Traffic Light System

Project Report

Project Title: Traffic Light System Simulation

Problem Statement: To design a simple traffic light system simulation using Python, which displays traffic light sequences and visualizes light durations through graphs. The system should also allow user interaction to stop the process after each cycle.

Submitted by: Ayushmaan

University Roll no. : 202401100400070

Branch: CSE (AIML)

Date: 10th March, 2025

Introduction

Problem:

In urban traffic management, traffic lights are crucial for maintaining order and safety. This project aims to simulate a basic traffic light system where lights switch between Red, Yellow, and Green in a timed sequence. Additionally, graphical representations are provided to visualize light durations.

Objective:

- Create a traffic light simulation using Python.
- Display light changes with clear visual representation.
- Include graphs (bar chart, pie chart, and line graph) for data visualization.
- Allow user interaction to stop or continue the simulation after each iteration.

Methodology

Approach:

1. **Design the Light System:** Define three lights — Red, Yellow, and Green — each with a fixed duration.

2. **Console Display:** Use emojis to represent the lights and switch their states based on time intervals.

3. **Graphs:** Utilize the Matplotlib library to generate bar, pie, and line charts to show light durations visually.

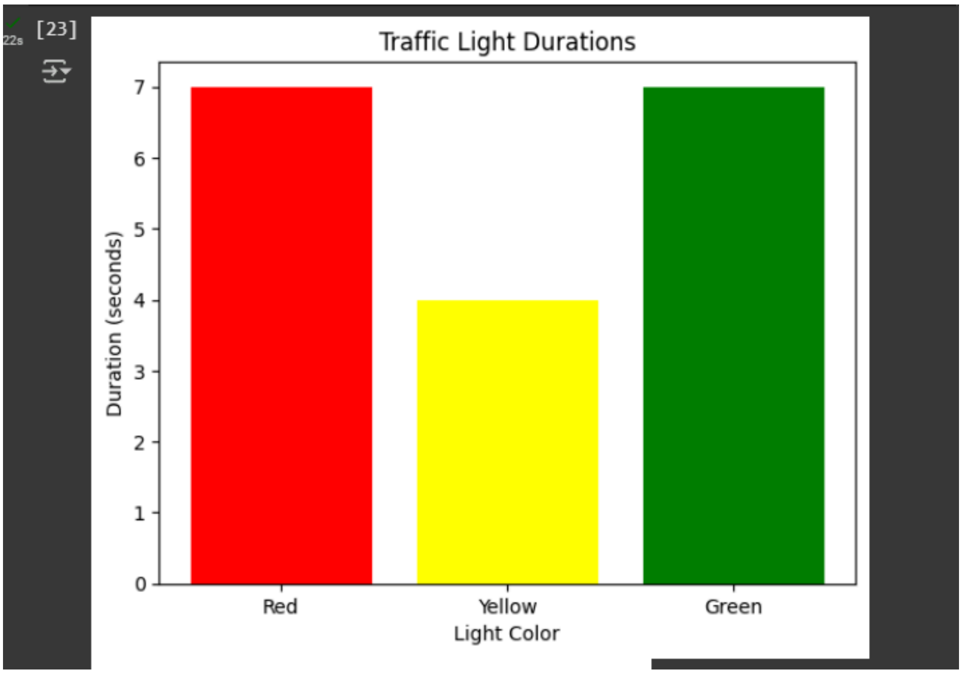
4. **User Interaction:** Implement a loop allowing users to stop or continue the system after each cycle.

5. **Testing:** Run and test the system to ensure smooth transitions and correct graph plotting.

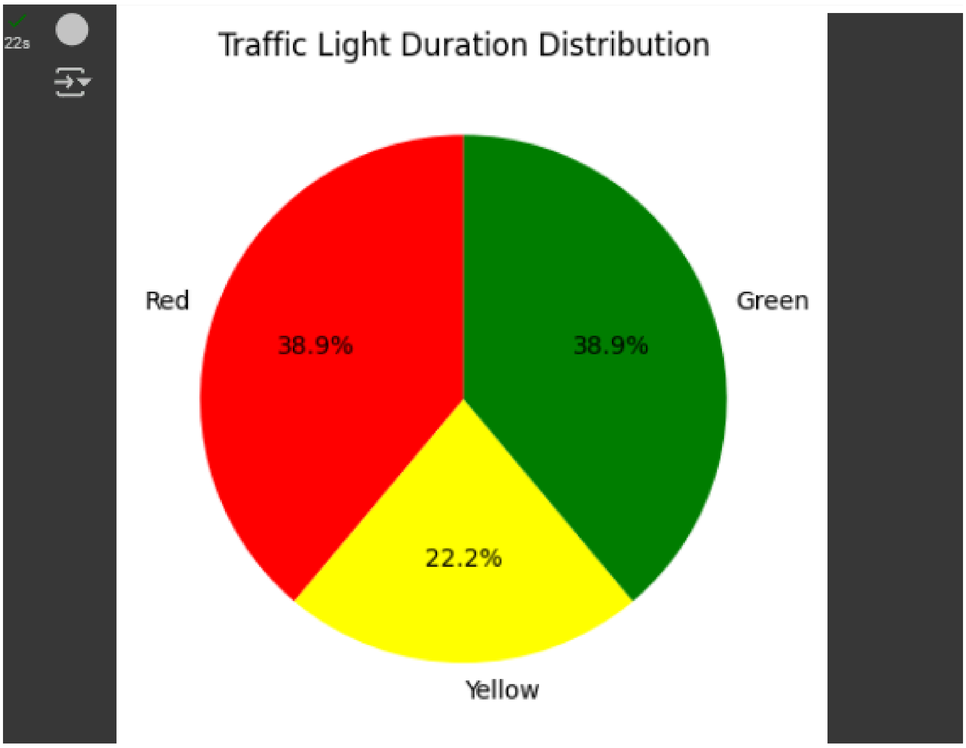
Code:

```
import time
import os
import matplotlib.pyplot as plt
# Function to display the current state of the traffic light
def display_traffic_light(color):
 # Clear the console for better visualization of the traffic light changes
 os.system('cls' if os.name == 'nt' else 'clear')

 # Define the symbols for each light color
 colors = {"Red": "🔴", "Yellow": "🟡", "Green": "🟢"}

 # Define how each light should look based on the current active color
 light_states = {
 "Red": ["🔴", "⚫", "⚫"],
 "Yellow": ["⚫", "🟡", "⚫"],
 "Green": ["⚫", "⚫", "🟢"]
 }

 # Display header
 print("\n" + "=" * 30)
 print(" 🚦 TRAFFIC LIGHT SYSTEM 🚦")
 print("=" * 30 + "\n")

 # List of lights in the order they should display
 lights = ["Red", "Yellow", "Green"]

 # Display each light's state
 for i, light in enumerate(lights):
 state = light_states.get(color, ["⚫", "⚫", "⚫"])
 print(f" {state[i]} {light}")
 print("\n" + "=" * 30)

# Function to plot a bar graph showing light durations
def plot_traffic_light_durations(lights):
 colors = ["red", "yellow", "green"]
 labels, durations = zip(*lights)

 plt.bar(labels, durations, color=colors)
 plt.title("Traffic Light Durations")
 plt.xlabel("Light Color")
 plt.ylabel("Duration (seconds)")
 plt.show()

# Function to plot a pie chart showing light durations
def plot_traffic_light_pie_chart(lights):
 colors = ["red", "yellow", "green"]
 labels, durations = zip(*lights)

 plt.pie(durations, labels=labels, colors=colors, autopct='%1.1f%%', startangle=90)
 plt.title("Traffic Light Duration Distribution")
 plt.show()

# Function to plot a line graph showing light durations
def plot_traffic_light_line_graph(lights):
 colors = ["red", "yellow", "green"]
 labels, durations = zip(*lights)

 plt.plot(labels, durations, marker='o', color='blue')
 plt.title("Traffic Light Durations")
 plt.xlabel("Light Color")
 plt.ylabel("Duration (seconds)")
 plt.grid(True)
 plt.show()

# Function to control the traffic light system
def traffic_light_system():
 # Define each light's color and its duration in seconds
 lights = [
 ("Red", 5), # Red light for 5 seconds
 ("Yellow", 2), # Yellow light for 2 seconds
 ("Green", 5) # Green light for 5 seconds
 ]
 # Show various graphs before starting the system
 plot_traffic_light_durations(lights)
 plot_traffic_light_pie_chart(lights)
 plot_traffic_light_line_graph(lights)

 running = True # Flag to keep the system running

 # Main loop to cycle through lights
 while running:
 for color, duration in lights:
 display_traffic_light(color) # Show the current light
 time.sleep(duration) # Wait for the duration of the light

 # Ask the user if they want to stop or continue after each full cycle
 user_input = input("\nType 'stop' to end the system or press Enter to continue:
").strip().lower()
 if user_input == 'stop':
 running = False # Exit the loop if user types 'stop'
# Run the traffic light system and handle keyboard interruptions gracefully
try:
 traffic_light_system()
except KeyboardInterrupt:
 print("\nTraffic light system stopped.")
```

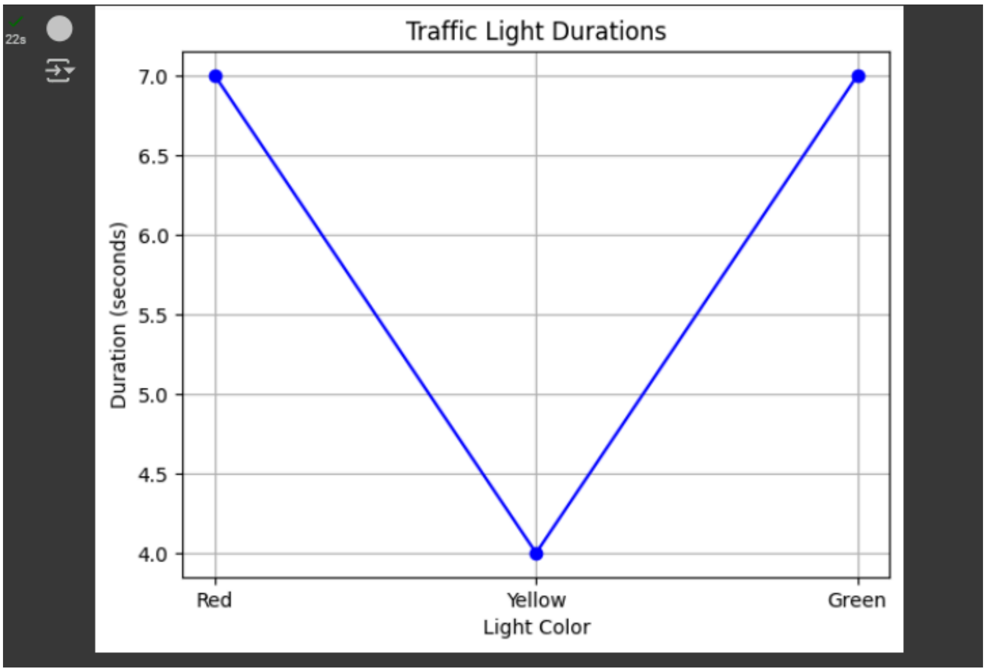
Output:



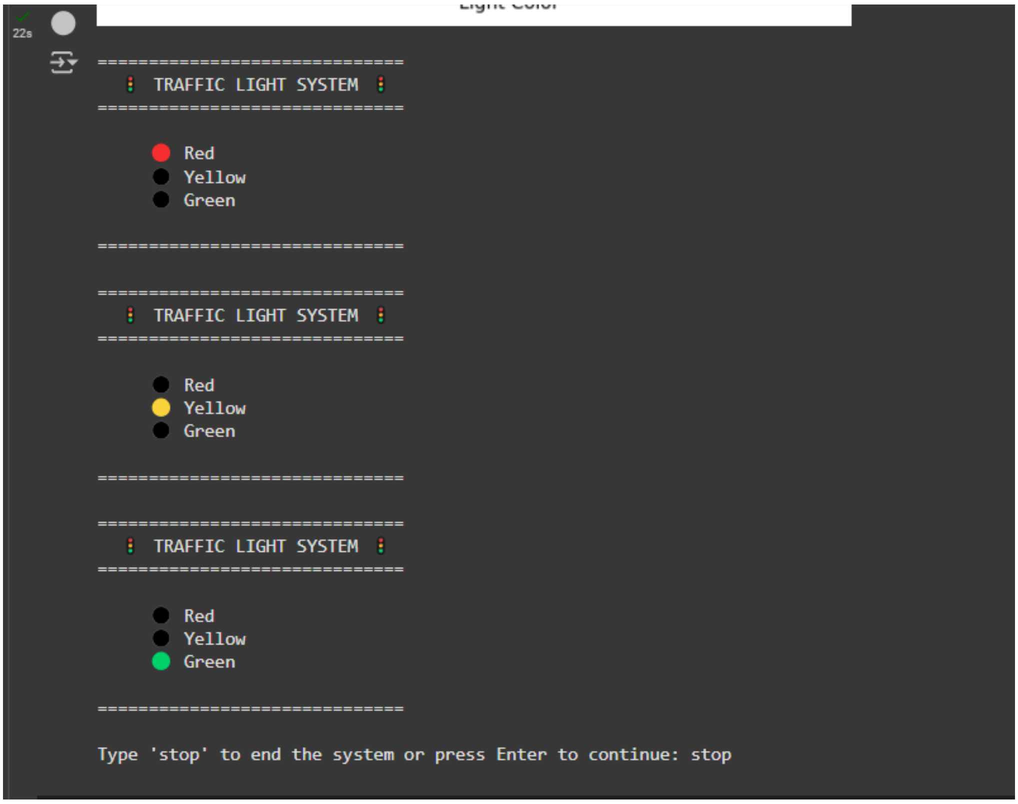
This Graph represents duration of each light in Bar Graph



This Graph represents duration of each light in Pie Graph



This Graph represents duration of each light in Line Graph



```
==============================
  🚦 TRAFFIC LIGHT SYSTEM 🚦
==============================

     🔴 Red
     ⚫ Yellow
     ⚫ Green

==============================

==============================
  🚦 TRAFFIC LIGHT SYSTEM 🚦
==============================

     ⚫ Red
     🟡 Yellow
     ⚫ Green

==============================

==============================
  🚦 TRAFFIC LIGHT SYSTEM 🚦
==============================

     ⚫ Red
     ⚫ Yellow
     🟢 Green

==============================

Type 'stop' to end the system or press Enter to continue: stop
```

This is the output run by the code on GOOGLE COLAB