NAME: PRATEEK DUBEY

DEPARTMENT: CSE-AIML

ROLL NO: 20241100400143

Date: 10-Mar-25

PROBLEM STATEMENT

AI – Traffic Light Simulation

SUBMITTED TO: MR. ABHISHEK SHUKLA SIR

INTRODUCTION

Traffic congestion and accidents are frequent problems in urban areas as a result of uncontrolled vehicle movement. Traffic lights are instrumental in controlling the movement of vehicles by indicating when to stop, prepare, and move. Nonetheless, knowing how these signals function and their timing is important for improved traffic control.

A traffic light simulation assists in the visualization of the light sequence and their durations, providing a smooth and secure traffic flow. The project shows a real-life traffic light system cycling through green, yellow, and red lights with a countdown timer and respective instructions.

Methodology

Approach to Solve the Problem

To simulate a traffic light system, we use **Python and Matplotlib** to create a visual representation of the signal lights. The methodology involves the following steps:

1. **Designing the Traffic Light Structure**
   - Using `matplotlib.patches` to draw a vertical traffic light with three circular lights.
   - Arranging the lights in the standard order: Red (top), Yellow (middle), and Green (bottom).
2. **Implementing the Light Change Cycle**
   - Using a loop to control the **sequential transition** between red, yellow, and green lights.
   - Assigning a **fixed duration** for each light:
     - 🔴 **Red Light** → 10 seconds (STOP)
     - 🟡 **Yellow Light** → 5 seconds (READY TO GO)
     - 🟢 **Green Light** → 10 seconds (GO)
   - Using `time.sleep()` to introduce a countdown timer.
3. **Updating the Display Dynamically**
   - Utilizing `IPython.display.clear_output()` to refresh the output in Google Colab.
   - Highlighting the **active light** while keeping others dimmed.
   - Displaying a **message** alongside the countdown timer to indicate the action for each light.
4. **Ensuring User-Friendly Interaction**
   - The program continuously loops, simulating real-world traffic signals.
   - The timer helps users **understand the duration** of each light phase.

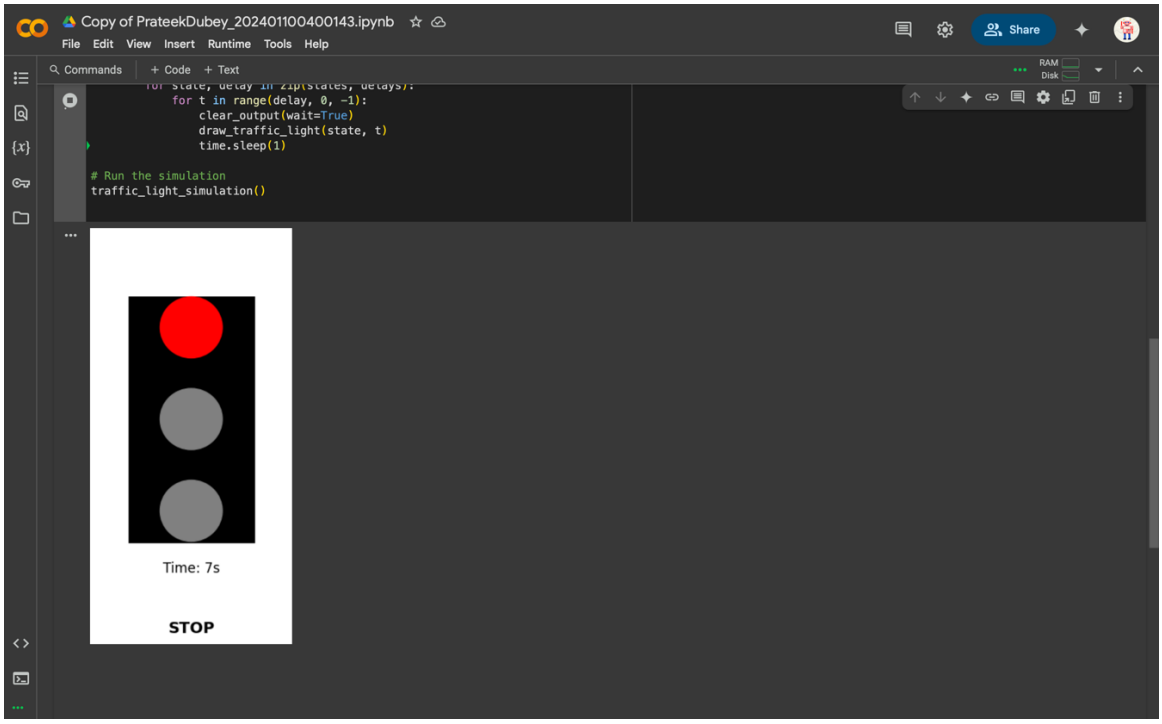
CODE:

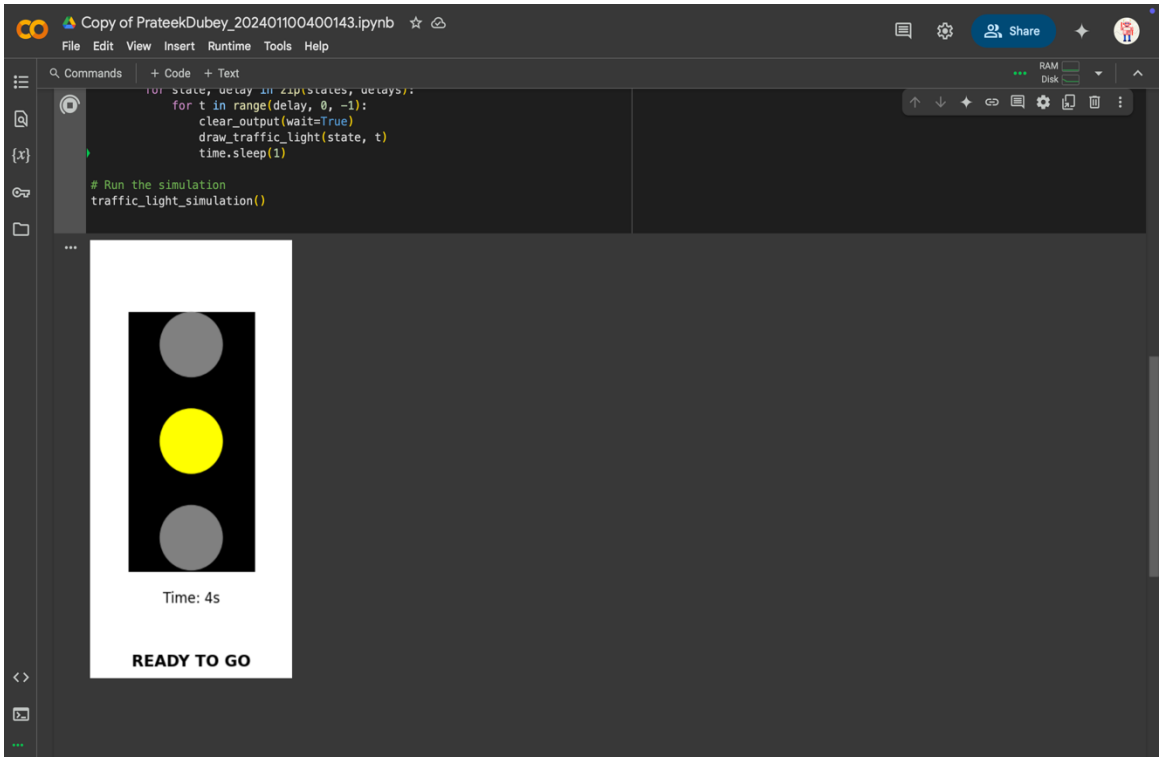
```
import matplotlib.pyplot as plt
import matplotlib.patches as patches
import time
from IPython.display import display, clear_output

# Function to draw the traffic light with timer and message
def draw_traffic_light(state, timer):
    fig, ax = plt.subplots(figsize=(3, 6))
    ax.set_xlim(0, 3)
    ax.set_ylim(0, 6)
    ax.set_xticks([])
    ax.set_yticks([])
    ax.set_frame_on(False)

    # Draw traffic light box
    box = patches.Rectangle((0.5, 1), 2, 4, linewidth=2,
edgecolor='black', facecolor='black')
    ax.add_patch(box)

    # Draw traffic lights (default gray)
```

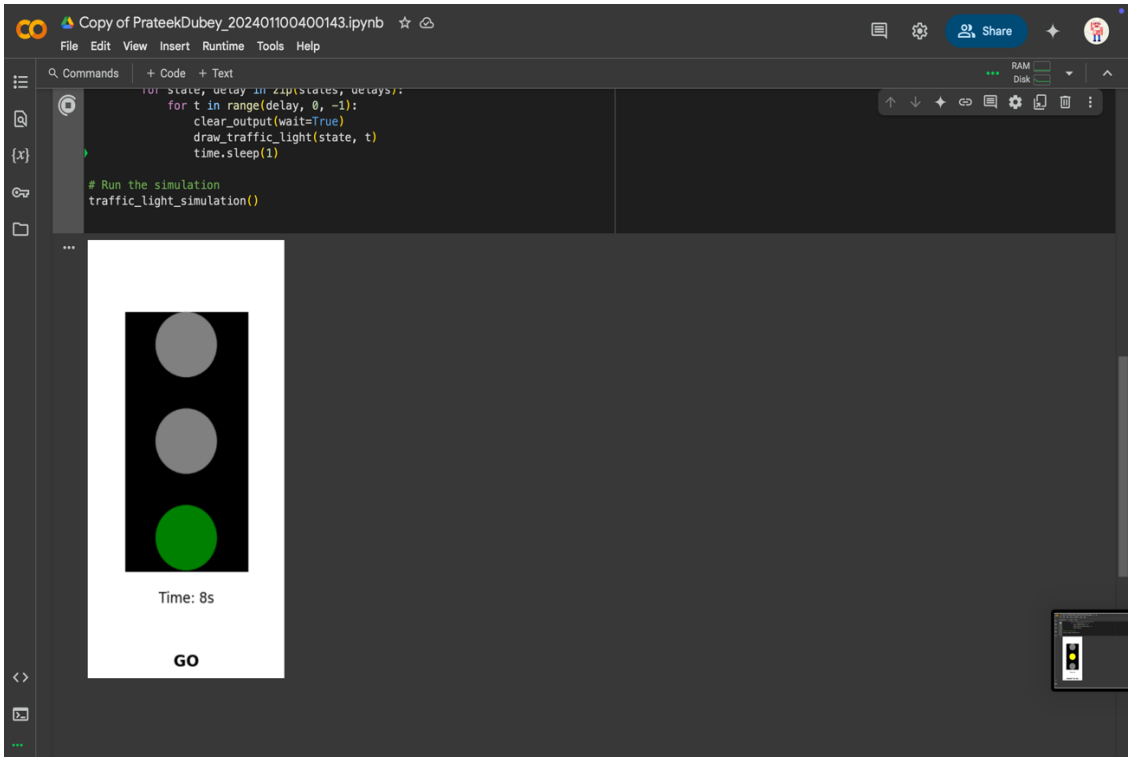
```
    colors = ['gray', 'gray', 'gray']
    message = ""
    if state == 0:
        colors[0] = 'red'   # Red light on
        message = "STOP"
    elif state == 1:
        colors[1] = 'yellow' # Yellow light on
        message = "READY TO GO"
    elif state == 2:
        colors[2] = 'green' # Green light on
        message = "GO"


    ax.add_patch(patches.Circle((1.5, 4.5), 0.5, color=colors[0])) # Red light
    ax.add_patch(patches.Circle((1.5, 3), 0.5, color=colors[1]))   # Yellow light
    ax.add_patch(patches.Circle((1.5, 1.5), 0.5, color=colors[2])) # Green light


    # Display Timer
    ax.text(1.5, 0.5, f'Time: {timer}s', fontsize=12, ha='center', color='black')
```

```
    # Display Message
    ax.text(1.5, -0.5, message, fontsize=14, ha='center',
color='black', fontweight='bold')

    plt.show()

# Function to change traffic light states with timer and message
def traffic_light_simulation():
    states = [0, 1, 2]  # Red -> Yellow -> Green
    delays = [10, 5, 10]  # Corresponding delays

    while True:
        for state, delay in zip(states, delays):
            for t in range(delay, 0, -1):
                clear_output(wait=True)
                draw_traffic_light(state, t)
                time.sleep(1)

# Run the simulation
traffic_light_simulation()
```

OUTPUT


Copy of PrateekDubey_202401100400143.ipynb
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text
RAM
Disk
for t in range(delay, 0, -1):
clear_output(wait=True)
draw_traffic_light(state, t)
time.sleep(1)
# Run the simulation
traffic_light_simulation()
Time: 7s
STOP


Copy of PrateekDubey_202401100400143.ipynb
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text
RAM
Disk
for t in range(delay, 0, -1):
clear_output(wait=True)
draw_traffic_light(state, t)
time.sleep(1)
# Run the simulation
traffic_light_simulation()
Time: 4s
READY TO GO



CREDITS:

ChatGpt