

## **Report writing steps**

According to the map given to each team, develop a controller for each intersection (plant), that controller is a closed-loop one (with the  $in_{(1..n)}$  input channels that is connected to its intersection's output channels  $op_{(1..n)}$  and an Intersections (with the  $OPs$  output channels). The controller must have dynamic delays feature to extend the time of the green light in case of a traffic jam. (Project session 3, and 5)

### **Specifications:**

1. Paste a screen shot of the entire given map,
2. Draw a simplified one showing the intersections and the middle street that connects them, if the street has output and input lanes, they should be drawn and implemented at the end.

### **Design:**

1. Draw the OETPN model for the Plant (the intersections and the middle street).
2. Write the Place types, grd&map for the entire OETPN.
3. Draw the OETPN model for the controllers.
4. Write the Place types, grd&map for the entire OETPN.
5. Draw the component diagram for the entire system (depending on your implementation, each OETPN is considered a component) and show the names of the input and output channels.

### **Implementation:**

1. Add the repository link here. The repository must have all the team members as collaborators and the students should work and commit code together.

### **Testing:**

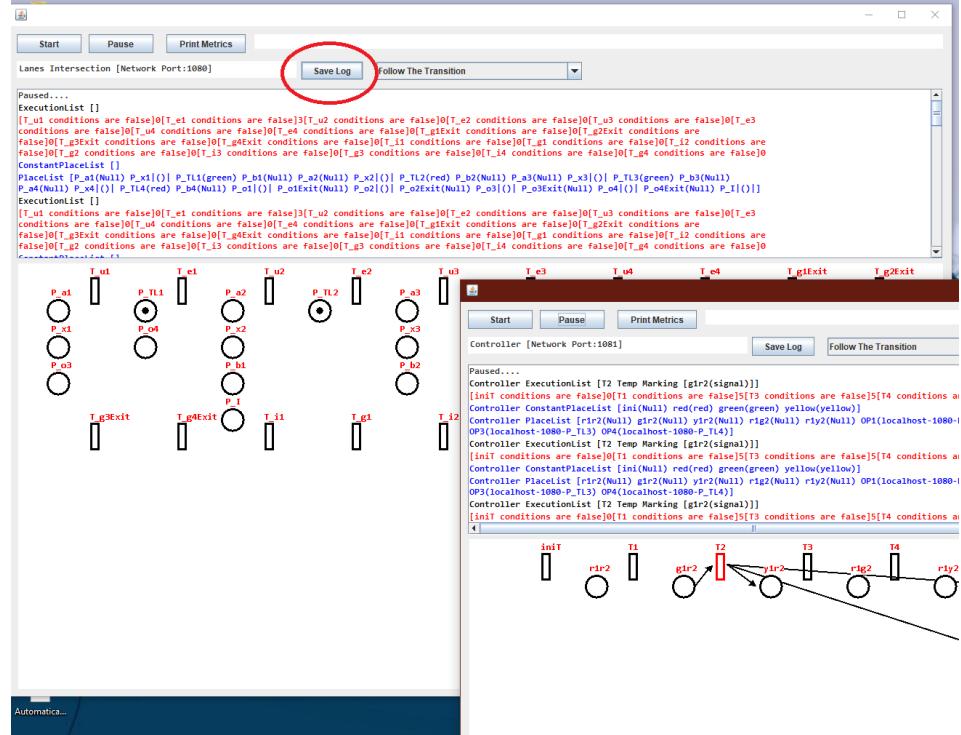
1. Send a car from the 1<sup>st</sup> intersection, that should go through the middle street and exit from one of the exit lanes from the 2<sup>nd</sup> intersection. Attach screen shots showing how the car moves and at the end of the test, pause the intersection OETPN and click on the save log button, save it as test1\_intersection 1.txt and test1\_intersection 2.txt if you have implemented them in two separate OETPNs. Then add the text file/s to the repository.
2. Traffic jam: for each intersection, create a traffic jam case by sending the maximum number of cars to the input lane of the intersection, start the controller, then send the last car. The controller should receive a signal from the plant (intersection) and the transition that is responsible for sending a yellow light to that lane where you input the cars to, should have changed the delay to 10 sec. Let the controller OETPN run until it reaches the same transition (2 loops) to show that the delay is changed back to 5 sec. pause the controller OETPN and click on the save log button, save it as test2.txt and add the text file to the repository.

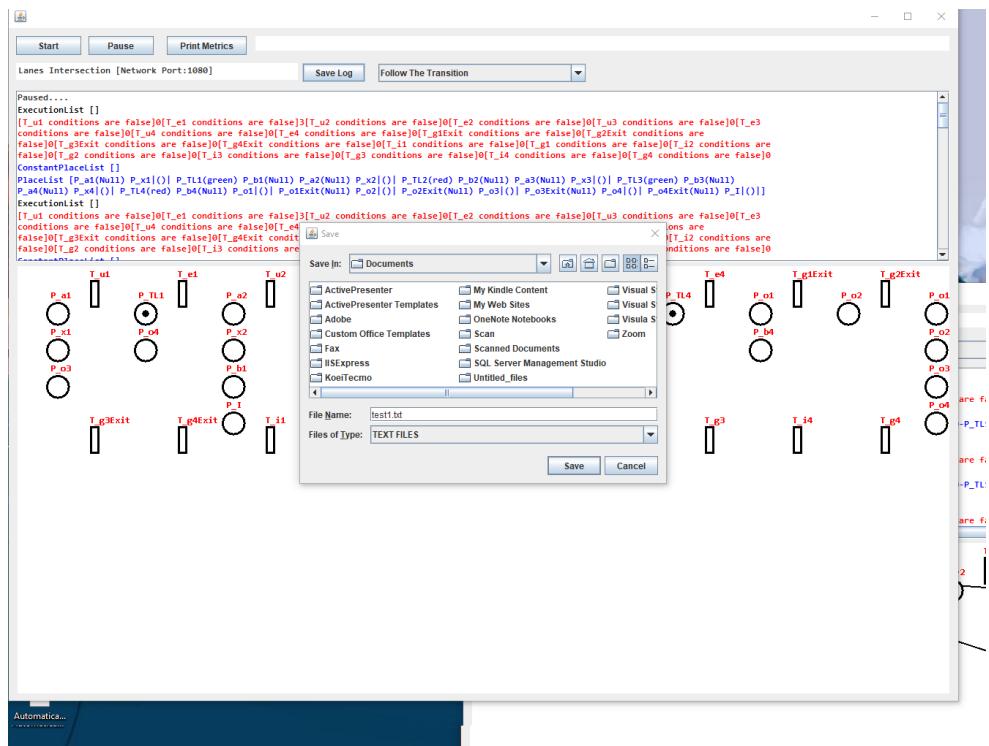
### **Notes:**

- The OETPNs must not be drawn by hand, you have to use a PN drawing tool, you can use any tool you are familiar with. I have uploaded petripipe.
- The component diagram must not be drawn by hand, you have to use a UML drawing tool, you can use any tool you are familiar with. I have uploaded starUML.

- If the middle street has traffic lights, for simplicity purposes do not implement them.
- Do not take into account the bus or the bike lanes.

### Save a log tutorial:





```
test1.txt - Notepad
File Edit Format View Help
Paused.....
ExecutionList []
[T_u1 conditions are false]0[T_e1 conditions are false]3[T_u2 conditions are false]0[T_e2 conditions are false]0[T_u3 conditions are false]0[T_e3 condi
ConstantPlaceList []
PlaceList [P_a1(NULL) P_x1|) P_TL1(green) P_b1(NULL) P_a2(NULL) P_x2|() P_TL2(red) P_b2(NULL) P_a3(NULL) P_x3|() P_TL3(green) P_b3(NULL)
ExecutionList []
[T_u1 conditions are false]0[T_e1 conditions are false]3[T_u2 conditions are false]0[T_e2 conditions are false]0[T_u3 conditions are false]0[T_e3 condi
ConstantPlaceList []
PlaceList [P_a1(NULL) P_x1|) P_TL1(green) P_b1(NULL) P_a2(NULL) P_x2|() P_TL2(red) P_b2(NULL) P_a3(NULL) P_x3|() P_TL3(green) P_b3(NULL)
ExecutionList []
[T_u1 conditions are false]0[T_e1 conditions are false]3[T_u2 conditions are false]0[T_e2 conditions are false]0[T_u3 conditions are false]0[T_e3 condi
ConstantPlaceList []
PlaceList [P_a1(NULL) P_x1|) P_TL1(green) P_b1(NULL) P_a2(NULL) P_x2|() P_TL2(red) P_b2(NULL) P_a3(NULL) P_x3|() P_TL3(green) P_b3(NULL)
ExecutionList []
[T_u1 conditions are false]0[T_e1 conditions are false]3[T_u2 conditions are false]0[T_e2 conditions are false]0[T_u3 conditions are false]0[T_e3 condi
ConstantPlaceList []
PlaceList [P_a1(NULL) P_x1|) P_TL1(green) P_b1(NULL) P_a2(NULL) P_x2|() P_TL2(red) P_b2(NULL) P_a3(NULL) P_x3|() P_TL3(green) P_b3(NULL)
$$$$$$$$$$$$$$$$ I got an Input From Network for P_TL3
$$$$$$$$$$$$$$$$ I got an Input From Network for P_TL1
ExecutionList []
[T_u1 conditions are false]0[T_e1 conditions are false]3[T_u2 conditions are false]0[T_e2 conditions are false]0[T_u3 conditions are false]0[T_e3 condi
ConstantPlaceList []
PlaceList [P_a1(NULL) P_x1|) P_TL1(red) P_b1(NULL) P_a2(NULL) P_x2|() P_TL2(red) P_b2(NULL) P_a3(NULL) P_x3|() P_TL3(red) P_b3(NULL)
ExecutionList []
[T_u1 conditions are false]0[T_e1 conditions are false]3[T_u2 conditions are false]0[T_e2 conditions are false]0[T_u3 conditions are false]0[T_e3 condi
ConstantPlaceList []
PlaceList [P_a1(NULL) P_x1|) P_TL1(red) P_b1(NULL) P_a2(NULL) P_x2|() P_TL2(red) P_b2(NULL) P_a3(NULL) P_x3|() P_TL3(red) P_b3(NULL)
ExecutionList []
[T_u1 conditions are false]0[T_e1 conditions are false]3[T_u2 conditions are false]0[T_e2 conditions are false]0[T_u3 conditions are false]0[T_e3 condi
ConstantPlaceList []
PlaceList [P_a1(NULL) P_x1|) P_TL1(red) P_b1(NULL) P_a2(NULL) P_x2|() P_TL2(red) P_b2(NULL) P_a3(NULL) P_x3|() P_TL3(red) P_b3(NULL)
$$$$$$$$$$$$$$$$ I got an Input From Network for P_TL4
$$$$$$$$$$$$$$$$ I got an Input From Network for P_TL3
$$$$$$$$$$$$$$$$ I got an Input From Network for P_TL2
$$$$$$$$$$$$$$$$ I got an Input From Network for P_TL1
ExecutionList []
[T_u1 conditions are false]0[T_e1 conditions are false]3[T_u2 conditions are false]0[T_e2 conditions are false]0[T_u3 conditions are false]0[T_e3 condi
ConstantPlaceList []
PlaceList [P_a1(NULL) P_x1|) P_TL1(NULL) P_b1(NULL) P_a2(NULL) P_x2|() P_TL2(NULL) P_b2(NULL) P_a3(NULL) P_x3|() P_TL3(NULL) P_b3(NULL)
Waiting For Commands over this port:1080
`
```