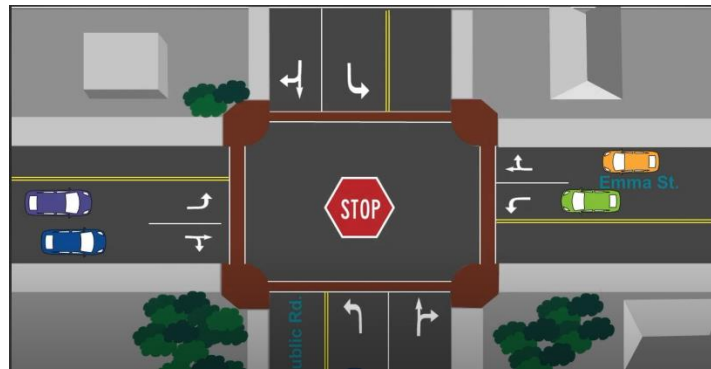


# Revisión 2 - Modelación agentes

Salvador Salgado Normandia A01422874

Stephan Guingor Falcon A01029421

Para esta actividad final se nos ha dado la tarea de simular un four-way intersection la cual está controlada por señales de alto. En este tipo de intersecciones los coches se pueden mover por la intersección dependiendo del carril que estén ocupando. En el caso de ubicarse en el carril de en medio solo pueden girar hacia la izquierda, mientras que en el carril de la derecha pueden moverse ya sea hacia adelante o a la derecha.

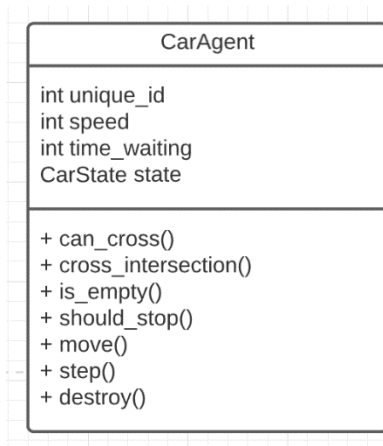


Debido a que esta intersección se controla por medio de señales de tráfico, existe un protocolo que los autos deben de obedecer para evitar accidentes:

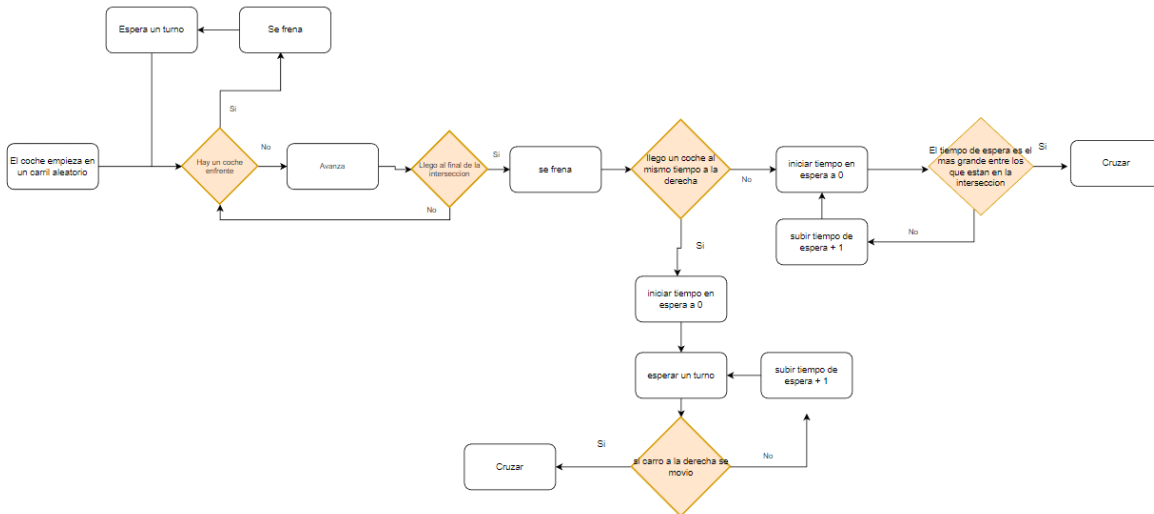
1. Siempre al llegar a la intersección los vehículos deben de hacer un alto total
2. La prioridad de coche avanza primero se otorga según quien llegó antes al cruce
3. En caso de que dos o más vehículos hayan llegado primero se da prioridad al que se encuentre a hasta la derecha.
4. Al momento de cruzar la intersección, el coche debe de dirigirse al carril válido.

Antes de explicar a más detalle el funcionamiento de nuestro agente hemos diseñado distintos diagramas que ejemplifican su comportamiento durante el modelado.

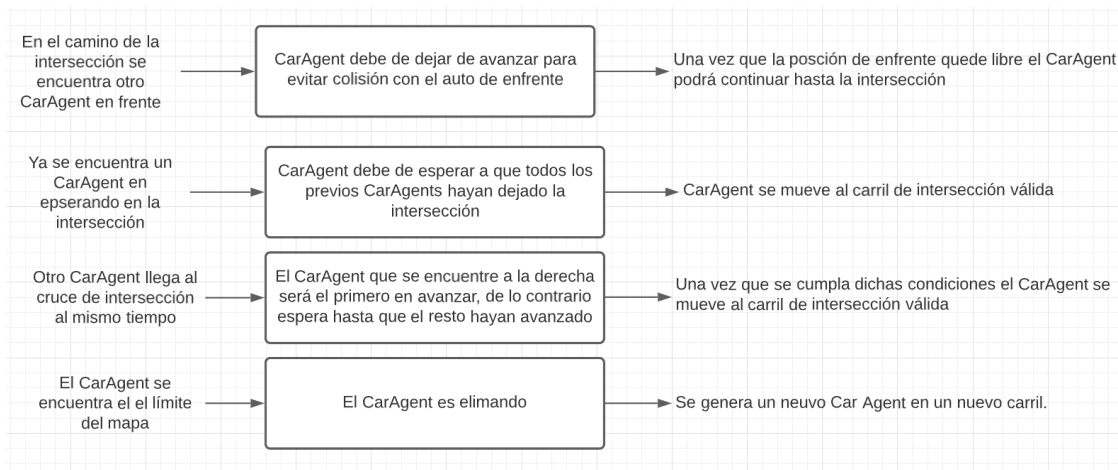
Diagrama de Clase



## Flujo de decisiones CarAgent



## Protocolo de Agentes



Una vez analizado el procedimiento que se realiza para el cruce de intersección fue momento de pensar en cómo sería la mejor manera de implementar este concepto en el modelo de multiagentes. Para este caso consideramos que sólo contaremos con un tipo de agente llamado CarAgent, el cual representaría a un coche real. En similitud con un vehículo, este agente contará con un atributo de velocidad, el cual le dirá qué tan rápido se debe de mover en los carriles. En el caso de que en su carril se encuentren coches enfrente de él, será necesario que permanezca en su posición actual hasta que la posición en frente se encuentre libre. Una vez que el agente llegue a la intersección, este deberá de inicializar su variable de **time\_waiting**, donde cada turno que permanezca en espera de cruzar se irá añadiendo en dicha variable.

Durante cada turno el CarAgent revisará el resto de las posiciones de la intersección en busca de un agente con un mayor tiempo de espera que él. En caso de que no encuentre ninguno será la señal para que inicie el recorrido a hacia el carril valido. Por el contrario, si encuentra uno o más CarAgent con su mismo tiempo de espera será necesario que decidan quien irá primero por medio de quién se encuentre hasta la derecha.

Una vez que el CarAgent se haya incorporado de manera correcta en su nuevo carril este continuará su movimiento hasta una cierta posición donde será destruido del modelo. Esta simulación contará con un máximo de coches que podrán existir al mismo tiempo en el modelo, logrado evitar que se lleguen a saturar los carriles.

En este proyecto estaremos implementando diferentes estados para nuestros agentes, logrando un mayor control y estructura en las diversas acciones que debe de generar en cada step.