
Autonomous Vehicle Decision Simulator

Submitted by

Prathamesh

25MCI10270

Master of Computer Application (MCA)
University Institute of Computing (UIC)

Chandigarh University

Submitted to

Dr. Rinku Sharma

(Assistant Professor, University Institute of Computing)

In partial fulfilment for the award of the degree of



Session: Aug 2025 – Nov 2025

INDEX

| S. No. | Contents | Page No. |
|---------------|-----------------------------|-----------------|
| 1 | Acknowledgement | i |
| 2 | Introduction | 1 |
| 3 | Objectives of the Project | 2 |
| 4 | System Overview | 2 |
| 5 | System Architecture | 3 |
| 6 | Tools and Technologies Used | 4 |
| 7 | Skills Utilized | 4 |
| 8 | Code | 5 |
| 9 | Results and Output | 6 |
| 10 | Advantages of the System | 7 |
| 11 | Future Scope | 7 |
| 12 | Conclusion | 7 |
| 13 | Bibliography | 8 |

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to my guide Dr. Rinku Sharma, Assistant Professor, University Institute of Computing, Chandigarh University, for her invaluable guidance, constant encouragement, and expert supervision throughout the completion of this project. Her insightful feedback and technical suggestions motivated me to explore and learn beyond my comfort zone.

I am also grateful to Dr. Krishan Tuli (H.O.D, UIC) for his continued support and motivation during the project period. His leadership and commitment to academic excellence have been a source of inspiration.

My heartfelt thanks to Dr. Manisha Malhotra, Additional Director, UIC, for fostering a strong research-oriented environment and for providing the resources necessary to complete this work effectively.

Finally, I would like to extend my sincere thanks to my parents and friends for their continuous encouragement, patience, and moral support throughout the project. Their belief in me has been the driving force behind the successful completion of this work.

Date: 09-11-2025

Place: Chandigarh University, Mohali, Punjab

By: Prathamesh

Introduction

In the era of rapid technological advancement, Artificial Intelligence (AI) has emerged as a transformative force, reshaping industries and redefining automation. One of the most exciting and impactful applications of AI lies in the automotive industry, where the concept of autonomous vehicles (self-driving cars) has transitioned from science fiction to real-world innovation. Autonomous vehicles are designed to navigate safely, make intelligent decisions, and respond effectively to environmental conditions without human intervention.

The core of any autonomous driving system lies in its decision-making capability — the ability to interpret real-time data and select the most appropriate driving action. In real-world conditions, a vehicle continuously encounters challenges such as changing weather, unexpected obstacles, and the sudden appearance of pedestrians. Each of these factors requires careful analysis and instant response to ensure the safety of passengers and the public.

The Autonomous Vehicle Decision Simulator is a Python-based mini-project that aims to replicate this critical decision-making mechanism through a simplified, rule-based AI simulation. The system randomly generates real-world driving scenarios and determines suitable actions such as proceed, slow down, reroute, or stop, based on environmental inputs. Although the simulator operates in a text-based environment, it mirrors the logical framework used by real AI systems that guide self-driving cars.

This simulator demonstrates how artificial intelligence principles can be applied using basic programming logic to imitate human-like judgment in uncertain conditions. The project uses conditional reasoning to evaluate different driving parameters such as:

- Weather Conditions (clear, foggy, rainy, or stormy),
- Obstacles (none, car, tree, debris), and
- Pedestrian Detection (presence or absence).

By combining these factors, the program makes a safe and rational driving decision — prioritizing safety above all else. The reasoning behind each action is also displayed, helping users understand how AI interprets complex data before making a choice.

Objectives of the Project

The primary objectives of this project are:

1. To simulate AI-based driving decisions using conditional logic and randomized scenarios.
2. To create a rule-based system capable of adapting to environmental factors like weather, obstacles, and pedestrian detection.
3. To demonstrate the basic reasoning mechanism of autonomous driving systems.
4. To use Python programming constructs effectively, such as functions, conditionals, and randomization.
5. To provide an interactive simulation that prints reasoning behind each decision made by the AI.
6. To enhance understanding of AI-driven automation and decision support systems.

System Overview

The Autonomous Vehicle Decision Simulator is a simple yet intelligent Python application that mimics the decision-making process of a self-driving car. The system evaluates various driving conditions and selects an appropriate driving action based on safety and environmental factors.

Working Process:

1. A random driving scenario is selected from a predefined list.
2. The system analyses three key parameters:
 - Weather condition

-
- Presence of obstacles
 - Pedestrian detection
3. Based on these conditions, it determines whether to:
- Proceed normally
 - Slow down due to bad weather
 - Reroute to avoid obstacles
 - Stop immediately if a pedestrian is detected
4. The AI provides not only a decision but also the reasoning behind it, helping users understand the logic flow.

System Architecture

The project follows a simple modular architecture consisting of three key components:

1. Input Layer:
 - A predefined list of random driving scenarios (weather, obstacle, pedestrian presence).
 - Random selection simulates real-world unpredictability.
2. Decision Logic Layer:
 - The function `make decision()` applies conditional rules to determine actions.
 - Logical checks evaluate safety and visibility factors.
3. Output Layer:
 - Displays the chosen driving action.
 - Prints detailed reasoning for transparency and explainability.

Tools and Technologies

| Category | Tools / Technologies |
|----------------------|-----------------------------------|
| Programming Language | Python |
| IDE / Editor | Visual Studio Code / PyCharm |
| Libraries Used | random |
| System Type | Console-based Simulation |
| Operating System | Windows / Linux compatible |
| Version Control | Git & GitHub (for project backup) |

Skills Utilized

- Python Programming Fundamentals
- Decision-Making using Conditional Logic
- Function-Based Modular Design
- Random Data Generation (`random.choice()`)
- Debugging and Console Output Formatting
- Logical and Algorithmic Thinking
- AI Reasoning Conceptualization

Code

```
import random

# List of possible driving scenarios
scenarios = [
    {"weather": "clear", "obstacle": "none", "pedestrian": False},
    {"weather": "foggy", "obstacle": "tree", "pedestrian": True},
    {"weather": "rainy", "obstacle": "car", "pedestrian": False},
    {"weather": "clear", "obstacle": "none", "pedestrian": True},
```

```
{"weather": "stormy", "obstacle": "debris", "pedestrian": True}

]

# Function to make a driving decision based on the scenario

def make_decision(scenario):

    reasons = [] # To store reasoning behind the AI's decision

    action = "proceed" # Default action

    # Check weather condition

    if scenario["weather"] != "clear":

        reasons.append("Bad weather, visibility reduced.")

        action = "slow down"

    # Check for obstacles

    if scenario["obstacle"] != "none":

        reasons.append(f"Obstacle detected: {scenario['obstacle']}.")

        action = "reroute"

    # Check for pedestrians

    if scenario["pedestrian"]:

        reasons.append("Pedestrian ahead! Immediate stop required.")

        action = "stop"

    return action, reasons

# Main simulation function

def run_simulation():

    print("===== Autonomous Vehicle Decision Simulator =====\n")

    # Randomly select one scenario

    scenario = random.choice(scenarios)

    print("Scenario:", scenario)

    # AI makes a decision based on the chosen scenario

    action, reasons = make_decision(scenario)
```

```
print("\nDecision:", action.upper())
print("Reasoning:")
for r in reasons:
    print("-", r)
run_simulation()
```

Results and Output

Output: Example

Autonomous Vehicle Decision Simulator

Scenario: {'weather': 'foggy', 'obstacle': 'tree', 'pedestrian': True}

Decision: STOP

Reasoning:

- Bad weather, visibility reduced.
- Obstacle detected: tree.
- Pedestrian ahead! Immediate stop required.

This demonstrates how the AI prioritizes safety over mobility and always chooses the safest action based on environmental conditions.

Advantages of the System

- Demonstrates AI reasoning and conditional decision-making.
- Simple and easy to understand for students learning Python.
- Replicates real-world concepts of autonomous vehicle behaviour.
- Provides clear explanations for each AI action.
- Can be extended into advanced simulation models with sensors and ML.

Future Scope

1. Integration with sensor data for realistic simulations.

-
2. Implementation of machine learning models for adaptive decision-making.
 3. Addition of a graphical user interface (GUI) to visualize scenarios.
 4. Use of external APIs for real-time weather or traffic data.
 5. Expansion into a full-fledged self-driving simulation environment.

Conclusion

The Autonomous Vehicle Decision Simulator successfully models how AI-driven systems can evaluate multiple real-world parameters and make safe, logical decisions. It provides valuable insight into rule-based AI behaviour, helping beginners understand how autonomous driving systems operate at a conceptual level.

This project enhanced my understanding of Python programming, logic design, and AI reasoning principles. With further improvements, it could serve as an educational tool to demonstrate the working of autonomous systems in controlled environments.

Bibliography

1. Python Official Documentation – <https://docs.python.org/>
2. GeeksforGeeks – Python Conditional Statements and Loops
3. W3Schools – Python Random Module
4. Towards Data Science – AI Decision-Making Models
5. TutorialsPoint – Python Functions and Logic Programming Concepts
6. ChatGPT by OpenAI – Concept Structuring and Report Formatting Assistance