

Problem:

Ordinary traffic control works by setting predetermined durations for green lights on each lane, which isn't efficient given the high variation in traffic, day to day especially with weather changes happening, and road emergencies.

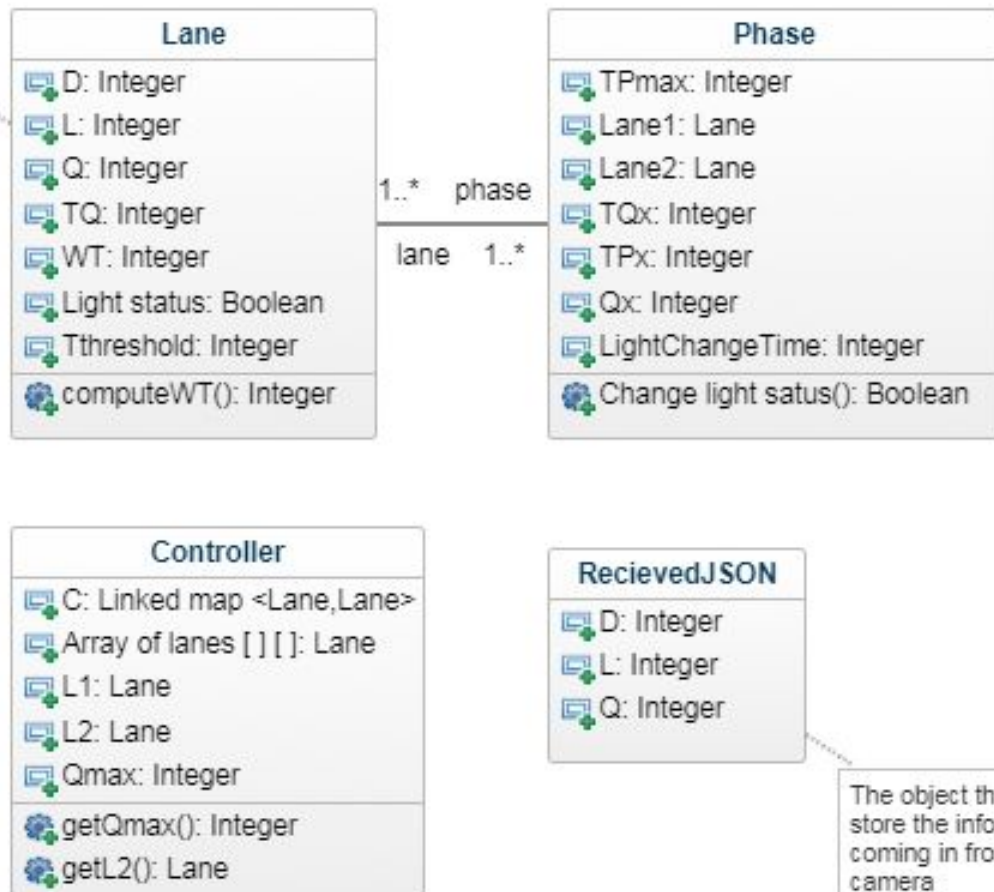
This problem had to be put to an end, and a dynamic, smart, autonomous traffic control system had to be put together.

Problem variables:

Notation	Name	explanation
$D = \{N,S,E,W\}$	Direction	The position of a certain lane w.r.t. the intersection (north,south,east,west)
$L = \{F,R\}$	Lane	Heading of the cars on the lane, whether it's forwards or turning right
$C = \{1,2,3,...,12\}$	phase	A valid instance of green traffic lights combinations
$Q(d,l)$	Lane waiting queue	The amount of cars in a certain lane
$TQ(d,l) =$ (Passing time of the first vehicle) + ((Time taken by the vehicle to get to the next position) * ($Q(d,l) - 1$))	Queue passing time	The time needed for the whole queue in a certain lane to pass the traffic light
$WT(d,l)$	Queue waiting time	Wait time of the 1st vehicle in a certain lane
$Qx = \max(TQ(d,l))$ (where x is a certain phase)	Phase waiting queue	Max time till the whole queue pass in a phase
$TPx = \min(TQx,TPmax)$	Green light time	Time to keep the green light

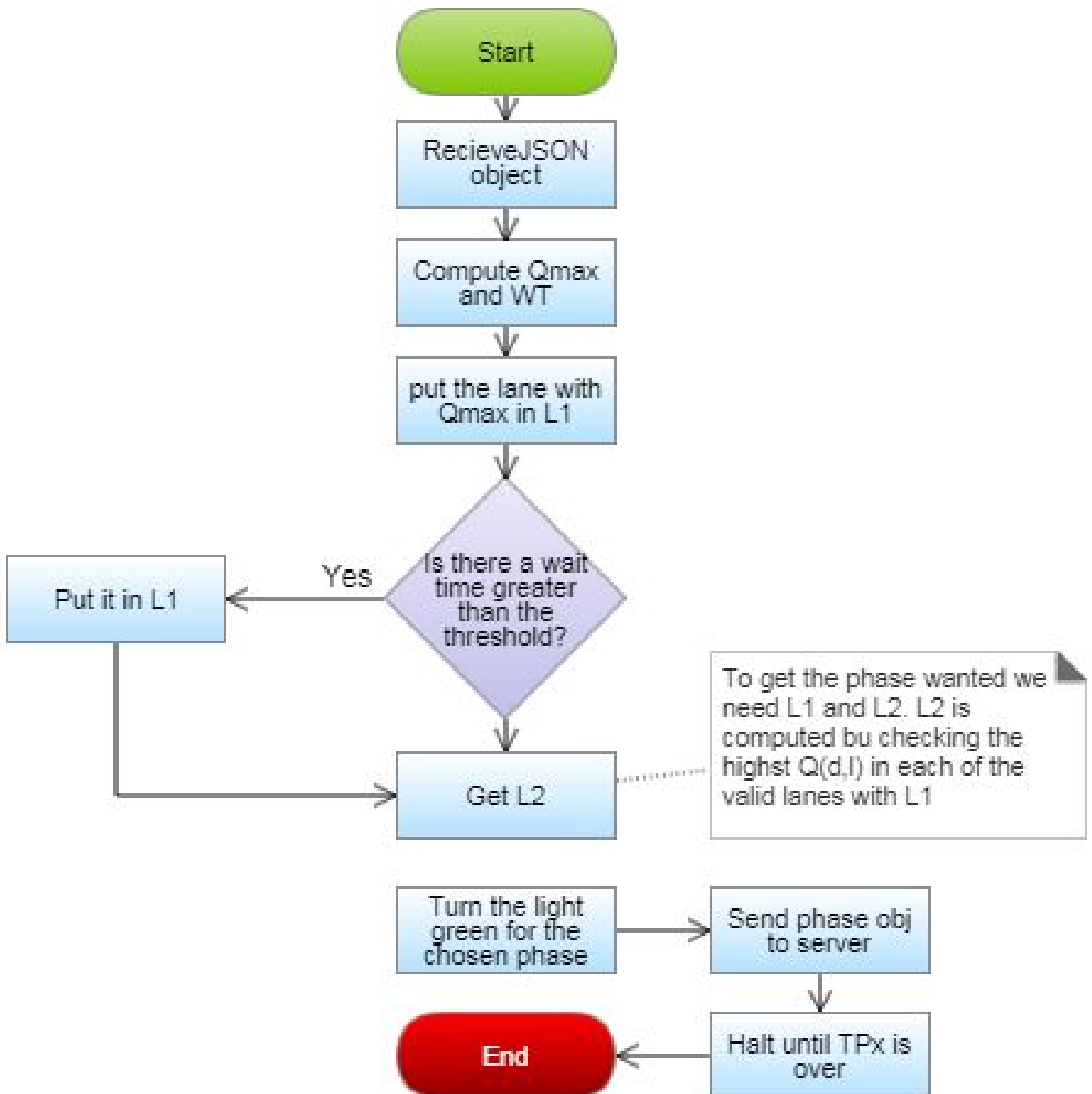
Class diagram (UML 2.0):

D & L are an integers to avoid searching the array in the controller. The integer numbers are as follows:
N=1,S=2,E=3,W=4
F=1,R=2



The object that will store the information coming in from the camera

FlowChart:



Pseudo code:

- $A[\text{recivedJSON.D}][\text{recivedJSON.L}].Q = \text{recivedJSON.Q}$
- $Q_{\max} = \text{controller.getQmax}(A)$ // puts the $A[d][l]$ with Q_{\max} in $L1$
- For all A :
 - $A[d][l].\text{computeWT}()$
 - If $(A[d][l] > T_{\text{threshold}})\{$
 - $A[d][l] = L1$
 - $\}$
- $\text{controller.getL2}(L1)$
- New phase $P (L1, L2)$
- $p.\text{changeLightStatus}$
- Send P to server
- While $(\text{now}() - p.\text{timeOfChange}) < p.TPx)\{\}$

Important functions' flow:

- **Controller.getQmax ($A[D][L]$)**
Int $Q_{\max} = 0$;

For all A :

 If $(A.Q > Q_{\max})\{$
 $Q_{\max} = A[d][l].Q$

$L1 = A[d][l]$

```

}
Return Qmax
}

```

```

• Lane.computeWT(){
    if(this.lightStatus==1){WT=0}
    Else{WT=(now() - timeOfChange)}
}

• Phase.changeLightStatus(){
    if(lightStatus==green){lightStatus=red}
    Else{light status = green}
// repeat for L1 and L2
this.timeOfChange = now()
}

```

Valid phases:

