Traffic Light Control System

Name: Yash Jangi
Roll No.: 202401100400216

Introduction

This project aims to simulate a simple yet effective AI-based traffic control system that dynamically adjusts traffic signals based on real-time vehicle counts. The system uses conditional logic to prioritize roads with higher traffic, ensuring optimal vehicle clearance and reducing congestion.

The simulation mimics real-world conditions by introducing randomness to vehicle movement and includes rules to prevent traffic lights from unfairly favoring one road over the other. The AI controller manages the lights using a count-based logic to ensure fair alternation while giving priority to the busier road when necessary.

Methodology

The system follows a structured approach to manage and simulate traffic flow:

1. **Traffic Simulation:**
    - Each road's traffic count is adjusted randomly to simulate real-life conditions.
    - Random values are used to mimic the arrival and departure of cars.
2. **AI Control Logic:**
    - The AI analyzes the traffic levels on both roads.
    - If one road has significantly more cars (based on a defined threshold), that road gets priority.
    - To prevent unfair favoritism, the system ensures no road receives more than two consecutive green signals unless the congestion is severe.
3. **Traffic Light Switching:**
    - The lights alternate based on car counts, ensuring fair traffic flow.
    - After a green light for a road, some cars are removed to simulate clearing the traffic.
4. **Loop and Delay:**
    - The system continuously updates traffic conditions and switches lights accordingly.
    - A delay is introduced to simulate real-time behavior.

Code

```
import random

import time
```

```
def simulate_traffic(road1_cars, road2_cars, road1On, max_cars=20):
    """Simulates traffic flow with gradual changes and a maximum limit."""
    if road1On:
        road2_cars += random.randint(-2, 2)
        road1_cars = random.randint(-1, 2)
    else:
        road2_cars = random.randint(-1, 3)
        road1_cars += random.randint(-2, 2)


    road1_cars = max(0, min(road1_cars, max_cars))
    road2_cars = max(0, min(road2_cars, max_cars))


    return road1_cars, road2_cars
def ai_control(road1_cars, road2_cars, road1On, road1_green_count=0, road2_green_count=0,
threshold=3):
    """Controls traffic lights, switching after 2 consecutive greens and clearing most cars."""
    if road1_cars > road2_cars + threshold and road1_green_count < 2:
        road1_green_count += 1
        road2_green_count = 0
        return "road1_green", road1_green_count, road2_green_count  # Return counts as well
    elif road2_cars > road1_cars + threshold and road2_green_count < 2:
        road2_green_count += 1
        road1_green_count = 0
        return "road2_green", road1_green_count, road2_green_count  # Return counts as well
    else:
        # Switch light after 2 consecutive greens or if traffic is relatively equal
        if road1_green_count >= 2:
            road1_green_count = 0
            road2_green_count = 1
            return "road2_green", road1_green_count, road2_green_count  # Return counts
        elif road2_green_count >= 2:
            road2_green_count = 0
            road1_green_count = 1
            return "road1_green", road1_green_count, road2_green_count  # Return counts
        else:  # If traffic is relatively equal, alternate the light
            road1_green_count, road2_green_count = (1, 0) if road1On else (0, 1)
            return "road1_green" if road1On else "road2_green", road1_green_count,
road2_green_count

# Initialize traffic conditions
road1_cars = 5
road2_cars = 5
road1On = True
road1_green_count = 0
road2_green_count = 0

# Start the simulation loop
while True:
    road1_cars, road2_cars = simulate_traffic(road1_cars, road2_cars, road1On)
    light_status, road1_green_count, road2_green_count = ai_control(road1_cars, road2_cars,
road1On, road1_green_count, road2_green_count)  # Update counts
    road1On = not road1On if light_status == "road2_green" else road1On

    print(f"Road1 cars: {road1_cars}, Road2 cars: {road2_cars}, Light status: {light_status}")

    # Clear most cars when the light turns green
    if light_status == "road1_green":
        road1_cars = random.randint(0, 2)    # Clear 3 to all cars
    else:  # light_status == "road2_green"
        road2_cars = random.randint(0, 2)   # Clear 3 to all cars

    time.sleep(2)  # Simulate time passing
```

Output/Result

```
Road1 cars: 0, Road2 cars: 4, Light status: road2_green

Road1 cars: 0, Road2 cars: 1, Light status: road2_green

Road1 cars: 2, Road2 cars: 1, Light status: road1_green

Road1 cars: 1, Road2 cars: 2, Light status: road1_green

Road1 cars: 0, Road2 cars: 4, Light status: road2_green

Road1 cars: 0, Road2 cars: 3, Light status: road2_green

Road1 cars: 0, Road2 cars: 3, Light status: road1_green

Road1 cars: 2, Road2 cars: 3, Light status: road1_green

Road1 cars: 1, Road2 cars: 2, Light status: road1_green

Road1 cars: 1, Road2 cars: 2, Light status: road1_green

Road1 cars: 0, Road2 cars: 4, Light status: road2_green

Road1 cars: 0, Road2 cars: 1, Light status: road2_green

Road1 cars: 1, Road2 cars: 0, Light status: road1_green

Road1 cars: 0, Road2 cars: 1, Light status: road1_green

Road1 cars: 2, Road2 cars: 0, Light status: road1_green

Road1 cars: 1, Road2 cars: 2, Light status: road1_green

Road1 cars: 2, Road2 cars: 2, Light status: road1_green
```