Revisión 3 - Avance al 60%

Naomi Anciola Calderon A01750363, Sandra Ximena Téllez Olvera A01752142

Redacten un documento que incluya los siguientes segmentos:

- Descripción del medio ambiente,
- diagramas de agente
- protocolos de interacción finales.
- Código (al menos el 60%) de la implementación de los agentes.
- Código (al menos el 60%) de la implementación de la parte gráfica de la solución.
- Plan de trabajo y aprendizaje adquirido.

En TODAS sus presentaciones deben incluir el plan de trabajo actualizado y el aprendizaje adquirido como equipo. El plan de trabajo debe incluir al menos:

- Las actividades pendientes y el tiempo en el que se realizarán.
- Para las actividades planeadas indicar los responsables de llevarlas a cabo, la fecha en las que las realizarán y el intervalo de esfuerzo estimado.
- Para las actividades finalizadas indicar el tiempo que les tomó realizarlas y la diferencia con su estimación.

Especificaciones de entrega

Por medio de su repositorio del equipo en Github.

El commit deber contener el mensaje "REVISION 3 - 60%"

SMA - Avance al 60%	40 pts El avance presentado corresponde al 60% de la solución del reto	20 pts El avance presentado es menor al 60% de la solución del reto	0 pts No cumple o no entregado	40 pts
Este criterio depende de una competencia de aprendizaje Implementación Gráfica - Avance al 60%	40 pts El avance presentado corresponde al 60% de la solución del reto	20 pts El avance presentado es menor al 60% de la solución del reto	0 pts No cumple o no entregado	40 pts
Este criterio depende de una competencia de aprendizaje Descripción del medio ambiente	5 pts Describe el medio ambiente y entrega análisis justificado de las 5 características del medio ambiente		0 pts No cumple o no entregado	5 pts
Este criterio depende de una competencia de aprendizaje Diagramas de Agentes	5 pts Diagramas de agentes de acuerdo a la especificación AUML		0 pts No cumple o no entregado	5 pts
Este criterio depende de una competencia de aprendizaje Diagramas de Protocolos de Interacción	5 pts Diagramas de protocolos de Interacción de acuerdo a la especificación AUML		0 pts No cumple o no entregado	5 pts
Este criterio depende de una competencia de aprendizaje Plan de trabajo y aprendizaje adquirido	5 pts Entrega plan de trabajo y aprendizaje adquirido de acuerdo a las especificaciones	2 pts Solamente entrega el plan de trabajo o el aprendizaje adquirido.	0 pts No cumple o no entregado	5 pts

Descripción breve del sistema

Los autos se mueven a lo largo de calles:

- cuando llegan a un lugar donde una calle se incorpora a otra, proceden de forma segura
- cuando llegan a bifurcaciones escogen una dirección
- cuando llegan a semáforos siguen indicación (si es verde pasan, si es rojo frenan)
- los autos pueden salir o llegar de los destinos de forma aleatoria

El Entorno

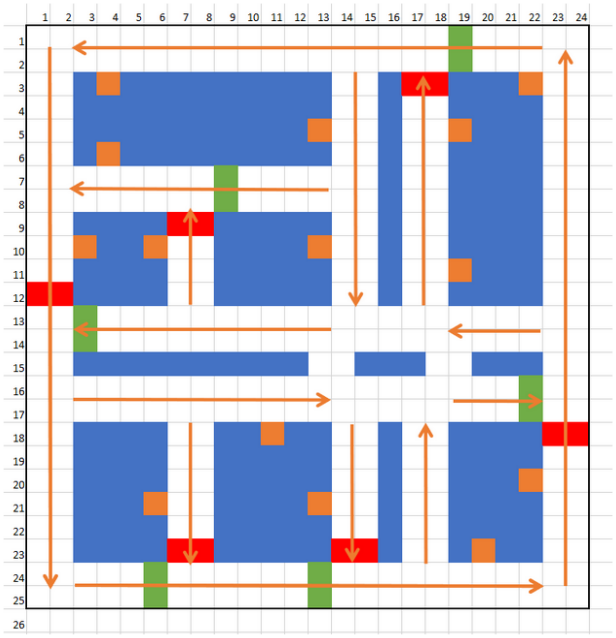
Nuestro entorno es un sistema de calles, las cuales están interconectadas entre ellas, este lo podemos describir en 2 niveles, uno siendo las calles individuales y el segundo siendo el sistema de calles.

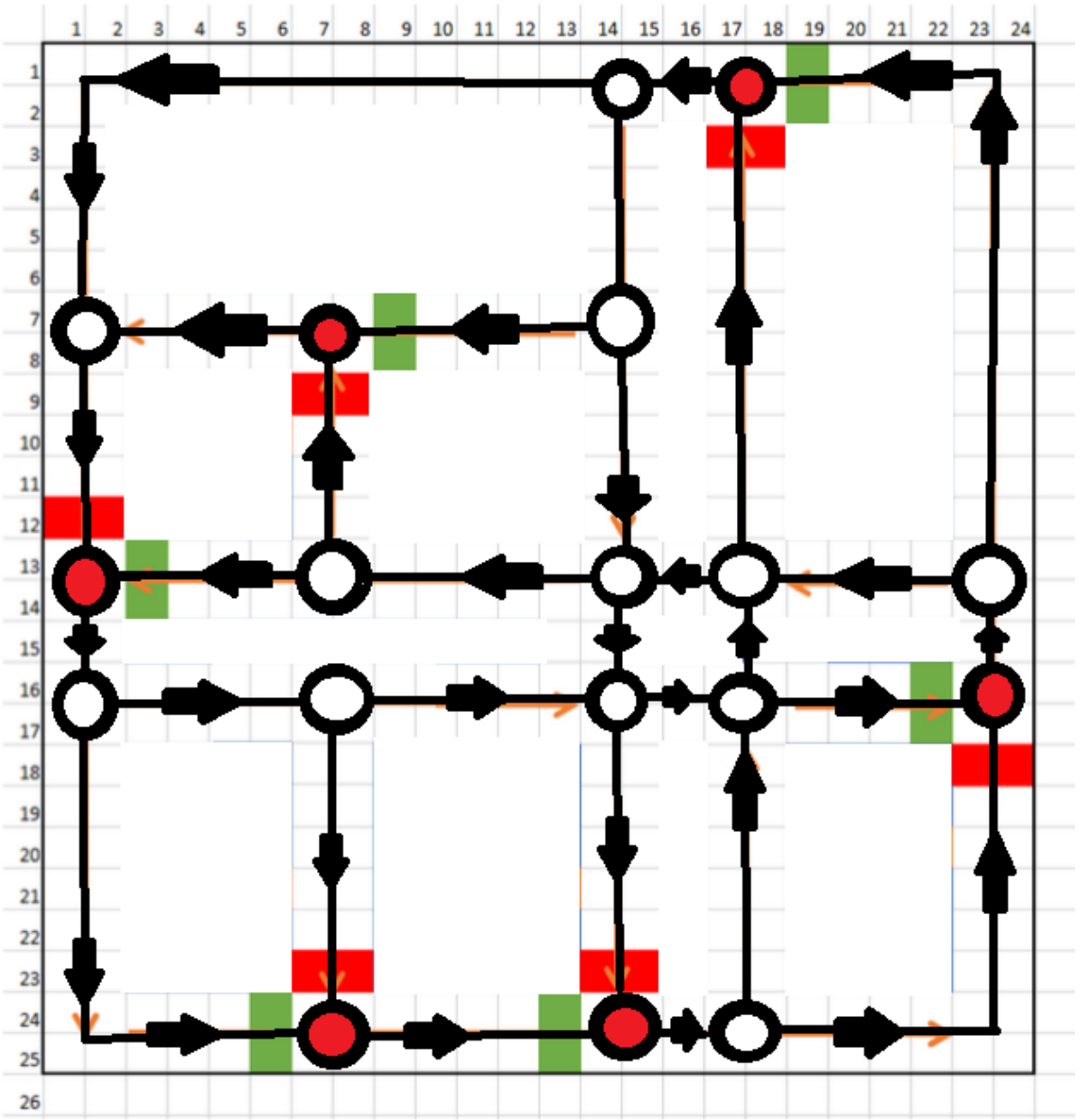
Observando el mapa provisto, las calles siempre tienen 2 carriles, ambos en la misma dirección, que existen a lo largo de 1 de 4 posibles direcciones. Estas calles pueden estar definidas por los 4 puntos 2 donde inician y 2 donde acaban y su dirección.

De las intersecciones entre calles hay de 2 tipos. las primeras, que tienen semáforos, y las segundas sin semáforos (los agentes tienen que poder manejar los 2 tipos). A las que tienen semáforos se les llaman intersecciones controladas.

Las calles están organizadas de forma que en las intersecciones que tienen semáforos nunca hay necesidad de tomar una decisión simultáneamente porque no hay más de una opción para el flujo. Esto reduce significativamente la complejidad de la situación, ya que solo hay 2 semáforos por intersección y no hay varias opciones de flujo.

Por otro lado, en las intersecciones sin semáforos hay de 2 tipos, en las que son instancias de bifurcaciones se tiene que tomar una decisión de cual seguir después, mientras que en las que son casos de que una calle se incorpora a otra, se debe de decidir cuándo cruzar de manera que no ocurran colisiones.





- 12 intersecciones sin semáforos (círculos blancos)
- 6 intersecciones con semáforos (círculos rojos),
    - 24 semáforos (un semáforo por carril, 2 carriles, 2 calles con semáforos cada una)
- 28 calles
- el número de conductores será variable
- 14 destinos

Agentes involucrados

Para este escenario los agentes involucrados son

los conductores,

los semáforos

Los conductores buscan seguir su ruta y cruzar la intersección de forma segura rigiéndose por lo que comunique el semáforo, y los semáforos buscan eficientar el flujo a través de la intersección.

Descripción PEAS de conductor

PEAS Conductor / Persona/ Automovil	
Medida de Actuación	manejo a lo largo de calles, cruce en intersección sin colisiones, seguir semáforos
Entorno	sistema de calles de dos carriles con intersecciones con y sin semáforos, poblado por automóviles
Actuadores	avanzar en 4 direcciones
Sensores	visión (conoce otros coches en la calle cercanos, estado del semáforo)

Descripción PEAS de semáforos

PEAS Semáforos	
Medida de Actuación	emite señales para indicar a los coches cuando cruzar en una intersección, informa al controlador sobre estado del trafico. minimizar tiempo de espera en intersección para automóviles
Entorno	Intersección entre 2 calles
Actuadores	Cambiar de color (verde, amarillo, rojo)
Sensores	visión

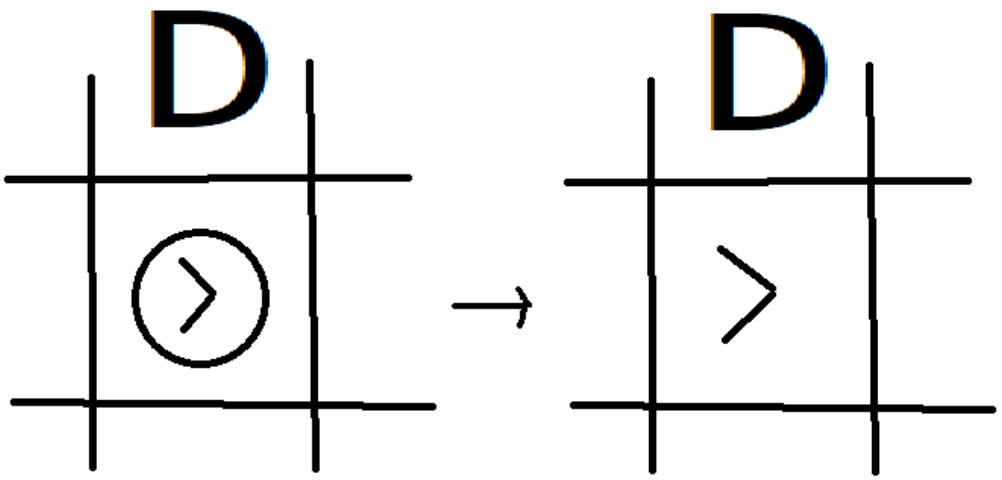
Diagramas de Agente usando AUML

Semáforo
Grupo: *infraestructura* ***rol:*** *regular tráfico*
Servicio: *permite o bloquea el flujo de tráfico a lo largo de las calles, comunica a los autos si pueden o no pasar en una intersección*
Protocolos: *manejar flujo,* *avisar sobre tráfico actual*
Eventos: *Llegó un coche a la intersección -> lo cuenta, estima tiempo de arribo, manda mensaje a semáforos cercanos y updatea el modelo*
Objetivo: *que haya poco tráfico y que se optimice el flujo*

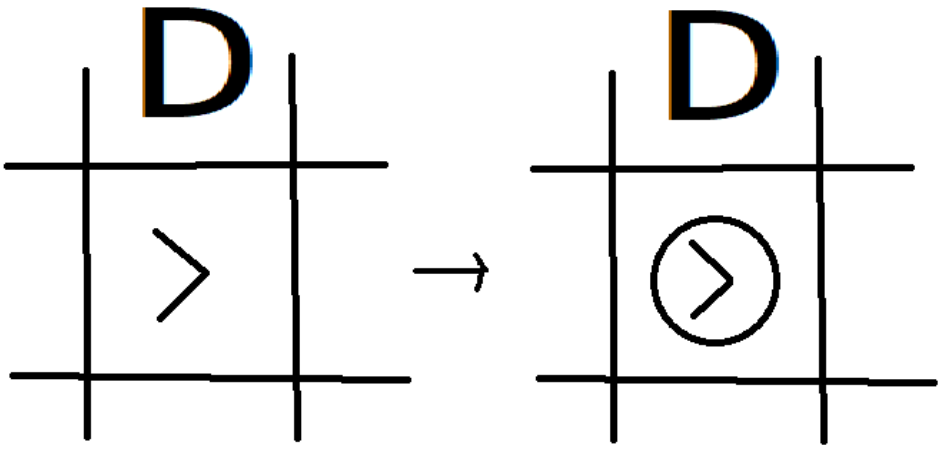
Personas/ Conductores / Automóviles
Grupo: *Civil* ***Rol:*** *movimiento dentro del mundo*
Servicio: *manejar automovil*
Protocolo: *manejar automovil*
Eventos: *B1: intersección con otra calle con semáforo* ● *Semáforo rojo -> espera* ● *Semáforo verde -> cruza* ● *Semáforo amarillo -> Frena Si Estaba en movimiento, Acelera* *B2: Intersección con otra calle sin semáforo* ● *Hay coches visibles? -> Si -> Están en movimiento? -> No -> Cruzar* ● *Hay coches visibles? -> No -> Cruzar* *B3: Objeto en calle* ● *Hay espacio en el otro carril? -> cambiar de carril* Colisión Inicio de viaje / incorporación al tráfico Fin de viaje / llegada al destino
Objetivo: *Manejar, evitar colisiones*
Memoria: ***Objetos en la calle:*** *automóviles visibles -> coches delante (en N carriles) y detrás de si* *semáforo y su color*

Diagramas

Destinos: el principio y fin de cada viaje



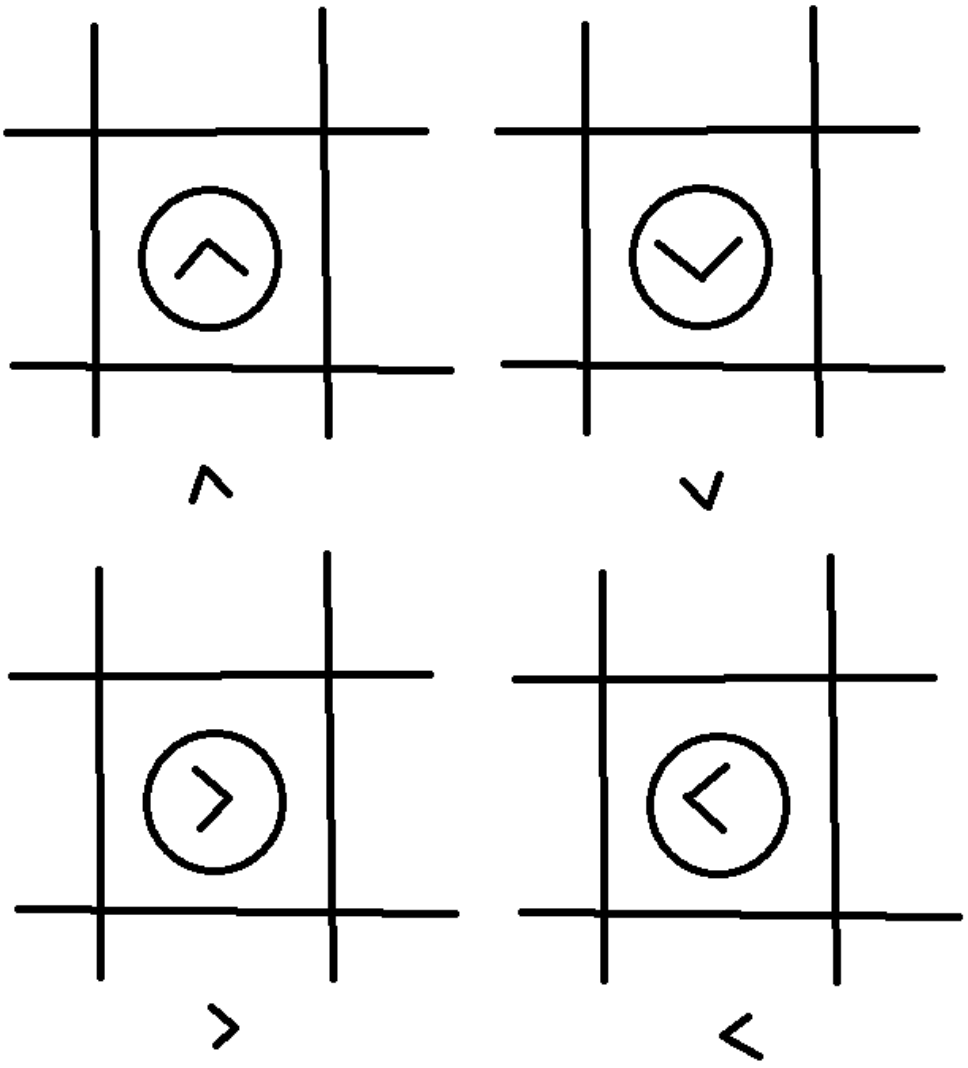
Es posible que un conductor al llegar a un destino, desaparezca



Es posible que un conductor salga de un destino
y se incorpore al tráfico

Diagramas

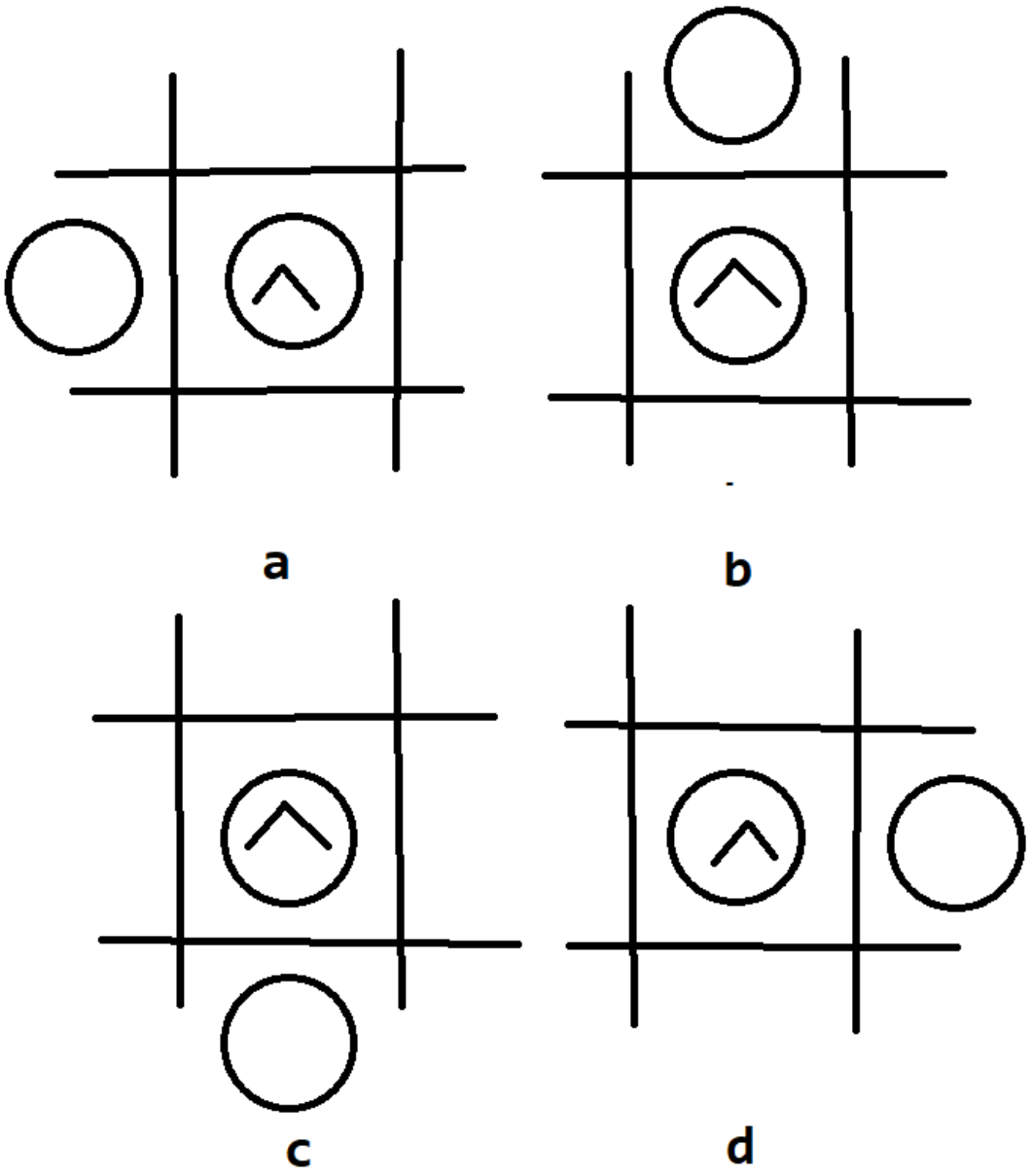
Direcciones posibles de las calles



Las calles pueden tener 4 direcciones posibles,
el conductor las lee para saber a qué celda moverse después

Diagramas

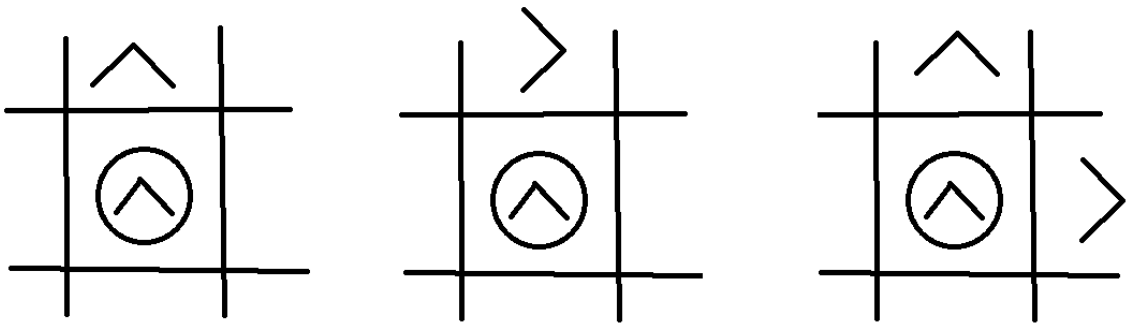
Autos vecinos



a: coche en otro carril, b: coche enfrente,
c: coche detrás, d: coche en otro carril

Diagramas

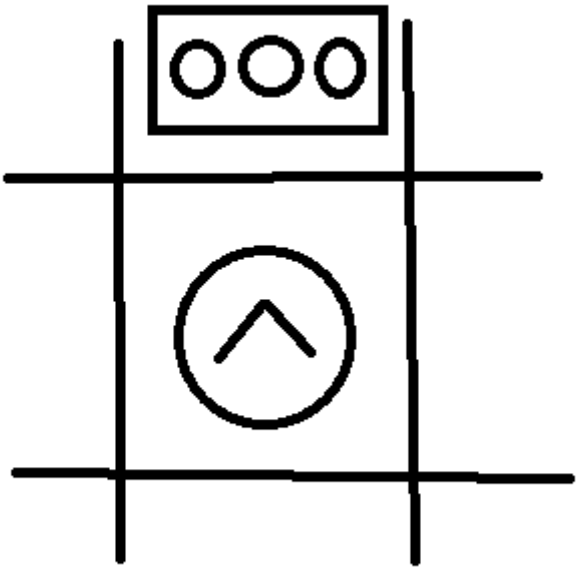
Estados en calle:



a: carril normal, b: vuelta, c: bifurcación

Diagramas

Semáforos



Cuando se llega a un semáforo, se hace lo que indique la luz.

Ontología Final

- [ ] vehículo / coche / conductor
- [ ] calle
- [ ] carril
- [ ] semáforo
- [ ] intersección
- [ ] nombre de calle
- [ ] lugar
- [ ] manejando
- [ ] frenado
- [ ] girar
- [ ] destino
- [ ] obstáculo

class: vehículo / coche / conductor
propiedades:

estado: en movimiento, frenado
Tipo: Bool,

ubicación
Cardinalidad: 1,
Tipo: Pair(X,Y),

class: semáforo
propiedades:
posición en donde se ubica,
Cardinalidad: 1,
Tipo: Pair (X, Y),

Intersección a que pertenece
Cardinalidad: 1,
Tipo: int,

memoria de coches actuales en intersección
Cardinalidad: Variable,
Tipo: String,
Rango: Todos los semáforos,
Dominio: Semáforos, vehículos)

class: obstáculo
propiedades:
posición
tipo Pair (X,Y)

calle a la que pertenece

class: destino
propiedades:
posición
calle a la que pertenece
% de llegada
% de salida

class: calle
propiedades:

nombre,
Cardinalidad: Variable,
Tipo: String,

longitud,
Cardinalidad: Variable,
Tipo: int,

sentido,
Cardinalidad: 4,
Tipo: int, (1,2,3,4 -> N,S,E,W)

número de carriles,
Cardinalidad: 2,
Tipo: Int,

intersecciones en orillas con que conecta,
Cardinalidad: Variable,
Tipo: pair (int) -> idorigen, id_destino,

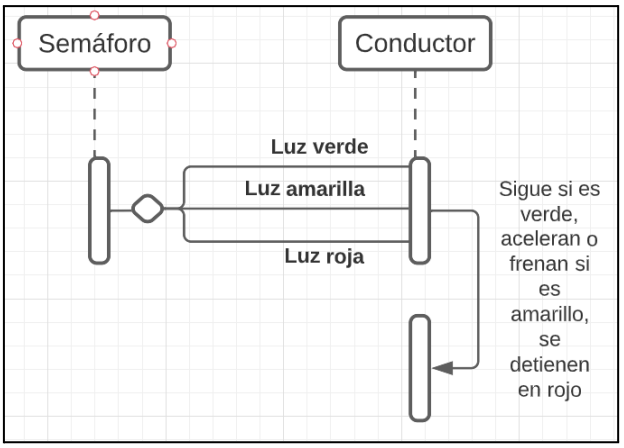
coches que contiene
Cardinalidad: Variable,
Tipo: Int,
Rango: Todos los lugares,
Dominio: Lugares y vehículos

class: intersección
propiedades:

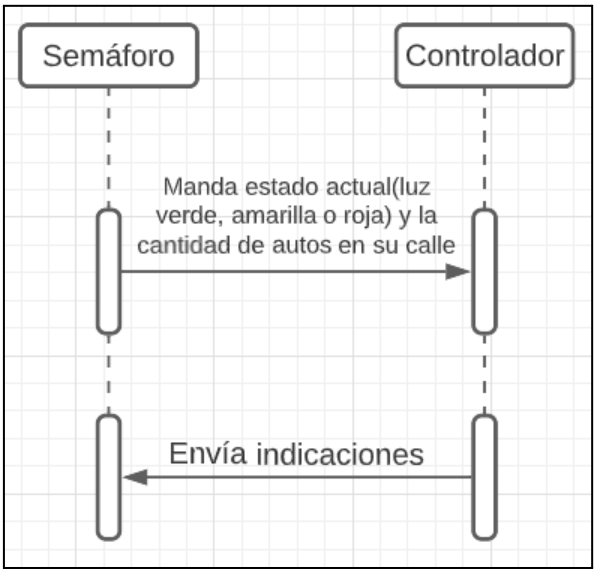
semáforos
calles que conecta
coches que contiene (los contiene si están esperando ahí), y la calle a la que pertenecen

Diagramas de protocolos de interacción

Interacción entre conductor y semáforo

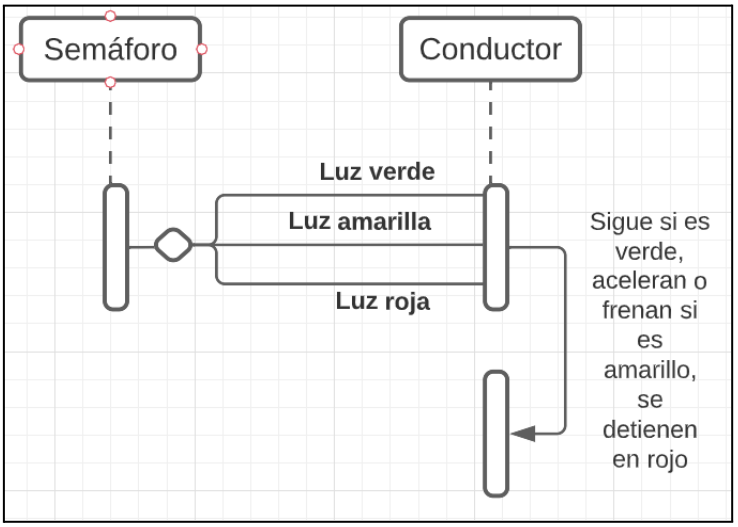


Interacción entre semáforo y controlador



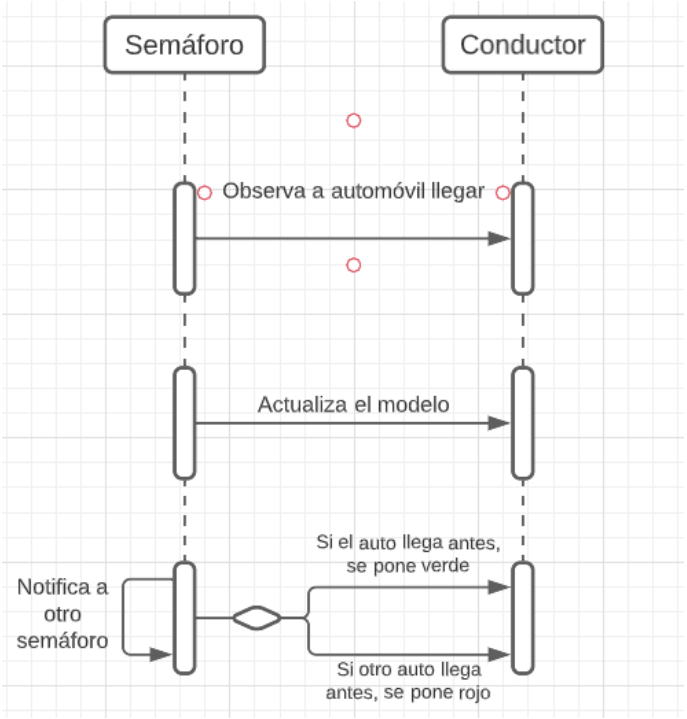
Interacción entre conductor y semáforo 1:

Conductor observa luces del semáforo para decidir si cruzar o no

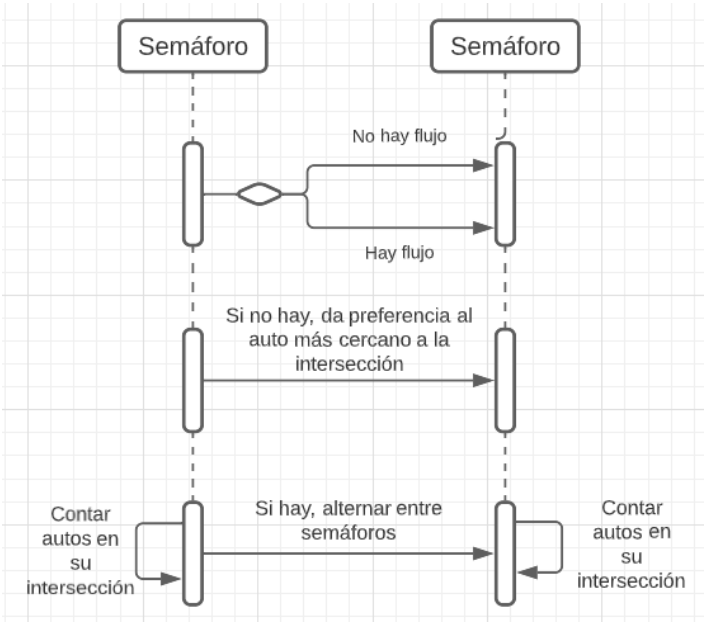


Interacción entre conductor y semáforo 2:

Semáforo monitorea calle para optimizar flujo



Interacción entre semáforos de la misma intersección



Plan de trabajo

En cuanto al modelo:

Para implementar el modelo se han hecho las siguientes cosas: que los destinos generen y absorban automóviles, que los automóviles se mueven a lo largo de las calles siguiendo el sentido de las mismas, que giren de forma segura, que los autos esperen en caso de que un semáforo este en rojo pero sigan su camino si no fuera así, que en caso de que el paso está bloqueado esperen a que se libere espacio para avanzar.

Falta modificar el mapa para que se adecue al que se pide, agregar que los semáforos se coordinen entre ellos y que los autos rebasen en caso de que sea posible, y que al llegar a una bifurcación escojan a veces irse por otra calle. Además si da tiempo estaría bien agregar la funcionalidad de análisis y de mover los parámetros que ofrece mesa. De esto es responsable Naomi, y la idea es acabarlo a lo largo de la semana.

En cuanto a los gráficos:

Para la generación del mapa se utilizó el código de unity brindado por el profesor Octavio, sin embargo, a este se le aplicaron unas cuantas modificaciones, principalmente dentro de los assets.

En cuanto a la conexión entre el modelo y Unity:

En este caso, falta probar el servidor de flask junto al código en mesa, esto se realizará una vez finalizado el código, posteriormente, este servidor se conectará con unity a través de un script por desarrollar.

Actividades faltantes:

Semana 5					
Entrega de avance de 60%	Riesgo alto	Ambas	100%	28/11/2022	1
Ajustes del mapa en mesa		Naomi	75%	29/11/2022	1
Coordinación entre semáforos	Riesgo bajo	Naomi	0%	29/11/2022	1
Servidor de flask		Sandra	80%	30/11/2022	1
Unión del modelo con unity	Según lo previsto	Sandra	0%	30/11/2022	2

Actividades realizadas

Actividad	Tiempo estimado	Tiempo real utilizado
Modelo: que los destinos generen y absorban automóviles	1 hr	3 hr
Modelo: que los autos se muevan por las calles como se debe	3 hr	4 hr
Elección de assets	2 horas	1 hora
Modificación del código de generación de la ciudad en Unity	1 hora	1 hora
Servidor flask	1 hora	1 hr 30 min

Aprendizaje adquirido

A lo largo de este curso aprendimos sobre los diferentes tipos de acercamientos que existen para programar agentes, las formas en las que estos se representan por medio de diferentes diagramas y guías. También aprendimos a implementarlos usando el framework de Mesa, generando así una simulación, la cual, pueda ser conectada a Unity a través de Flask, herramienta, la cual, logra facilitar mucho la conexión.

El conocimiento de la herramienta de Unity resulta de gran utilidad, ya que, a través de la misma, puede hacerse una representación de cualquier tipo de simulación, permitiendo que, sea mucho más atractiva para cualquier persona, es importante también conocer la tienda de Assets, ya que esta permite que, cualquier desarrollo en Unity pueda ser personalizada.