Workshop - 3

Workshop Value: 10 marks (4.4% of your final grade)

Learning Outcomes

Upon successful completion of this workshop, you will have demonstrated the abilities:

- to decipher and identify a problem
- to analyze and decompose a problem
- to identify the required detailed steps to solve a problem
- to communicate the solution to fellow peers and non-technical business persons

Workshop Grading and Promotion Policy

Workshops for this course will be assessed using the following criteria:

- Workshops must be completed before the class time to be graded
- You must successfully complete 9 workshops (if more than 9 are completed, the best 9 will be used)
- Each student is expected to be a presenter of the workshop solution at least once by the end of the term
- Workshop solutions and presentations will be evaluated using the published workshop rubrics

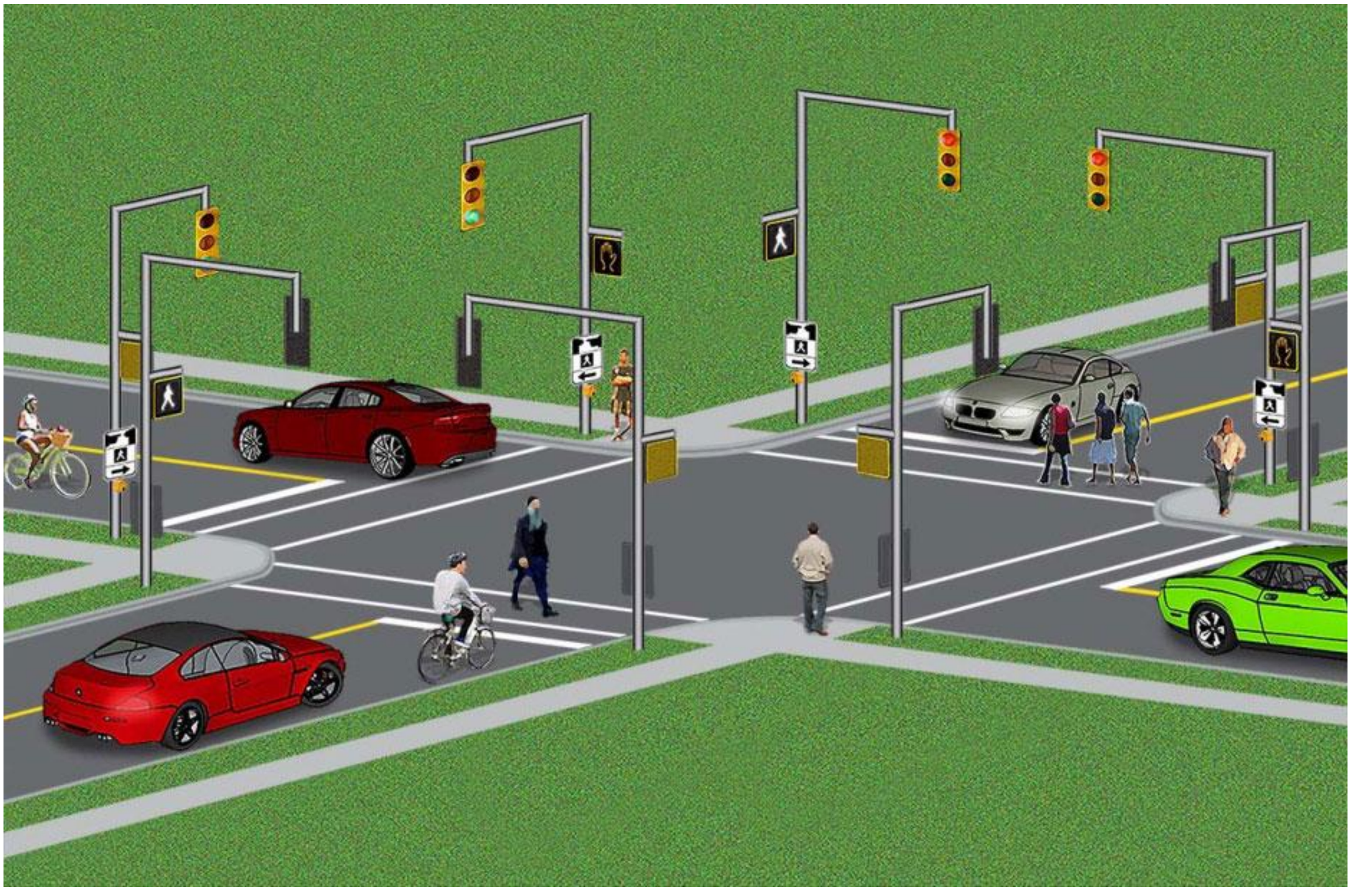
Workshop Overview

Finite state machines are around us everywhere. State machines put simply, are logical units that are limited to being in only one state at any given time. A finite state machine is one that is limited to a defined set of states (there are defined limits to the number of possible states it can be in).

Traffic lights are a good example of finite state machines. In a typical traffic light, there are 3 possible states: Green (go), Amber (don't begin to cross the intersection because the red light is about to trigger), and Red (stop). Each state is strictly timed (can be different in duration for each state) and synchronized with the opposing set of lights.

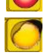
Workshop Details

A busy intersection in downtown Toronto needs to upgrade the current 3-light 3-state traffic light system to a **4-light 5-state** system to address an increasingly heavy demand for left turns. The intersection with the 3 light system looks like this:



Looking at the light that is red, we see that the cars are stopped, waiting for the light to turn green. If 6 of more cars line up, the car sensor will signal the light to turn green sooner, as described below. The pedestrian who wants to go in the same direction as the cars stopped on the red light is also stopped. If the pedestrian presses the walk button on the light pole, this will also decrease the wait time for the red light as described below.

The 4 lights in the new system are as follows:

1. Red 
2. Amber 
3. Green 
4. Advance left turn  (can be Green or Amber)

4-Light 5-State Traffic Light: ***Sequence of states***

The sequence of a 4-light **5-state** system is shown in the above illustration. Below are the 5-states including the time duration for each state:

1. Solid Green + Advanced Left **Green** Arrow **[15 seconds]**
2. Solid Green + Advanced Left **Amber** Arrow **[5 seconds]**
3. Solid Green + Advanced Left Arrow (**off**) **[40 seconds]**
4. Solid **Amber** **[10 seconds]**
5. Solid **Red** **[?? seconds: !!! YOU FIGURE THIS OUT !!!]**

Additional Features

Request-To-Walk Button

A button is provided for pedestrians to press to alert the system a person needs to cross. This is only effective when the lights are in a red-light state. This request will reduce the remaining wait time of the red-light state to 25 seconds.

Car Sensor

There is a sensor mounted on the light fixtures that monitors the number of waiting cars when in a red-light state. When the number of cars waiting reaches 6 or more, the sensor will trigger the system to reduce the remaining wait time of the red-light state to 25 seconds.

Interrupt

The defined system should take into account an "interrupt". This is something that can occur at any time and will take over/supersede the active state. Examples of this type of interruption can be the traffic controllers setting all lights to flash amber (for caution) or to flash red (4-way stop) etc.. **You don't need to specify what the interrupt should do – only account for it by checking if there is an interrupt to the normal process flow**.

Your Tasks

1. Create the necessary processes to accomplish the 4-light 5-state traffic light system
   Note: The system should start in a **solid green light + advanced left green arrow state**
2. Define the solution applying what you know about the computational thinking approach to problem solving
3. Communicate the solution using pseudo code followed by a flowchart
4. One student will prepare a video and describe the problem and solution from a high level perspective and argue why this is a good solution to the problem.

[Logic 1] This will define a 3 light (red, green, amber) light system without an advance green, walk light or interrupt.
[Logic 2] This will define a red-green-amber light system with a request to walk button and a car sensor. The system will not have an advance green light.
[Logic 3] This will define a red-green-amber system with an interrupt. The system will not have an advance green light, walk button or car sensor.
[Combined] Will put all this logic together and add the advanced left green and amber arrows.

Task	Subtask	Member(s)	Marks	Comments
Pseudocode	Logic 1	1	40%	
	Logic 2	2	40%	
	Logic 3	3	40%	

	Combined	1-3	60%	
FlowChart	Logic 1	4	40%	
	Logic 2	5	40%	
	Logic 3	6	40%	
	Combined	4-6	60%	
Video	Presentation	3 or 6	100%	Members rotate weekly