Major Project: PR1107

Topic:

Simulation of Autonomous Vehicle Movement using Python, Py-game, and Pathfinding Algorithms

By:

Asit kumar Giri (2021btech028)
Tirumala Sumith(2021btech114)

Index

Problem Statement

Due to the rapid growth of autonomous vehicle research and development worldwide, it is essential to comprehend the foundation on which self-driving cars make intelligent choices. An autonomous vehicle needs to not only adhere to a set route but also respond and adjust in real-time to its environment. This demands a precise understanding of the surroundings, effective route planning, and cognitive skills that emulate human "road awareness."

Decision-making responsibilities involve staying in lanes, switching lanes, passing slower vehicles, and yielding to avoid collisions. Software-based simulations offer a safe and effective method for developing a controlled understanding of these behaviours. The suggested project intends to emulate navigation and decision-making for autonomous vehicles using graph-based road networks integrated with shortest-path algorithms, thus offering academic understanding of the computational principles underlying these systems.

Objectives

- To create and execute a road simulation scenario utilizing Python and Py-game.
- To incorporate pathfinding methods like Dijkstra's Algorithm for determining the shortest and most secure paths between waypoints.
- To replicate self-governing actions like:
- Changing lanes when obstructed,
- Passing vehicles that are moving slower,
- Yielding or giving way to prevent collisions.
- To develop a system for detecting and avoiding collisions among multiple vehicles.
- To conceptually connect these software methods with hardware sensors (LiDAR, RADAR, Ultrasonic, Cameras) that facilitate such features in truly autonomous vehicles.
- To offer a learning resource that illustrates how logic and algorithms can mimic real-world autonomous decision-making.

Scope of Work

- The aim of this project is centered on academics and simulations. The upcoming tasks will be addressed:
- Road representation using a graph structure comprised of nodes (waypoints/junctions) and edges (routes).
- Vehicle agent path planning utilizing Dijkstra's algorithm for shortest routes.
- Simulation of vehicle behaviour incorporating lane management, overtaking, and yielding functions.
- Safety validation through a collision detection system in the simulation.
- Py-game visualization provides a clear interactive display of vehicles moving through a changing environment.
- Conceptual hardware integration: Describing how real sensors facilitate identical decision-making in autonomous vehicles in the real world.
- Real-world issues such as weather influences or sensor errors will not be included in the current scope to maintain concentration on algorithmic decision-making.

Methodology

- Environment Configuration:
    - Development of a 2D roadway system in Py-game.
    - Depiction of streets as graph edges and junctions as nodes.
- Navigating routes:
    - Utilizing the NetworkX library to execute Dijkstra's algorithm.
    - Vehicles dynamically determine the best routes within the graph.
- Logic of Decision:
    - Create logic for passing another vehicle when speed changes or obstacles are sensed.
    - Mechanism for changing lanes to replicate secure path transitions.
    - Yielding or pausing at junctions to show collaborative road usage.

- Simulating & Visualizing:
  - Vehicles (as agents/objects) navigate through the environment along planned routes.
  - Collision detection and prevention to guarantee lifelike movement.
  - Several vehicles implemented to evaluate interactions.

- Hardware Mapping (Conceptual):
  - LiDAR: Modelled as detection areas for objects and borders.
  - RADAR: Conceptually depicted for monitoring vehicle separation and relative velocity.
  - Ultrasonic Sensors: Designed for detecting at short distances.
  - Cameras: Lane identification and sign recognition detailed as potential integration areas.

Expected Outcome

- A complete operational simulation setting for autonomous vehicle guidance utilizing Python and Pygame.

- Routefinding paths calculated using Dijkstra's algorithm, showcasing navigation based on graphs.

- Representation of actual driving behaviors such as changing lanes, passing, and yielding.

- Showcasing methods for avoiding collisions.

- A conceptual connection between software simulation and hardware, linking the theoretical framework with practical autonomous vehicle technology.

Tools and Technology

- Coding Language: Python 3.x
- Libraries:
- Pygame – Designing and visualizing environments
- NetworkX – Graphs and shortest-path computation
- NumPy – Mathematical calculations
- Conceptual Mapping of Hardware:
- LiDAR, RADAR, ultrasonic sensors, and cameras.

Conclusion

This project demonstrates how computational methods and simulated logic can play a significant role in the development of autonomous driving systems. By modeling the movement and decision-making ability of vehicles inside a software environment, we can better understand the mechanisms that drive real-world self-driving cars.
While the scope is limited to algorithmic simulation, the conceptual connection with hardware sensors showcases how such systems can be extended to real-life applications. Thus, this project serves both as a learning tool and as a demonstration of how theory is linked to practice in the field of intelligent transportation systems.