KIET GROUP OF INSTITUTIONS

ARTIFICIAL INTELLIGENCE

MID SEMESTER EXAMINATION-1

PROBLEM STATEMENT:

TRAFFIC LIGHT CONTROL SYSTEM

SUBMITTED BY:

NAME- DIVYAM SINGH

BRANCH- CSE (AI)

SECTION- B

ROLL NO- 30

UNIVERSITY ROL NO. - 202401100300104

INTRODUCTION

Traffic lights are essential for managing vehicle movement and ensuring road safety. This project aims to simulate a traffic light control system with moving vehicles using Python and Matplotlib. The simulation includes a visual representation of traffic lights changing between red, yellow, and green states while vehicles respond accordingly. This enhances the realism of the simulation and demonstrates basic traffic flow concepts.

Methodology

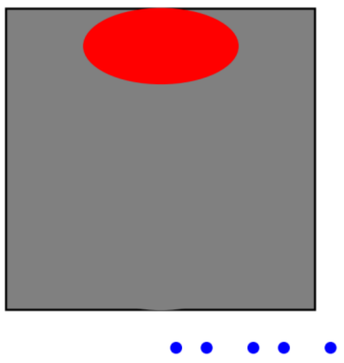
The implementation of the traffic light control system follows these steps:

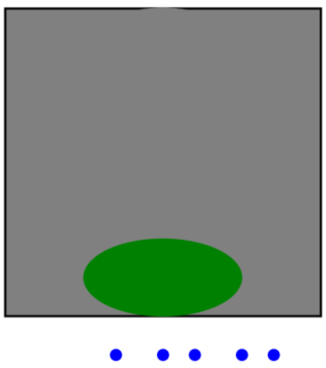
1. **Designing the Traffic Light Interface:** A traffic light is represented using Matplotlib, where a rectangle serves as the traffic light housing, and circles represent the red, yellow, and green signals.
2. **Vehicle Movement Simulation:** Cars are displayed as moving points along a road. When the light is green or yellow, cars move forward. When the light is red, they stop.
3. **State Management:** A finite state machine controls transitions between red, yellow, and green states.
4. **Timing Control:** The timings array dictates the duration of each light state.
5. **Continuous Looping:** The simulation continuously cycles through the light states and updates the vehicle positions dynamically using Matplotlib's FuncAnimation.

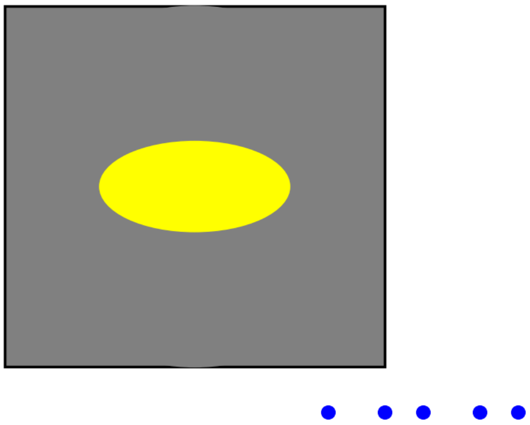
CODE:

```python
import time
import matplotlib.pyplot as plt
import matplotlib.patches as patches
from IPython.display import display, clear_output
import numpy as np

def draw_traffic_light(state, car_positions):
    fig, ax = plt.subplots(figsize=(6, 6))
    ax.set_xlim(0, 6)
    ax.set_ylim(0, 6)
    ax.set_xticks([])
    ax.set_yticks([])
    ax.set_frame_on(False)

    # Draw traffic light box
    box = patches.Rectangle((2, 1), 2, 4, linewidth=2, edgecolor='black',
facecolor='gray')
    ax.add_patch(box)

    # Draw lights
    colors = {'red': 'gray', 'yellow': 'gray', 'green': 'gray'}
    colors[state] = state  # Activate the current light

    red_light = patches.Circle((3, 4.5), 0.5, color=colors['red'])
    yellow_light = patches.Circle((3, 3), 0.5, color=colors['yellow'])
    green_light = patches.Circle((3, 1.5), 0.5, color=colors['green'])

    ax.add_patch(red_light)
    ax.add_patch(yellow_light)
    ax.add_patch(green_light)

    # Draw moving cars (dots)
    for x in car_positions:
        ax.plot(x, 0.5, 'bo', markersize=10)  # Cars as blue dots

    display(fig)
    plt.close(fig)

# Traffic light cycle
states = ['red', 'green', 'yellow']
timings = [3, 3, 2]  # Time delays in seconds
car_positions = np.linspace(-1, 0, 5)  # Initial positions of cars

while True:
    for state, delay in zip(states, timings):
        for _ in range(delay):
            clear_output(wait=True)

            if state == 'red':
                # Stop cars if red
                car_positions = car_positions
            else:
                # Move cars if green or yellow
                car_positions += 0.5  # Cars move forward
                car_positions = np.where(car_positions > 6, -1, car_positions)
# Reset if off screen

            draw_traffic_light(state, car_positions)
            time.sleep(1)
```

OUTPUT:







CONCLUSION:

Conclusion

This project successfully simulates a traffic light control system with moving vehicles using Python and Matplotlib. The implementation effectively demonstrates the coordination between traffic signals and vehicle motion. Future enhancements may include adding pedestrian crossings, real-time sensor-based controls, and more complex traffic patterns.

REFERENCES:

1. Python Official Documentation - https://docs.python.org/
2. Matplotlib Documentation - https://matplotlib.org/stable/contents.html
3. NumPy Documentation - https://numpy.org/doc/