

```

import random
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation

# Configuration
NUM_PHASES = 4
CYCLE_TIME = 120
HEADWAY_TIME = 2
labels = ['North', 'East', 'South', 'West']
vehicle_types = ['Car', 'Truck', 'Bike', 'Auto']

vehicle_data_summary = {label: {v: {'count': 0, 'total_speed': 0} for v in vehicle_types} for label in labels}
vehicle_log = [] # Records: [vehicle type, speed, direction, time, count @ time]

def vehicle_count(green_time):
    return int(green_time // HEADWAY_TIME)

def generate_vehicle_types(count):
    return random.choices(vehicle_types, k=count)

def generate_vehicle_speeds(count):
    return [random.uniform(20, 60) for _ in range(count)]

def update(frame):
    global vehicle_log
    ax1.clear(); ax2.clear(); ax3.clear(); ax4.clear(); ax5.clear()

    green_times = [random.randint(20, 40) for _ in range(NUM_PHASES)]
    vehicle_counts = [vehicle_count(g) for g in green_times]

    for i in range(NUM_PHASES):
        direction = labels[i]
        count = vehicle_counts[i]
        vtypes = generate_vehicle_types(count)
        speeds = generate_vehicle_speeds(count)

        for j in range(count):
            vt = vtypes[j]
            speed = speeds[j]
            vehicle_data_summary[direction][vt]['count'] += 1
            vehicle_data_summary[direction][vt]['total_speed'] += speed
            vehicle_log.append([vt, f"{speed:.1f}", direction, frame, count])

    ax1.bar(labels, green_times, color='green')
    ax1.set_title("Green Light Durations")
    ax1.set_ylim(0, CYCLE_TIME)

    for idx, vt in enumerate(vehicle_types):
        speeds = []
        for dir in labels:
            data = vehicle_data_summary[dir][vt]
            avg = data['total_speed'] / data['count'] if data['count'] > 0 else 0
            speeds.append(avg)
        ax2.bar(np.arange(NUM_PHASES) + idx*0.2, speeds, width=0.2, label=vt)
    ax2.set_title("Avg Speeds per Vehicle Type")
    ax2.set_xticks(np.arange(NUM_PHASES) + 0.3)
    ax2.set_xticklabels(labels)
    ax2.legend()

    for idx, vt in enumerate(vehicle_types):
        counts = [vehicle_data_summary[dir][vt]['count'] for dir in labels]
        ax3.bar(np.arange(NUM_PHASES) + idx*0.2, counts, width=0.2, label=vt)
    ax3.set_title("Vehicle Type Counts")
    ax3.set_xticks(np.arange(NUM_PHASES) + 0.3)
    ax3.set_xticklabels(labels)
    ax3.legend()

```

```

for idx, vt in enumerate(vehicle_types):
    counts = [vehicle_data_summary[dir][vt]['count'] for dir in labels]
    ax3.bar(np.arange(NUM_PHASES) + idx*0.2, counts, width=0.2, label=vt)
ax3.set_title("Vehicle Type Counts")
ax3.set_xticks(np.arange(NUM_PHASES) + 0.3)
ax3.set_xticklabels(labels)
ax3.legend()

ax4.bar(labels, vehicle_counts, color='orange')
ax4.set_title("Vehicles This Frame")
ax4.set_ylim(0, max(vehicle_counts) + 5)
ax4.text(0.5, max(vehicle_counts) - 1, f"Time: {frame}s", fontsize=12)

ax5.axis('off')
table_data = vehicle_log[-10:]
col_labels = ["Type", "Speed", "Direction", "Time", "Count@Time"]
table = ax5.table(cellText=table_data, colLabels=col_labels, loc='center')
table.auto_set_font_size(False)
table.set_fontsize(8)
table.scale(1.2, 1.5)

fig, (ax1, ax2, ax3, ax4, ax5) = plt.subplots(1, 5, figsize=(25, 6))
ani = FuncAnimation(fig, update, frames=range(0, 100), interval=1000)
plt.tight_layout()
plt.show()

```