

LPS VISUALISATION

A visualiser system for logic-based framework LPS

Created by [Jinwei Zhang](#)

PRESENTATION OVERVIEW

- Background
- Implementation
- Conclusion

PRESENTATION OVERVIEW

- Background
- Implementation
- Conclusion

LPS

LPS stand for Logic Production Systems. It is a "logic-based framework for programming databases and AI (intelligent agent) applications"

LPS combines both production rules and logic program in a single logic-based framework.

Logic production system framework is in the form of $\langle R, L, D \rangle$

R : stands for a set of reactive Rules. It is in the form of
 $\forall X [antecedent \rightarrow \exists Y consequent]$.

L : stands for logic programming. $L = L_{int} \cup L_{events} \cup L_{aux}$.

D : stands for domain theory $D = D_{pre} \cup D_{post}$.

TRAFFIC BACKGROUND:

One-way,T-junction,Cross-junction,Roundabout,Overtaking

TECHNOLOGY STACK

Logic: LPS, LPS.js, LPS.cli(testing)

Front-end: Bootstrap, JavaScript, PIIX.js,

Back-end: AWS, Express.js MongoDB

IMPLEMENTATION

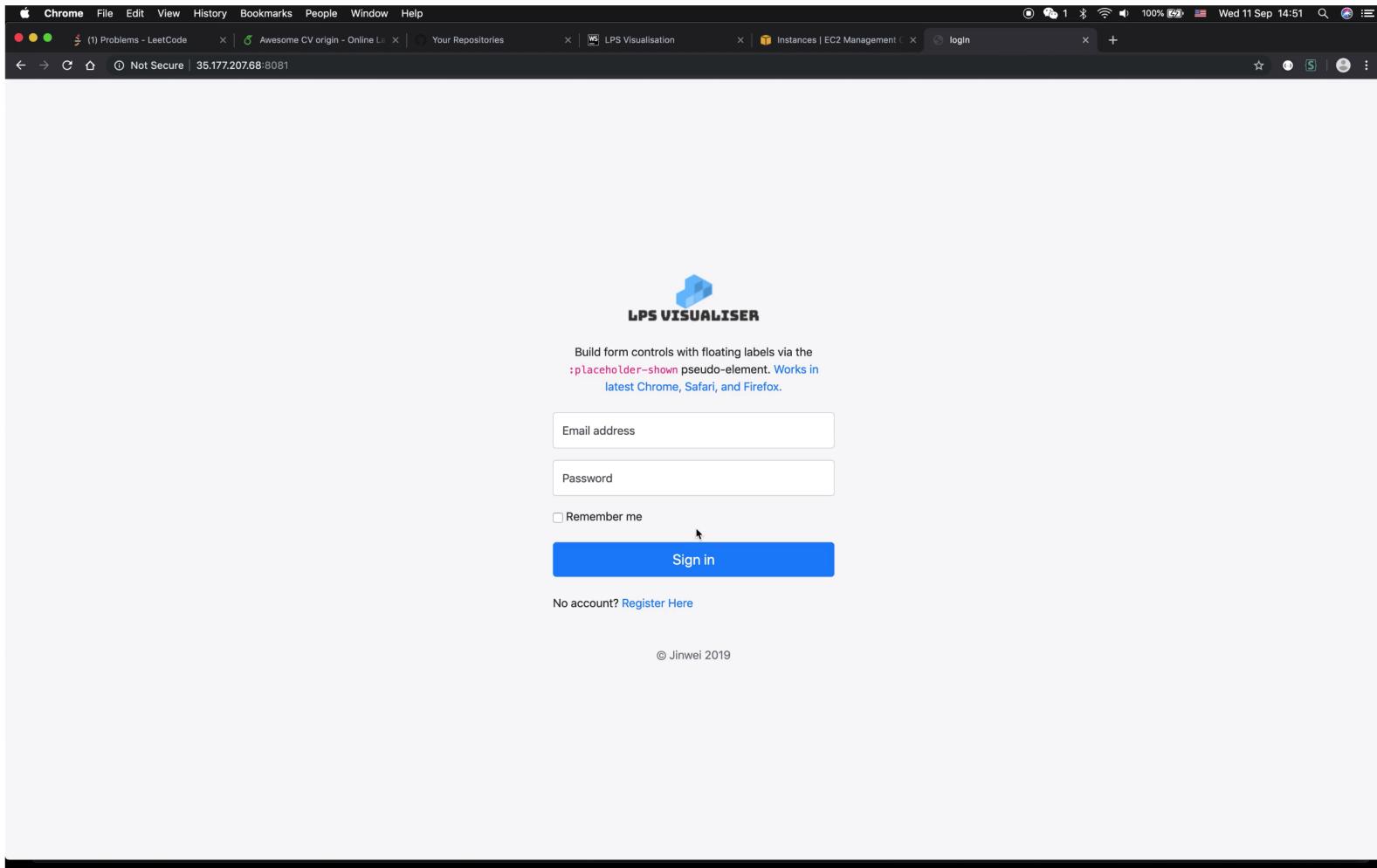
- LPS program with traffic norms
- System front end
- System back end

LPS PROGRAM

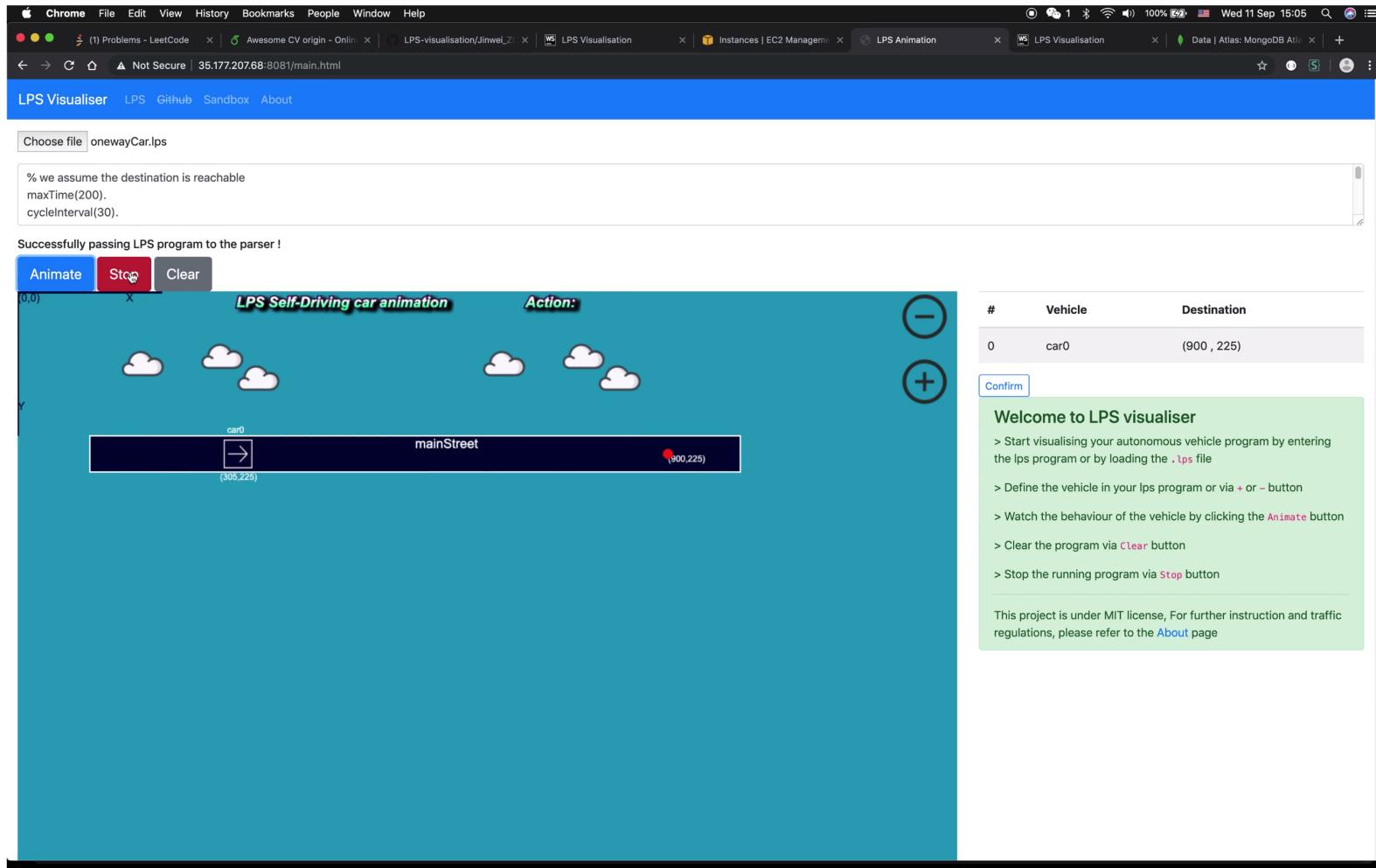
oneway.lps

```
1 % we assume the destination is reachable
2 maxTime(200).
3 cycleInterval(30).
4
5 loadModule('..../scripts/module.js').
6
7 fluents([
8     stopped(VehicleName),
9     velocity(VehicleName, Speed),
10    moving(VehicleName),
11    coordinate(X, Y),
12    location(VehicleName, coordinate(X, Y), Direction),
13    trafficLight(coordinate(X, Y), Working_status, Color, FacingDi
14    street(StreetName, coordinate(X, Y), Width, Height, Number_of_
15    goal(VehicleName,coordinate(X, Y))
16 ]).
17
```

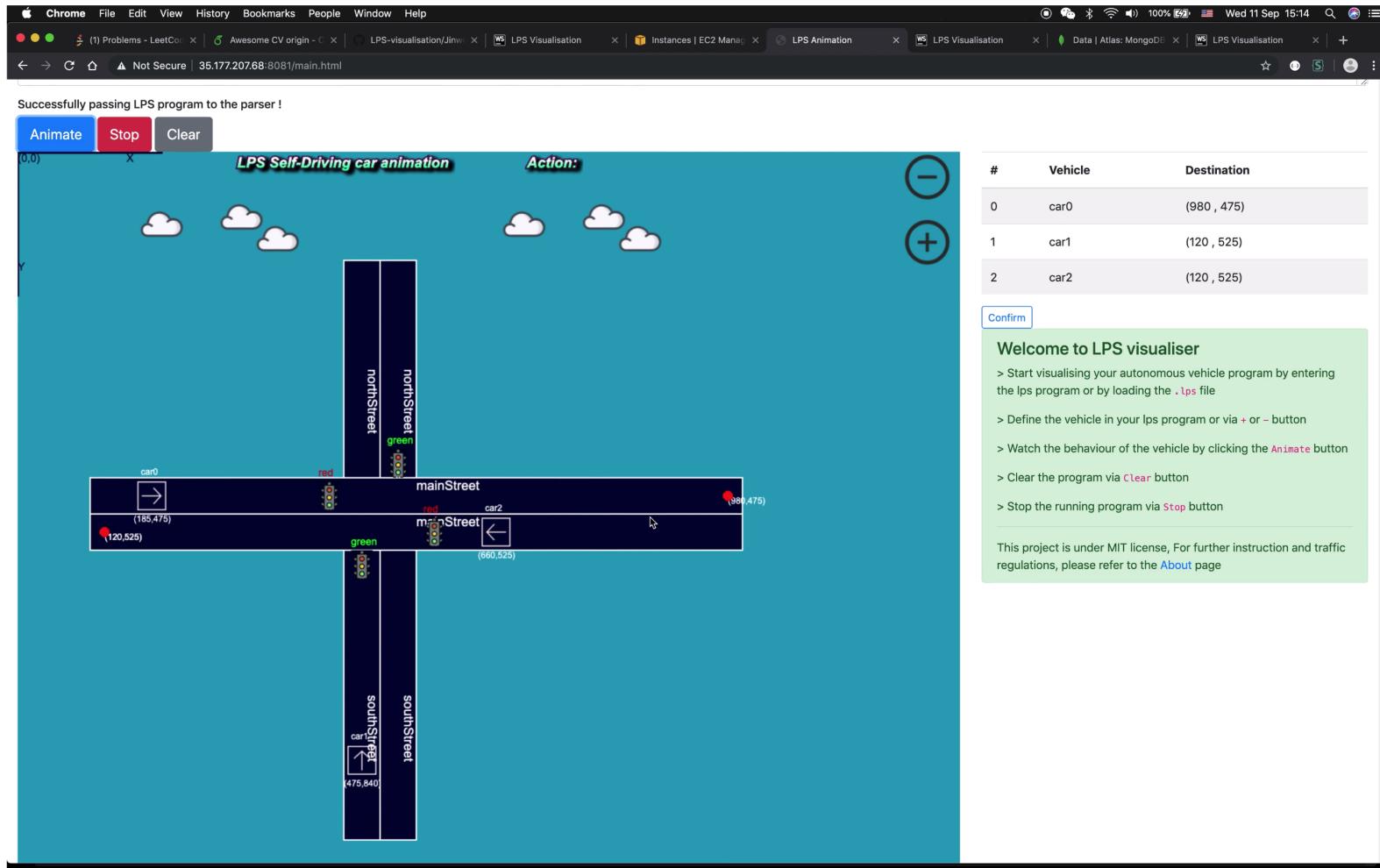
LOGIN/REGISTER



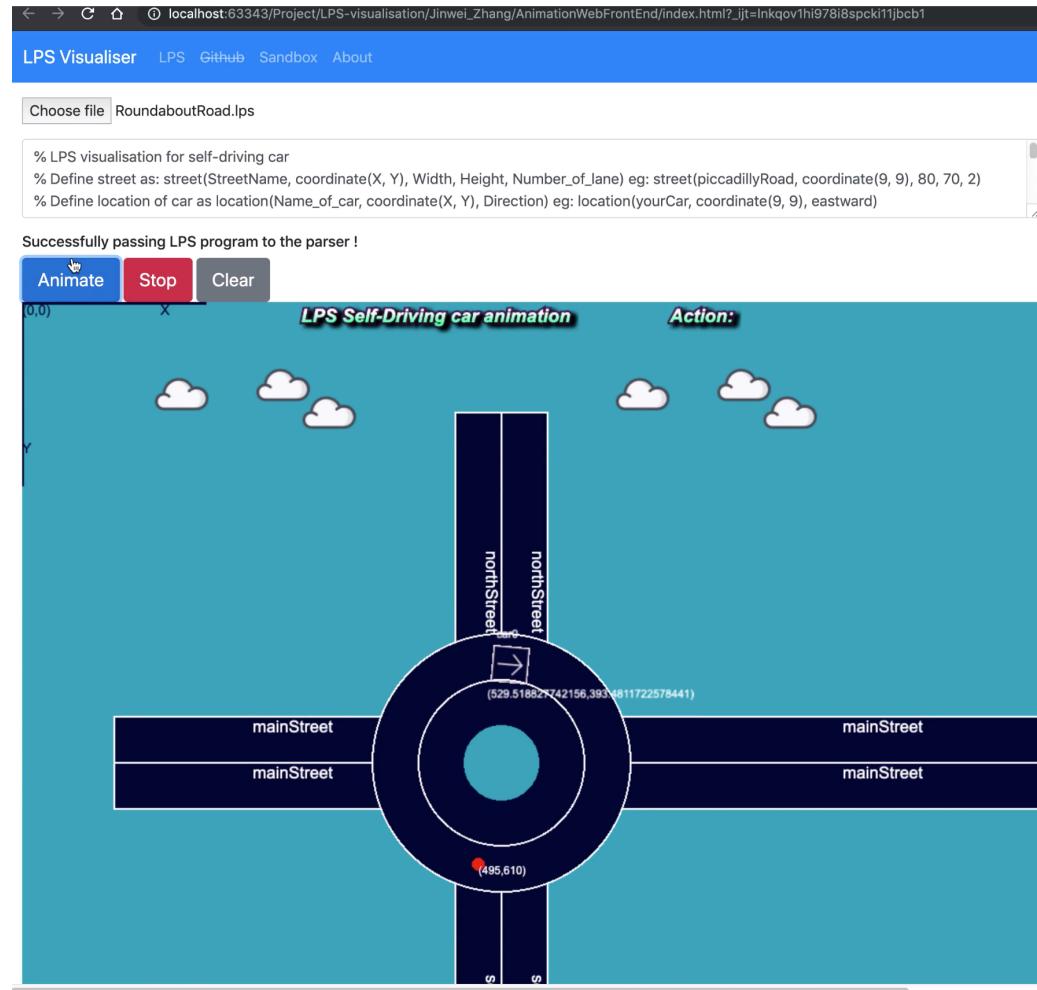
ONE LANE TRAFFIC ROAD



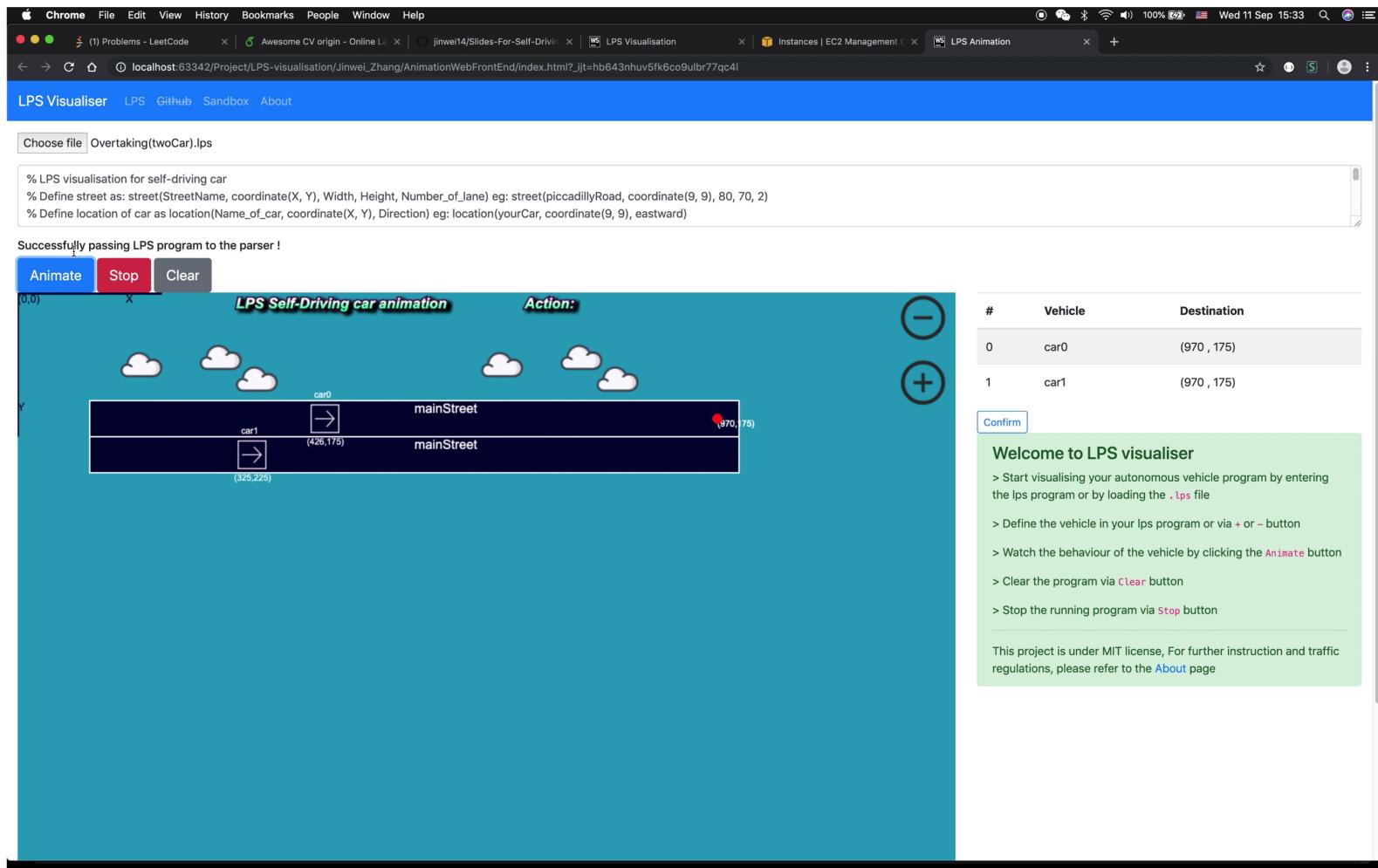
CROSS JUNCTION ROAD



ROUNABOUT



OVERTAKING



CONCLUSION

- Evaluation
- Challenge Faced
- Future Work

THE END

- Try the visualiser online
- Source code & documentation

