Simulation of a streets lights system using python

Obryl Donald Houaghon Nankap(523404)
Department of Informatics
Clausthal University of Technology
Clausthal, Germany
obryl.donald.houaghon.nankap@tu-clausthal.de

***Abstract*—With the increasing number of car users, their movement must be regulated to limit accidents and allow efficient use of the road. The road system present multiple form and the crossroads are not the same. The control of theses crossroads must be regulated by light system which cannot be same everywhere and the presence of a Human to control it is not recommended because this is a high demanding task, which can result of accident or an overflow of the traffic. Therefore the regulation of the traffic by a system which will every time present an is adapted to the current crossroad is great of importance. This paper present a solution of a system of light which can be implemented a crossroad which contain a principal lane and a second lane. This system propose an algorithm which will control different lane and ensure the good circulation. This system can be simulated and optimized before the implementation in the real world which can reduce cost of the production of the final system.**

***Key words*—Simulation, Automation, Light System, python**

I. Introduction

One of the best way to create and implement a new or an existent system is to simulate the system in a dedicated software. This method has the advantage to reduce coast and bring up more flexibility and can be modified and reused in other project. The boxes, and the line are illustrated to emphasize the comprehension of the proposed method. Moreover this implementation should be tested in order to verify it operability. In this academic project, a light system should be implemented, simulated and tested to show that the requirements are meet using python.

II. Presentation of the light system

The common approach to implement a system is to visualize it, in order to have a good comprehension and to choose the best way to realize it. The provided road and light system to simulate is illustrated in figure 1. As we can see this road count a second lane and a principal lane which contain also two side. The total light present here are 3. With specific time and event the road are controlled such that the traffic can be controlled. The principal lane should allow every car to pass until a car is present in the second lane, which is detected by sensor or a equivalent device. This presence enable an event to stop the car on the principal lane and allow the car in the second lane to pass. After this event the principal lane return to the initial condition until another are present in the second lane. A UML algorithm which illustrate different events is present in figure 2.

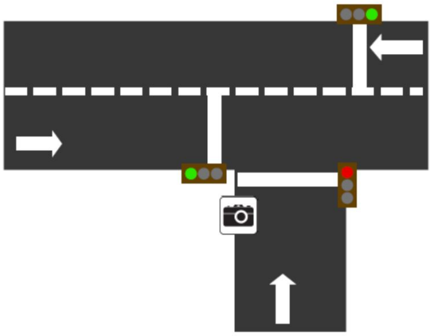


Fig. 1. Illustration of the light system

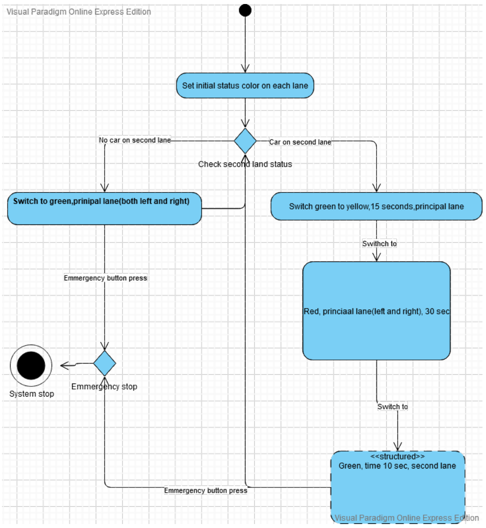
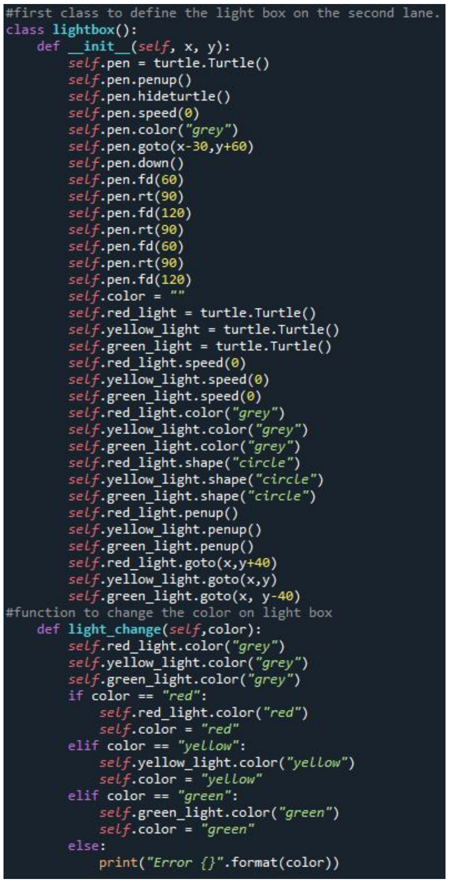

Visual Paradigm Online Express Edition
Set initial status color on each lane
No car on second lane
Car on second lane
Check second land status
Switch to green,prinipal lane(both left and right)
Switch green to yellow,15 seconds,principal lane
Swithch to
Emmergency button press
Red, princiaal lane(left and right), 30 sec
System stop
Emmergency stop
Switch to
<<structured>>
Green, time 10 sec, second lane
Emmergency button press
Visual Paradigm Online Express Edition

Fig. 2. Illustration of every state of the program

III. Implementation

The light system program was implemented using the python programming on Spyder(Python 3.8). Base on the description related in part II and the illustration in ”Fig

1” the different part to control light on different lane was implemented. The following figure represent the approach to realize the top down visual of the system using the a python library named turtle [2].



```
#first class to define the light box on the second lane.
class lightbox():
    def __init__(self, x, y):
        self.pen = turtle.Turtle()
        self.pen.penup()
        self.pen.hideturtle()
        self.pen.speed(0)
        self.pen.color("grey")
        self.pen.goto(x-30,y+60)
        self.pen.down()
        self.pen.fd(60)
        self.pen.rt(90)
        self.pen.fd(120)
        self.pen.rt(90)
        self.pen.fd(60)
        self.pen.rt(90)
        self.pen.fd(120)
        self.color = ""
        self.red_light = turtle.Turtle()
        self.yellow_light = turtle.Turtle()
        self.green_light = turtle.Turtle()
        self.red_light.speed(0)
        self.yellow_light.speed(0)
        self.green_light.speed(0)
        self.red_light.color("grey")
        self.yellow_light.color("grey")
        self.green_light.color("grey")
        self.red_light.shape("circle")
        self.yellow_light.shape("circle")
        self.green_light.shape("circle")
        self.red_light.penup()
        self.yellow_light.penup()
        self.green_light.penup()
        self.red_light.goto(x,y+40)
        self.yellow_light.goto(x,y)
        self.green_light.goto(x, y-40)
#function to change the color on light box
    def light_change(self,color):
        self.red_light.color("grey")
        self.yellow_light.color("grey")
        self.green_light.color("grey")
        if color == "red":
            self.red_light.color("red")
            self.color = "red"
        elif color == "yellow":
            self.yellow_light.color("yellow")
            self.color = "yellow"
        elif color == "green":
            self.green_light.color("green")
            self.color = "green"
        else:
            print("Error {}".format(color))
```

Fig. 3. Code for illustration of the traffic light

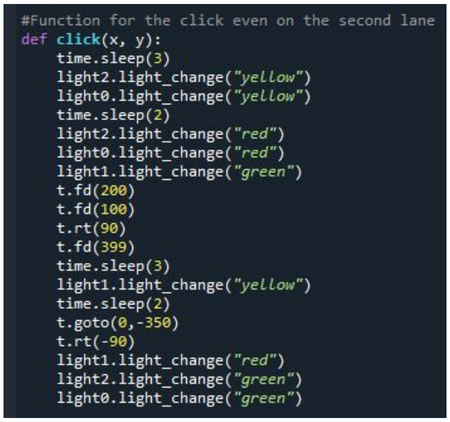
A. Event implementation

Another function is used to create an event which stops the car on the principal lane and allow the car on the second lane to pass when the light are red on the principal lane and green on the principal lane.

IV. Testing the Program

To test our program we assume that the program should not allow the green light on the second lane and on the principal lane at the same time. Moreover, the system should be able to permit a car on the second lane to pass while the car on the two principal lane are stopped. Our implementation meet those requirement, the time used in the program are evaluated in second. Those time are set in seconds are intentionally set in second to see an execution of the system. There are few step when we execute the program:

- The principal lane lights are set to green until a car is present on the second lane.
- The lights are set to green 3 seconds to yellow with 2 second sleep, and then to the red and stop for 1 seconds.



```
#Function for the click even on the second lane
def click(x, y):
    time.sleep(3)
    light2.light_change("yellow")
    light0.light_change("yellow")
    time.sleep(2)
    light2.light_change("red")
    light0.light_change("red")
    light1.light_change("green")
    t.fd(200)
    t.fd(100)
    t.rt(90)
    t.fd(399)
    time.sleep(3)
    light1.light_change("yellow")
    time.sleep(2)
    t.goto(0,-350)
    t.rt(-90)
    light1.light_change("red")
    light2.light_change("green")
    light0.light_change("green")
```

Fig. 4. Code for click event on the traffic light

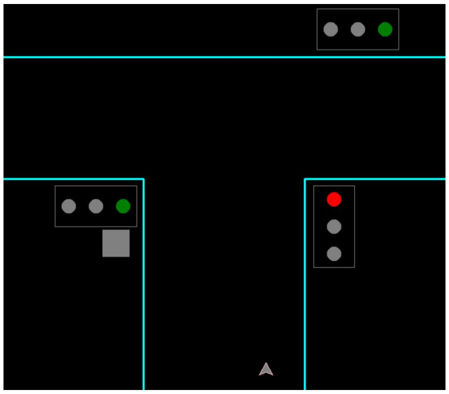


Fig. 5. Illustration of the traffic light with python

- When the the car return on the second lane the system wait until the button is triggered.

The time here are set in second to have a very responsive system to see how it run.

V. Discusion

The implemented program is capable to excute the instruction with every steps included to maintain the order in the traffic. However the program do include a function to stop the program. The cross button must be clicked to each time to close the program. Also the road as illustrated do not display each details as illustrated in ”Fig. 1”. The future work which can be done on this program are: the addition of real time capabilities, the addition of more details on the traffic. Those will increase the performance and the quality of the program.

VI. Conclusion

The goal of this project was to Implement a light traffic using the Python programming Language and the test if the requirement are meet. The execution of the system prove that the Implemented code contain all the required condition. All of this was achieve at the end of the project however more important optimization and update are yet to be done.

Acknowledgment

This project was developed as part of a student project in ”Simulation Engineering” lecture by apl. Prof. Dr. Umut Durak at the Technical University of Clausthal. I want to thank apl. Prof. Dr. Umut Durak for his oustanding commintment, support help during the online lecture and question regarding the project.

References

[1] Umut Durak , “Simulation Engineering,’ Clauthal University of Tehchnologiy’,2020, Week 3.

[2] Nikita Silaparasetty, The Beginner’s Guide to Python Turtle. https://realpython.com/beginners-guide-python-turtle/, Febuary 2021.