

|  |
| --- |
| **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** | 2117240030054 |
| **NAME** | JEEVA M |
| **PROJECT TITLE** | NUMBER GUESSING GAME |
| **DATE OF SUBMISSION** |  |
| **FACULTY IN-CHARGE** | **Mrs. M. Divya** |

**Signature of Faculty In-charge**

**INTRODUCTION**

* Artificial Intelligence (AI) is the simulation of human intelligence in machines that can think, learn, and make decisions.It enables computers to solve problems, recognize patterns, and adapt from experience.AI concepts include machine learning, reasoning, and natural language processing.
* This project, **Number Guessing Game**, demonstrates how an AI can make intelligent guesses using logic and feedback.  
  It applies the **binary search concept** to efficiently predict the user’s chosen number.  
  The project showcases how AI reasoning and decision-making can be used to solve simple interactive problems.

**PROBLEM STATEMENT**

* To design a program that enables an AI to correctly guess a number chosen by the user between 1 and 100, using logical feedback such as “higher,” “lower,” or “correct” to minimize the number of attempts.

**GOAL**

* The expected result is that the AI accurately identifies the user’s chosen number in the fewest possible attempts.  
  Possible outcomes include:
* The AI correctly guesses the number using logical deductions.
* The user provides consistent feedback, leading to an efficient solution.
* Inconsistent feedback may cause repeated or incorrect guesses.

**THEORETICAL BACKGROUND**

* + The project is based on the concept of **binary search**, an efficient algorithm used to find a target value by repeatedly dividing the search range in half.  
    In this game, the AI guesses a number and adjusts its range based on user feedback (“higher” or “lower”).  
    This approach demonstrates **AI reasoning**, where decisions are made logically using available information to reach the correct result with minimal attempts.

**LITERATURE SURVEY:**

* + **Binary Search (Deterministic):** Standard algorithm for ordered search; guarantees O(log⁡n)O(\log n)O(logn) guesses by halving the interval each step — optimal for this exact interactive setting.
  + **Linear / Sequential Search:** Simple brute-force approach (1 → 100); requires O(n)O(n)O(n) guesses in worst case — educational but inefficient.
  + **Randomized Guessing:** Random selection with feedback; can be faster on average in some noisy scenarios but lacks the worst-case guarantees of binary search.
  + **Information-Theoretic / Entropy Methods:** Choose guesses to maximize expected information gain (minimize remaining entropy); generalizes binary search for uneven priors.
  + **Learning-based Approaches:** Reinforcement learning or Bayesian inference can adapt to user patterns or noisy feedback, useful if feedback is inconsistent or biased.

**ALGORITHM EXPLANATION WITH EXAMPLE**

* The AI starts with a range of **1 to 100** and guesses the midpoint (50).
* If the user says **“higher,”** the new range becomes 51–100.
* If the user says **“lower,”** the range becomes 1–49.  
  This process continues until the AI guesses the correct number.  
  For example, if the user’s number is **73**, the AI might guess 50 → 75 → 62 → 68 → 71 → 73 in just a few steps.

**IMPLEMENTATION AND CODE**

**def ai\_guess\_number():**

**print("Think of a number between 1 and 100. The AI will try to guess it!")**

**low = 1**

**high = 100**

**attempts = 0**

**while low <= high:**

**guess = (low + high) // 2**

**attempts += 1**

**print(f"AI guesses: {guess}")**

**feedback = input("Is your number higher, lower, or correct? ").lower()**

**if feedback == "correct":**

**print(f"AI guessed your number {guess} in {attempts} tries!")**

**break**

**elif feedback == "higher":**

**low = guess + 1**

**elif feedback == "lower":**

**high = guess - 1**

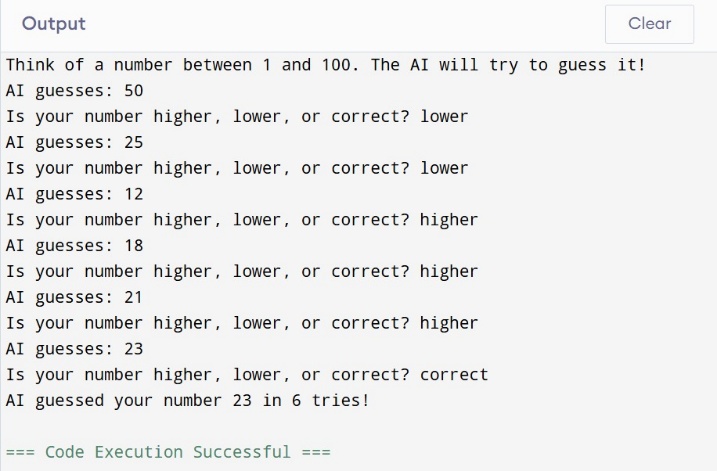
**else:**

**print("Please respond with 'higher', 'lower', or 'correct'.")**

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

**ai\_guess\_number()**

**OUTPUT**



**RESULTS AND FUTURE ENHANCEMENT**

* **Results:**  
  The AI successfully identifies the user’s chosen number within a minimal number of attempts using logical feedback.  
  It demonstrates the effectiveness of the binary search algorithm in decision-making and problem-solving.
* **Future Enhancements:**
* Add a **graphical user interface (GUI)** for better user interaction.
* Enable the AI to **learn from multiple sessions** to predict more efficiently.
* Expand the range and include **difficulty levels** or **randomized guessing strategies** for advanced gameplay.

|  |  |
| --- | --- |
| **Git Hub Link of the project and report** | **https://github.com/jeeva287/number-guessing-game-in-AI/commit/aace41b4aaf87db9090376cf01ce2c193d5e92b7** |

**REFERENCES**

* **Book:** “Artificial Intelligence: A Modern Approach” by Stuart Russell and Peter Norvig – for understanding AI reasoning and decision-making.
* **Website:** GeeksforGeeks – *“Binary Search Algorithm Explained”* (https://www.geeksforgeeks.org/binary-search/) – for algorithm logic and pseudocode.
* **E-Book:** “Python Programming for the Absolute Beginner” by Michael Dawson – for implementing Python-based games.
* **Blog:** Real Python – *“Building Interactive Command-Line Games with Python”* (https://realpython.com/) – for user-input game logic.
* **Video:** “Binary Search Explained Visually” – YouTube channel *Computerphile* – for conceptual understanding of search optimization.