rock" and ai\_choice == "Scissors") or \**

**(player\_choice == "Paper" and ai\_choice == "Rock") or \**

**(player\_choice == "Scissors" and ai\_choice == "Paper"):**

**result = "You Win 🎉"**

**color = "#4CAF50"**

**player\_score += 1**

**else:**

**result = "AI Wins 💻"**

**color = "#F44336"**

**ai\_score += 1**

**result\_label.config(**

**text=f"🧍 You: {emojis[player\_choice]} vs 🤖 AI: {emojis[ai\_choice]}\n\n{result}",**

**fg=color**

**)**

**score\_label.config(text=f"🏆 You: {player\_score} | AI: {ai\_score}")**

**def reset\_game():**

**global player\_score, ai\_score**

**player\_score = ai\_score = 0**

**score\_label.config(text="🏆 You: 0 | AI: 0")**

**result\_label.config(text="Make your move!", fg="white")**

**def toggle\_fullscreen():**

**global fullscreen**

**fullscreen = not fullscreen**

**root.attributes("-fullscreen", fullscreen)**

**if fullscreen:**

**fullscreen\_btn.config(text="⛶ Exit Fullscreen")**

**else:**

**fullscreen\_btn.config(text="⛶ Fullscreen")**

**def exit\_game():**

**root.destroy()**

**# --- GUI Setup ---**

**root = tk.Tk()**

**root.title("🎮 Rock Paper Scissors - AI Battle")**

**root.geometry("900x600")**

**root.configure(bg="#0F2027") # Deep gradient-like background**

**# --- Title Section ---**

**title\_label = tk.Label(root, text="⚡ ROCK PAPER SCISSORS ⚡",**

**font=("Comic Sans MS", 28, "bold"),**

**fg="#00E5FF", bg="#0F2027")**

**title\_label.pack(pady=20)**

**subtitle\_label = tk.Label(root, text="AI vs You — Let’s Play!",**

**font=("Poppins", 16, "italic"),**

**fg="#80DEEA", bg="#0F2027")**

**subtitle\_label.pack(pady=5)**

**# --- Score Section ---**

**score\_label = tk.Label(root, text="🏆 You: 0 | AI: 0",**

**font=("Consolas", 20, "bold"),**

**bg="#1C1C1C", fg="#00E676",**

**relief="ridge", bd=5, padx=20, pady=10)**

**score\_label.pack(pady=25)**

**# --- Result Display ---**

**result\_label = tk.Label(root, text="Make your move!",**

**font=("Consolas", 18, "bold"),**

**fg="white", bg="#0F2027")**

**result\_label.pack(pady=20)**

**# --- Buttons Frame ---**

**button\_frame = tk.Frame(root, bg="#0F2027")**

**button\_frame.pack(pady=40)**

**def create\_button(text, color, command):**

**return tk.Button(button\_frame, text=text, font=("Comic Sans MS", 18, "bold"),**

**width=12, height=2, bg=color, fg="white",**

**activebackground="#263238", activeforeground="white",**

**relief="raised", bd=5, cursor="hand2", command=command)**

**rock\_btn = create\_button("🪨 Rock", "#E91E63", lambda: play("Rock"))**

**paper\_btn = create\_button("📄 Paper", "#00BCD4", lambda: play("Paper"))**

**scissors\_btn = create\_button("✂️ Scissors", "#4CAF50", lambda: play("Scissors"))**

**rock\_btn.grid(row=0, column=0, padx=30)**

**paper\_btn.grid(row=0, column=1, padx=30)**

**scissors\_btn.grid(row=0, column=2, padx=30)**

**# --- Control Buttons ---**

**control\_frame = tk.Frame(root, bg="#0F2027")**

**control\_frame.pack(pady=20)**

**reset\_btn = tk.Button(control\_frame, text="🔄 Restart", font=("Poppins", 14, "bold"),**

**bg="#FFC107", fg="black", width=15, command=reset\_game, cursor="hand2")**

**reset\_btn.grid(row=0, column=0, padx=15)**

**fullscreen\_btn = tk.Button(control\_frame, text="⛶ Fullscreen", font=("Poppins", 14, "bold"),**

**bg="#2196F3", fg="white", width=15, command=toggle\_fullscreen, cursor="hand2")**

**fullscreen\_btn.grid(row=0, column=1, padx=15)**

**exit\_btn = tk.Button(control\_frame, text="❌ Exit", font=("Poppins", 14, "bold"),**

**bg="#F44336", fg="white", width=15, command=exit\_game, cursor="hand2")**

**exit\_btn.grid(row=0, column=2, padx=15)**

**# --- Footer ---**

**footer = tk.Label(root, text="© 2025 Rock Paper Scissors AI | Designed by You 💫",**

**bg="#0F2027", fg="#B2EBF2", font=("Arial", 10, "italic"))**

**footer.pack(side="bottom", pady=10)**

**# --- Keyboard Shortcut to Exit Fullscreen ---**

**root.bind("<Escape>", lambda event: [root.attributes("-fullscreen", False),**

**fullscreen\_btn.config(text="⛶ Fullscreen")])**

**root.mainloop()**

**Output**

The output shows the Rock-Paper-Scissors game developed using Python’s Tkinter. The user and AI make their choices, and the result (*You Win*, *AI Wins*, or *It’s a Tie*) is displayed based on game rules. The AI’s move is generated through a randomized algorithm, ensuring fairness and unpredictability. A dynamic scoreboard updates after each round, while the colorful GUI and control buttons provide an interactive and engaging user experience.

A screenshot of a video game

AI-generated content may be incorrect.

The output shows the Rock-Paper-Scissors game interface where the user plays against the AI. The scoreboard displays both scores, and the chosen moves are shown after each round. In this example, the user selected *Scissors* while the AI chose *Rock*, resulting in **AI Wins**. The buttons allow the user to select moves, restart the game, or exit. The GUI provides an interactive and user-friendly gaming experience.

A screenshot of a video game

AI-generated content may be incorrect.

The output shows the Rock-Paper-Scissors game screen where both the user and the AI selected Paper. Since both choices are the same, the result is displayed as “It’s a Tie.” The scoreboard indicates the current scores — the user has 1 point, and the AI has 0 points. The colorful buttons (Rock, Paper, Scissors) allow the player to choose moves, while Restart, Fullscreen, and Exit provide easy game control through the Tkinter-based GUI**.**

**RESULTS AND FUTURE ENHANCEMENT**

**Result**

The output displays the working of the **Randomized AI Decision Algorithm for Rock-Paper-Scissors** game, where the AI competes with the user using logical and random decision-making.  
Each round shows results such as **“You Win,” “AI Wins,”** or **“It’s a Tie”** based on the moves chosen by both sides.  
The AI’s move is generated through a **randomized algorithm**, making every game unique and unpredictable.  
The system follows the standard Rock-Paper-Scissors rules to determine the winner after each selection.  
A **Tkinter-based GUI** provides a colorful and interactive interface for smooth gameplay.  
The scoreboard updates dynamically to reflect the ongoing match status.  
The restart and exit options allow players to easily manage new sessions.  
Overall, the result proves that the AI can make autonomous, fair, and intelligent decisions, creating an engaging gaming experience.

**Future Enhancements**

* + **Multiplayer Mode:** Add an option for two users to play against each other locally or online.
  + **Smarter AI:** Improve AI decision-making using predictive algorithms or reinforcement learning.
  + **Difficulty Levels:** Include Easy, Medium, and Hard levels to make the game more engaging.
  + **Voice Commands:** Enable players to make their moves using speech recognition.
  + **Leaderboard System:** Track and display high scores of players over multiple sessions.
  + **Animated Interface:** Add smooth transitions, emojis, and effects for better visual appeal.
  + **Sound Effects:** Introduce victory, tie, and move sounds for an immersive experience.
  + **Statistics Feature:** Show player performance graphs and win rates for analysis.

|  |  |
| --- | --- |
| **Git Hub Link of the project and report** | **https://github.com/Archana200612/ROCK-PAPER-** |

**References**

1. Russell, S., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach (4th Edition).* Pearson Education.
2. GeeksforGeeks – *Rock Paper Scissors AI using Python.* Available at: <https://www.geeksforgeeks.org/>
3. W3Schools – *Python Random Module Tutorial.* Available at: <https://www.w3schools.com/python/module_random.asp>
4. Towards Data Science – *Building Simple AI Games with Python.* Available at: <https://towardsdatascience.com/>
5. Python Official Documentation – *Tkinter GUI Programming.* Available at: <https://docs.python.org/3/library/tkinter.html>