



**Tecnológico
de Monterrey**

Actividad Integradora

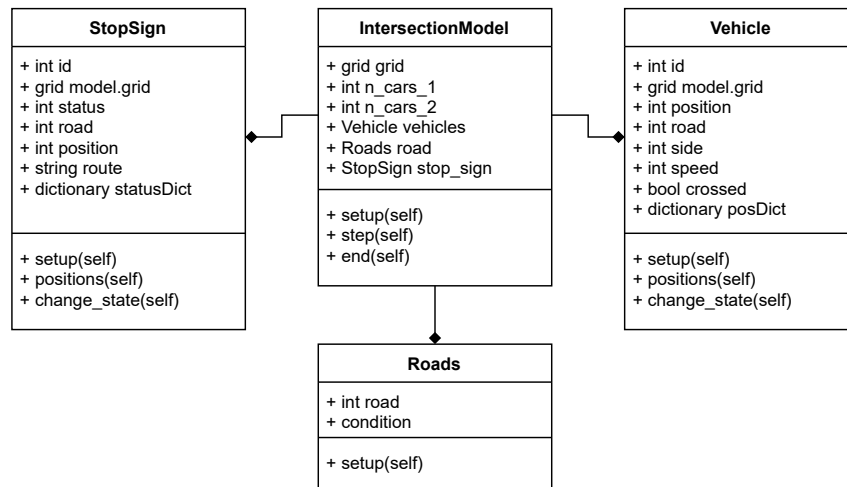
Diego Mojarro Tapia A01638460

Modelación de Sistemas Multiagentes con Graficas Computacionales

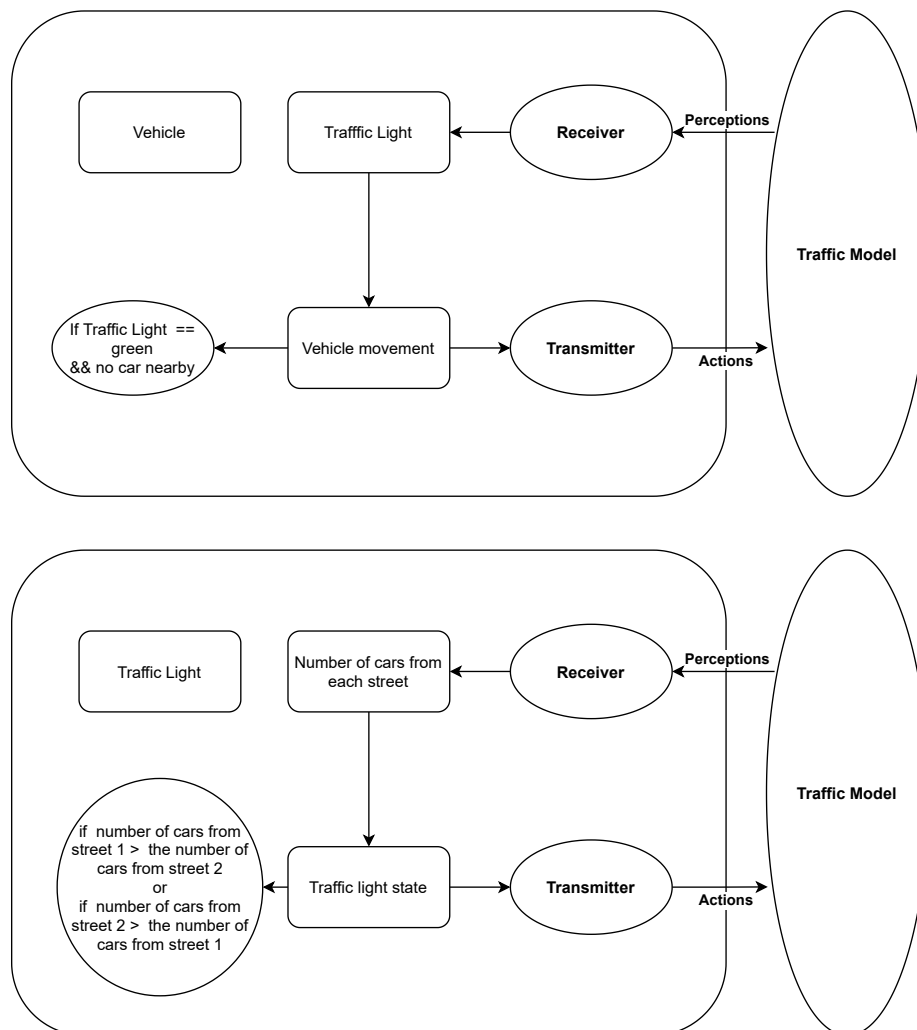
Grupo 2

Miércoles 01 de Diciembre del 2021

Diagrama de clases del modelo



Protocolo de agentes

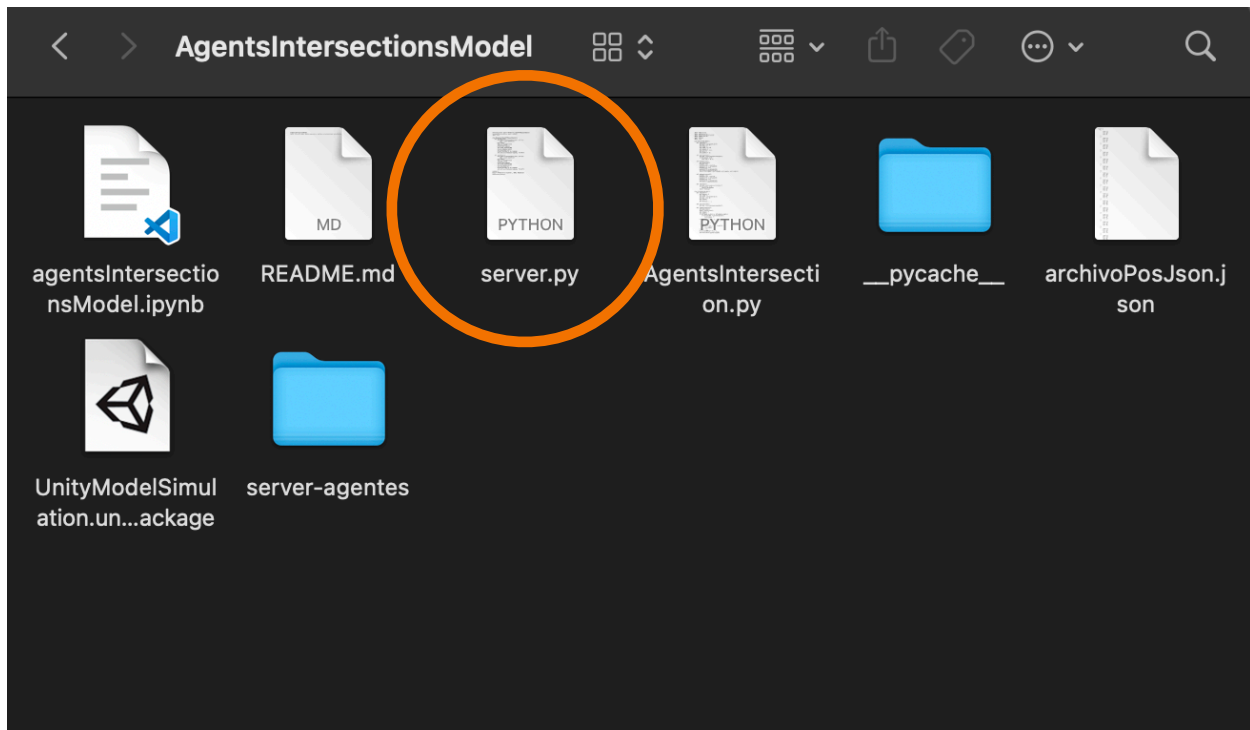


Estrategia cooperativa

La estrategia cooperativa que nosotros implementamos para esta simulación multiagentes con graficas computacionales fue básicamente por medio de la gestión entre los agentes (vehículos) mediante el agente (semáforo), este agente era un gestor entre la movilidad de los agentes (vehículos), en el cual su lógica era de la siguiente manera: se contaba el numero de agentes (vehiculos) de cada calle, y se le daba prioridad a la calle que tuviera mayor numero de agentes (vehículo) y esto era lo que determinaba el color del semáforo; finalmente cuando el semáforo de la calle esta en verde, el carro avanza y si esta en rojo el auto se detiene; algo importante de mencionar es que todos los agentes se encuentran en la misma red (grid) por lo que todos los agentes están en el mismo ambiente y de esta manera se pueden mover sobre la misma red (grid).

Github: <https://github.com/dmojarrot/AgentsIntersectionsModel>

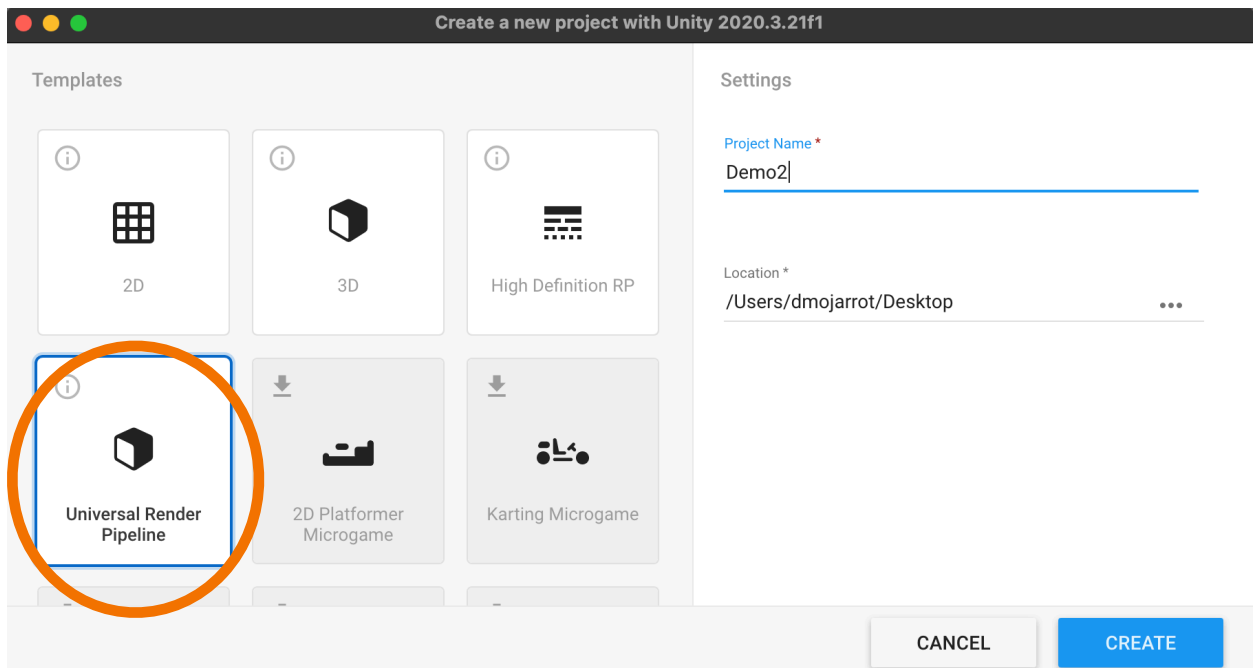
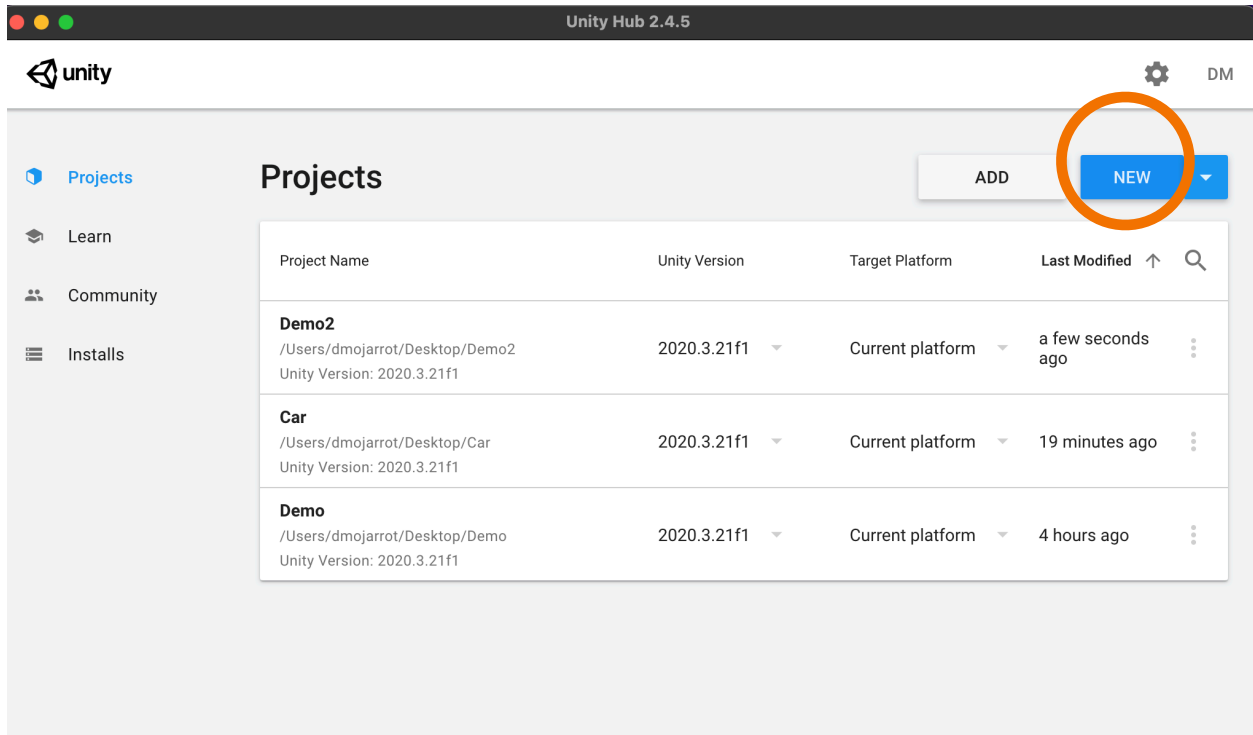
Código:



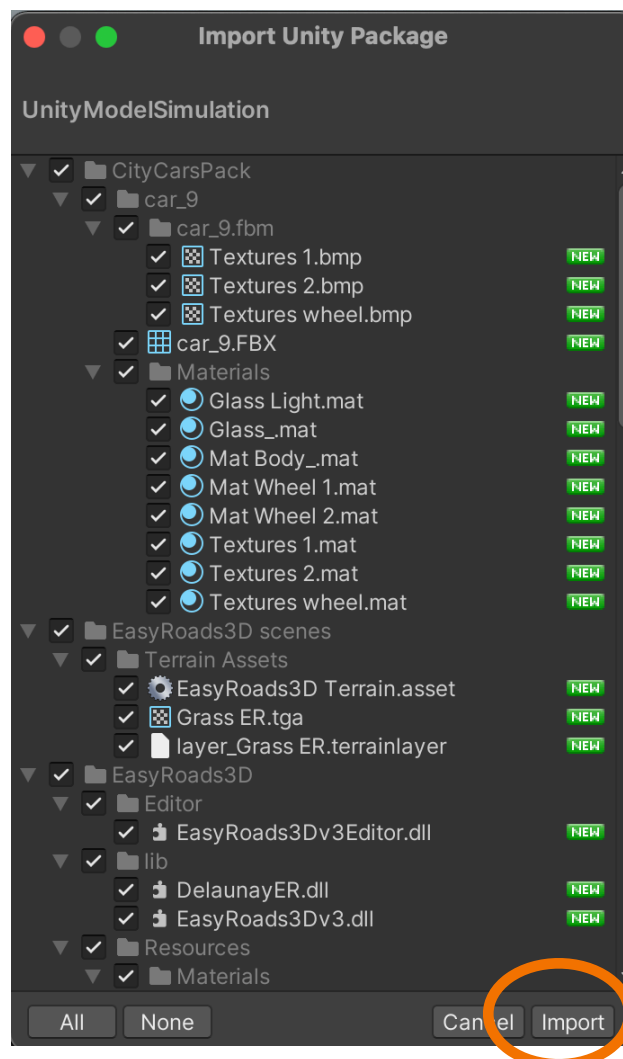
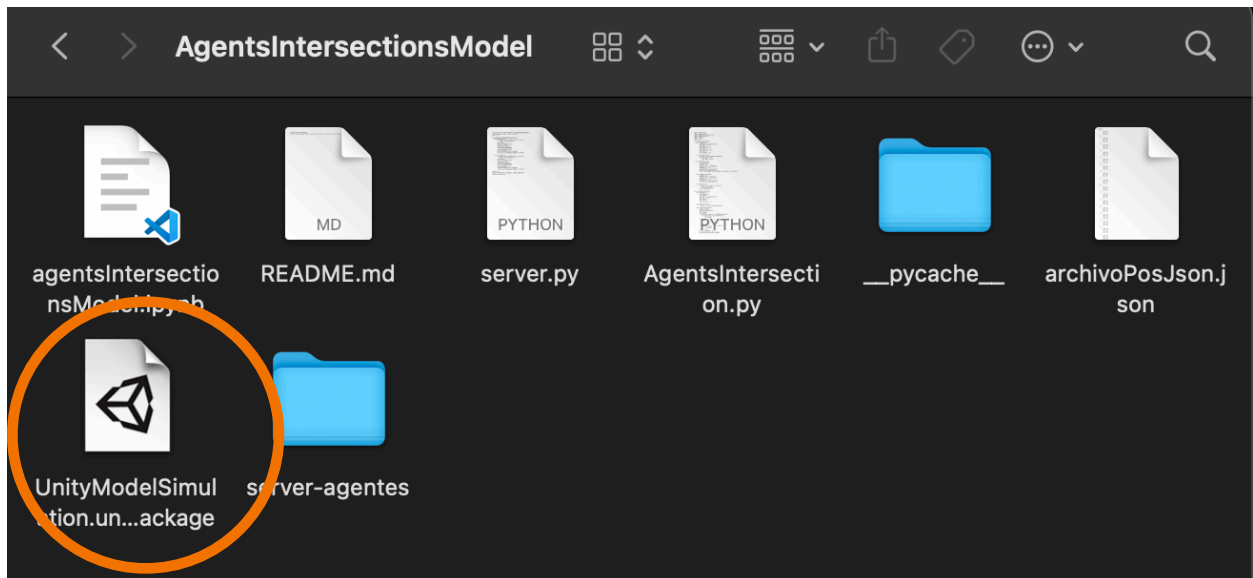
Abrir el archivo server.py y correrlo en la terminal.

```
[(base) -> AgentsIntersectionsModel git:(main) python server.py  
Completed: 498 steps[{ 'id': 0, 'x': 12, 'y': 0, 'z': 0 } { 'id': 0, 'x': 12, 'y':  
0, 'z': 1 }  
{ 'id': 0, 'x': 12, 'y': 0, 'z': 2 } ...  
{ 'id': 14, 'x': 12, 'y': 0, 'z': 12 } { 'id': 14, 'x': 12, 'y': 0, 'z': 13 }  
{ 'id': 14, 'x': 12, 'y': 0, 'z': 14 }]  
Completed: 500 steps[{ 'id': 0, 'x': 12, 'y': 0, 'z': 0 } { 'id': 0, 'x': 12, 'y':  
0, 'z': 1 }  
{ 'id': 0, 'x': 12, 'y': 0, 'z': 2 } ...  
{ 'id': 14, 'x': 12, 'y': 0, 'z': 13 } { 'id': 14, 'x': 12, 'y': 0, 'z': 14 }  
{ 'id': 14, 'x': 12, 'y': 0, 'z': 15 }]
```

Una vez iniciado el localhost en este caso en el puerto 8000, creamos un nuevo proyecto en unity hub, con el template de Universal Render Pipeline.



Finalmente importamos el paquete y lo corremos.



Ejemplo:

