



Tecnológico de Monterrey

M5. Revisión de avance 2

Modelación de sistemas multiagentes con gráficas computacionales (Gpo 570)

A00572499 - Santiago Gutierrez

A00832425 - Daniel Rubies Isla

A01252831 - Luis Carlos Rico Almada

A00837426 - Pedro Gabriel Sánchez Valdez

Equipo 5

Febrero de 2025

1. Introducción	3
Descripción del reto a desarrollar.	3
2. Créditos	4
3. Contexto y Problema	4
3.1 La Simulación de Multiagentes como Herramienta para Resolver Problemas de Movilidad Urbana	5
3.2 Conclusión investigación: Un Futuro Basado en Datos y Modelos	5
4. Objetivos generales	5
4.1 OBJETIVOS DE LA INVESTIGACIÓN	5
4.1 Objetivo General	5
4.2 Objetivos Específicos	5
4.2.1 Diseñar comportamientos variados para peatones que reflejan situaciones del mundo real:	6
4.2.2 Implementar comportamientos realistas en los conductores:	6
4.2.3 Simular escenarios urbanos con variaciones en la infraestructura y el entorno:	6
4.2.4 Incorporar comportamientos proactivos y reactivos en peatones y conductores:	6
4.3 Actividades planeadas	7
5. Restricciones	9
6. Historias de usuario	9
6.1 Historias de usuario Multiagentes	9
6.2 Historias de usuario aplicación gráfica	10
7. Descripción del sistema Multiagente	10
7.1 Diagrama de clase	10
7.2 Descripción agentes	12
7.3 Modelo de los Agentes	13
7.3.1 Simulación del agente con Q-Learning:	13
7.2 Análisis de la solución	14
7.3 Modelo del Entorno	15
7.4 Descripción del ambiente	16
7.5 Modelo de la Negociación	18
7.6 Modelo de la Interacción	20
8. DESCRIPCIÓN DEL MODELADO GRÁFICO	21
8.1 Escena a modelar	21
8.2 Componentes Gráficos	21
8.3 Prefabs	23
8.4 Scripts	25
9. Entregables de administración de proyecto	26
10. Referencias	26

1. INTRODUCCIÓN

DESCRIPCIÓN DEL RETO A DESARROLLAR.

Durante este periodo, el proyecto se centrará en el desarrollo de una simulación realista que modele el tránsito de peatones y vehículos en un entorno urbano dinámico y multifacético. La simulación estará diseñada para replicar las complejas interacciones que ocurren en una calle típica, incluyendo elementos críticos como pasos peatonales, semáforos, zonas de cruce no reguladas y comportamientos que desafían las normas de tránsito, como peatones que cruzan en áreas no permitidas.

En este reto, los peatones serán agentes autónomos con objetivos claros, como desplazarse de un punto de inicio a un destino específico. Cada peatón tomará decisiones basadas en factores como la proximidad de pasos peatonales, la densidad del tráfico, la presencia de semáforos y las situaciones de riesgo. Algunos peatones seguirán las reglas estrictamente, utilizando los pasos peatonales y respetando las señales de tránsito, mientras que otros adoptarán comportamientos más impredecibles, cruzando en áreas no señalizadas o ignorando los semáforos.

Por otro lado, los vehículos no tendrán metas específicas, pero estarán programados para seguir patrones de tráfico comunes, incluyendo detenciones en semáforos, aceleraciones y respuestas ante obstáculos imprevistos, como peatones que irrumpen en su camino. Estos vehículos servirán como elementos clave para generar situaciones de interacción y desafío, obligando a los peatones y conductores virtuales a tomar decisiones tanto reactivas como proactivas para evitar accidentes y garantizar el tránsito seguro.

La simulación también incluirá elementos adicionales del entorno urbano, como:

- **Semáforos peatonales y vehiculares:** Para evaluar cómo las señales de tránsito influyen en el flujo de peatones y vehículos.
- **Zonas de cruce reguladas y no reguladas:** Para observar cómo los peatones interactúan con vehículos en diferentes contextos.
- **Densidad variable de tráfico:** Para explorar escenarios desde calles tranquilas hasta intersecciones congestionadas.
- **Entornos dinámicos:** Con eventos imprevistos como vehículos detenidos o peatones que cambian de rumbo repentinamente.

El proyecto se desarrollará en **Unity**, una plataforma que permitirá crear un entorno 3D detallado y personalizable. Unity ofrecerá la posibilidad de programar comportamientos específicos para los agentes, diseñar un entorno urbano realista y establecer reglas dinámicas que modelen situaciones de tránsito complejas. A través de esta herramienta, se espera obtener datos precisos sobre las interacciones entre peatones y vehículos, incluyendo:

- **Tiempos de espera y cruces seguros.**
- **Frecuencia de incidentes en áreas no reguladas.**
- **Efectividad de los pasos peatonales y semáforos en la reducción de riesgos.**

Este reto busca no solo desarrollar una simulación funcional, sino también fomentar la reflexión sobre temas como la seguridad vial, el diseño inclusivo y las estrategias para mitigar conflictos en la movilidad urbana. Al aplicar conceptos de simulación multiagente, inteligencia artificial y diseño de entornos urbanos, tendremos la oportunidad de experimentar con soluciones innovadoras para mejorar la convivencia entre peatones y vehículos.

Al finalizar el semestre, se espera que la simulación proporcione insights valiosos que puedan aplicarse en la planificación de ciudades más seguras, accesibles y sostenibles. Este proyecto no solo representa un desafío técnico, sino también una oportunidad para explorar cómo los avances en simulación pueden influir en el diseño de políticas públicas y en la mejora de la calidad de vida urbana.

2. CRÉDITOS

Santiago Gutierrez - Desarrollador C# , Unity 3D model

Daniel Rubies Isla - Desarrollador Python

Luis Carlos Rico Almada - Desarrollador Python , Scrum master

Pedro Gabriel Sánchez Valdez - Desarrollador Python ,Unity 3D model

3. CONTEXTO Y PROBLEMA

3.1 LA SIMULACIÓN DE MULTIAGENTES COMO HERRAMIENTA PARA RESOLVER PROBLEMAS DE MOVILIDAD URBANA

La movilidad urbana enfrenta desafíos complejos debido al aumento de la urbanización, la congestión vial y la necesidad de sostenibilidad. La **simulación de multiagentes (MAS)** ofrece un enfoque innovador para analizar, planificar y optimizar sistemas de transporte urbano al modelar el comportamiento de los distintos actores en un entorno virtual dinámico y adaptativo.

3.2 CONCLUSIÓN INVESTIGACIÓN: UN FUTURO BASADO EN DATOS Y MODELOS

La simulación de multiagentes no es solo una herramienta técnica, sino una oportunidad para transformar la forma en que entendemos y solucionamos los problemas de movilidad urbana. Al combinar aspectos sociales, ambientales y económicos en modelos dinámicos, estas simulaciones permiten diseñar políticas e infraestructuras que respondan a las necesidades reales de las ciudades mexicanas. Con un enfoque inclusivo y sostenible, la movilidad puede convertirse en un motor para mejorar la calidad de vida y el desarrollo urbano, haciendo de nuestras ciudades espacios más habitables y equitativos.

4. OBJETIVOS GENERALES

4.1 OBJETIVOS DE LA INVESTIGACIÓN

4.1 OBJETIVO GENERAL

Desarrollar una simulación realista y funcional en Unity que modele las interacciones entre peatones y conductores en un entorno urbano, considerando una amplia variedad de comportamientos y situaciones inesperadas para analizar y mejorar la seguridad vial, la eficiencia del tránsito y la dinámica de movilidad urbana.

A continuación se detallan los objetivos específicos con distintas prioridades, donde, cada prioridad dependerá del tiempo y desarrollo del proyecto, dando mayor atención a aquellas que nos garanticen cumplir con lo esperado pero añadiendo algunas que nos den una mejor perspectiva y entendimiento del problema.

P0- Fundamental

P1- Muy valiosa

P2- Interesante

4.2 OBJETIVOS ESPECÍFICOS

4.2.1 Diseñar comportamientos variados para peatones que reflejan situaciones del mundo real:

- Peatones distraídos que caminan mientras usan dispositivos móviles. **P1**
- Peatones que cambian de dirección abruptamente debido a obstáculos o decisiones de última hora. **P2**
- Peatones que cruzan en áreas no reguladas, ignorando pasos peatonales o semáforos. **P0**

Métricas de medición:

- Porcentaje de peatones que cambian de dirección durante su trayectoria y el impacto en su seguridad.

4.2.2 Implementar comportamientos realistas en los conductores:

- Conductores que respetan las normas de tránsito y dan prioridad a los peatones en los pasos peatonales. **P0**

Métricas de medición:

- Porcentaje de conductores que respetan las normas en pasos peatonales y semáforos y su impacto en la seguridad y eficiencia en el transporte.

4.2.3 Simular escenarios urbanos con variaciones en la infraestructura y el entorno:

- Calles con alta densidad de tráfico y zonas peatonales congestionadas. **P1**

Métricas de medición:

- Tiempo promedio de espera de peatones en cruces regulados frente a semáforos descompuestos.

4.2.4 Incorporar comportamientos proactivos y reactivos en peatones y conductores:

- Peatones que toman decisiones proactivas al buscar cruces seguros. **P0**
- Conductores que reaccionan de manera anticipada al detectar peatones en áreas críticas. **P0**
- **Métricas de medición:**

- Tasa de éxito en decisiones proactivas de los peatones al evitar áreas de riesgo.
- Tiempo promedio de reacción de conductores frente a peatones en situaciones imprevistas y comparación con cómo se da en la realidad.

Este reto busca no solo desarrollar una simulación funcional, sino también fomentar la reflexión sobre temas como la seguridad vial, el diseño inclusivo y las estrategias para mitigar conflictos en la movilidad urbana. Al aplicar conceptos de simulación multiagente, inteligencia artificial y diseño de entornos urbanos, tendremos la oportunidad de experimentar con soluciones innovadoras para mejorar la convivencia entre peatones y vehículos.

Al finalizar el semestre, se espera que la simulación proporcione insights valiosos que puedan aplicarse en la planificación de ciudades más seguras, accesibles y sostenibles. Este proyecto no solo representa un desafío técnico, sino también una oportunidad para explorar cómo los avances en simulación pueden influir en el diseño de políticas públicas y en la mejora de la calidad de vida urbana.

4.3 ACTIVIDADES PLANEADAS

M5. Revisión de avance 3: Correcciones e implementación de las tareas pendientes para este entregable final.

- En cada clase debe estar especificado los tipos de mensajes (performativas) que cada agente puede manejar (los CAs) y la funcionalidad del tipo de razonamiento (métodos y atributos correspondientes), ya sea Práctico, Deductivo, o Híbrido (o Vertical u Horizontal (Reactivo)).
- Para los protocolos de interacción, incluye al menos un protocolo que se base en las interacciones vistas en clase: Contract Net, Votación, Subasta, Negociación.

Tarea	Responsable	Esfuerzo (horas)	Fecha de inicio	Fecha de fin	Impedimentos
Diagramas de clase y protocolos de	Luis Carlos	1 hora	4 de Febrero	4 de Febrero	Revisión de diagramas

interacción finales.					
Conductores que respetan las normas de tránsito y dan prioridad a los peatones en los pasos peatonales.	Pedro	8 horas	2 de Febrero	5 de Febrero	Implementar correctamente los peatones
Peatones que cruzan en áreas no reguladas, ignorando pasos peatonales o semáforos.	Daniel	4 horas	2 de Febrero	4 de Febrero	Ninguno
Áreas sin infraestructura adecuada para cruces seguros.	Santiago	4 horas	2 de Febrero	3 de Febrero	Cambios del ambiente
Peatones que toman decisiones proactivas al buscar cruces seguros.	Luis Rico	8 horas	2 de Febrero	4 de Febrero	Terminar de implementar los peatones correctamente
Conductores que reaccionan de manera anticipada al detectar	Pedro	6 horas	2 de Febrero	5 de Febrero	Terminar de implementar los agentes de coche

peatones en áreas críticas.					
Uso de semáforos funcionales y su impacto en la reducción de incidentes.	Daniel	3 horas	2 de Febrero	5 de Febrero	Agentes de coche y peatones implementados correctamente

5. RESTRICCIONES

Algunas restricciones que estamos encontrando al momento de el diseño de nuestras calles es como funcionan los semáforos para que los vehículos puedan girar sin interrumpir tanto el tráfico , pero igualmente garantizando que todos los agentes cumplan con su objetivo. Así mismo estamos con la restricción de cómo funcionarán los semáforos y cómo los programaremos para que igualmente pueda funcionar con la lógica de nuestro proyecto/ tráfico.

6. HISTORIAS DE USUARIO

6.1 HISTORIAS DE USUARIO MULTIAGENTES

HU1: Como peatón, quiero poder llegar a mi destino siguiendo rutas definidas, para cumplir con mis objetivos de desplazamiento dentro del entorno urbano.

HU2: Como peatón, quiero evaluar las condiciones de mi entorno, como proximidad de pasos peatonales y densidad de tráfico, para decidir si cruzar en un lugar permitido o no.

HU3: Como vehículo, quiero detenerme frente a un semáforo en rojo y reanudar mi marcha cuando esté en verde, para seguir las reglas de tránsito.

6.2 HISTORIAS DE USUARIO APLICACIÓN GRÁFICA

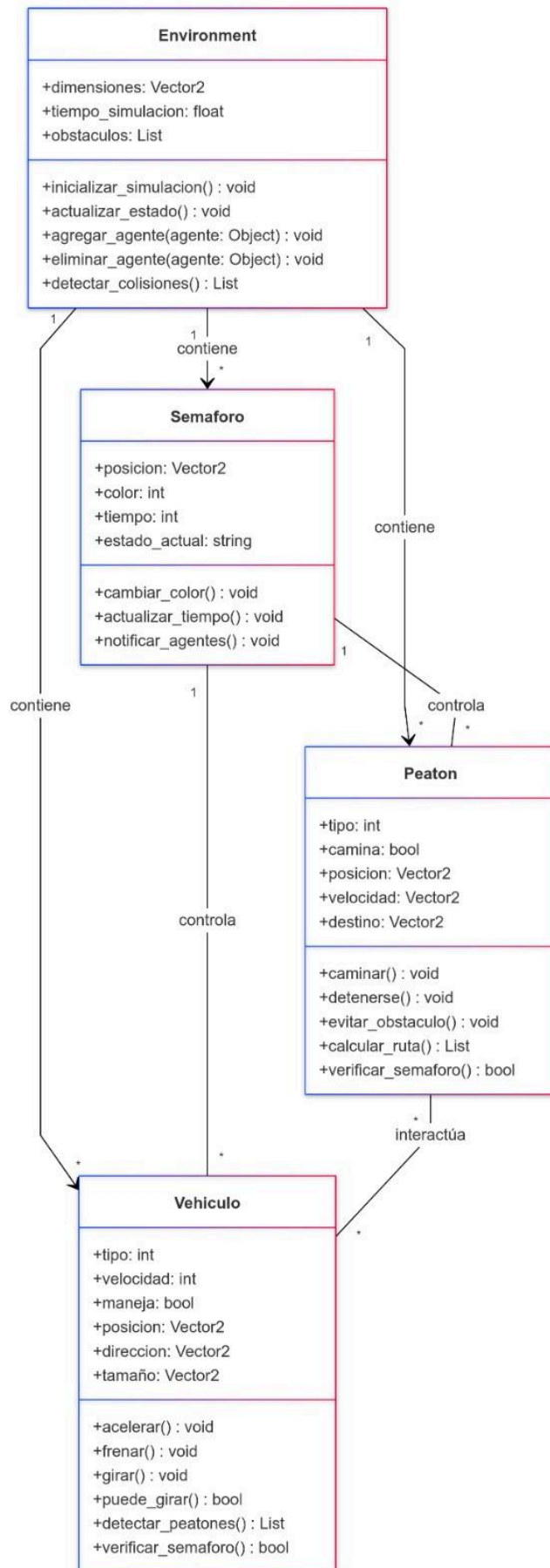
HU4: Como usuario de la simulación, quiero visualizar un entorno urbano en 3D con elementos como pasos peatonales, semáforos y zonas de cruce, para comprender cómo ocurren las interacciones entre peatones y vehículos.

HU5: Como usuario de la simulación, quiero observar el comportamiento de peatones que siguen reglas de tránsito y aquellos que las ignoran, para analizar las diferencias en su interacción con vehículos

7. DESCRIPCIÓN DEL SISTEMA MULTIAGENTE

7.1 DIAGRAMA DE CLASE

(El diagrama está abajo, es muy largo)



7.2 DESCRIPCIÓN AGENTES

Pensamos que los agentes que se encontrarán son :

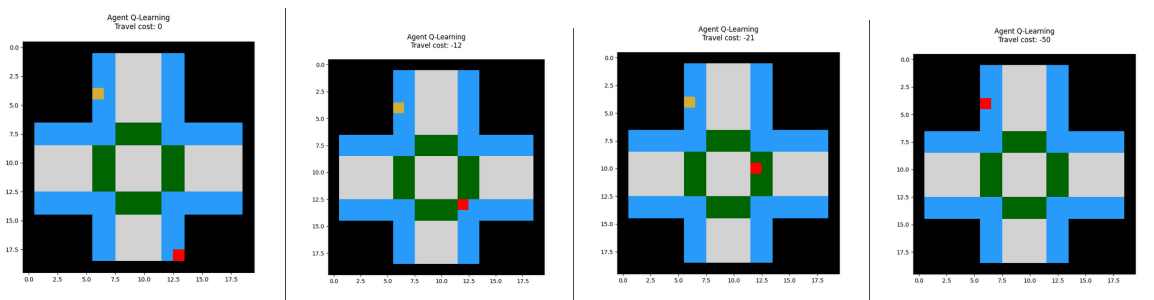
- El Environment es donde nuestros agentes estarán, tendrá una matriz, la cual será responsable para poder manejar el flujo de los diferentes agentes tales como los vehículos y peatones, pero a su vez, escuchando al agente del semáforo, el cual dictará qué tipo de agente entre vehículo y peatón puede pasar por cierta coordenada . Constituye de métodos para iniciar el modelo, actualizar el estado de su grid, podrá iniciar a los agentes, eliminarlo, y detectar las colisiones entre agentes.
- El peatón, el cual tendrá como atributos el tipo , que partirá en persona sin discapacidad 0 , persona con discapacidad 1 y un atributo de tipo bool que determina si el peatón está caminando o no. Sus métodos son caminar , el cual no recibirá parámetros ni regresará algún valor, solo se activará cuando la variable de camina sea true y caminará hacia la dirección de su objetivo respetando las zonas en las que pueda caminar, detenerse no recibe parámetros ni regresa algún valor, solo se activa si su siguiente paso lo pone en peligro, el método detenerse es para cambiar la variable de caminar a falso, lo que detiene completamente al peatón , por último, el evitar obstáculo es cuando existe alguna anomalía u otro agente persona que impida el paso, por lo que se realiza un cambio de posición para evitar la colisión.
- El vehículo comparten atributos similares al peatón, como el tipo que hace mención a coche 0 , camión 1, el atributo maneja tiene la misma lógica que el atributo de camina de la clase Peatón y por último la velocidad, esta variable nos servirá para poder manipular la velocidad de los vehículos y observar que parámetros son los necesarios para que el vehículo pueda frenar correctamente. En cuanto a los métodos, existen los de acelerar , el cual no toma parámetros ni regresa un valor, solo modificará la velocidad hasta llegar a la velocidad máxima por cada step/fotograma de la simulación , para frenar, no recibe parámetros ni regresa algún valor, de igual manera a diferencia del peatón, el vehículo de manera natural no puede frenar a velocidad 0 en menos de 1 segundo y menos en velocidad máxima, por lo que esta función se aplicará cuando el semáforo esté en amarillo , e irá reduciendo la velocidad por cada step que ocurra , por último, el método girar no recibe parametros ni regresa algún valor, pero este método depende de el método puede_girar, el cual no recibe parametros y revisa que el vehículo pueda dar una vuelta de forma segura , de ser posible regresa True, lo cual permitirá realizar la acción de girar al vehículo, de lo contrario, no podrá dar vuelta.

- Por último, el semáforo podría ser considerado un agente debido a que tiene diferentes estados, los cuales se define como el atributo color, verde, amarillo y rojo, lo cual modifica el enviroment, permitiendo a los peatones cruzar, detenerse y manejar el flujo vehicular, por lo que consideramos que es el agente más importante de esta simulación , además contará con el atributo de tiempo, lo cual contara los segundos, que cada uno tendrá para cambiar de color con ayuda de el método cambniar_color, el cuál no recibe parámetros ni regresa algún valor.

7.3 MODELO DE LOS AGENTES

7.3.1 SIMULACIÓN DEL AGENTE CON Q-LEARNING:

Realizamos dos prototipos para la simulación de los agentes usando Q-learning y A-estrella.



Con Q-learning creamos un ambiente discreto donde se evalúa cada cuadro con un valor de preferencia de los agentes. Esto mismo se pinta de la manera que observamos en las fotos del simulacro. El método de aprendizaje le da al agente una ruta casi óptima para llegar a su objetivo.

```
e, r, g, w, m, explorer, goal = 100, 21, 3, 2, 10, -1, -2
```

Para el ejemplo de arriba, vemos los valores de preferencia que tiene el agente de un peatón por los espacios de la matriz en la que habita. Nos indica el valor “w” azul que en este caso representa la banqueta es la que menos le pesa por así decirlo por lo que a la hora de llamar a la función “train” de entrenamiento, las matrices de decisiones van a inclinarse a que el peatón no salga de la banqueta. Por segundo, el valor “g” de color verde solamente tiene un 3 lo cual hace que el agente peatonal escoga caminar sobre los cruces si no encuentra camino su objetivo por la banqueta. Este orden de prioridad termina con los edificios y la calle y ambos son altamente improbables (aunque la

calle es más probable que el edificio) así efectivamente implementando una manera de evitar obstáculos en forma de cruces indebidos en choques de calle o rodear edificios por ejemplo.

```
e, r, g, w, m, explorer, goal = 100, 2, 3, 21, 10, -1, -2
```

La facilidad que nos brinda el modelo es que al cambiar estos valores de prioridad, podemos hacer nuevos tipos de agentes. En este caso, el valor “r” y “w” fueron intercambiados desde como estaban en el agente peatonal significando que ahora el nuevo agente preferiría navegar por la calle y evitará altamente las banquetas así simulando lo que sería el agente de un carro.

7.2 ANÁLISIS DE LA SOLUCIÓN

¿Por qué se seleccionó este modelo multi-agente? Q - learning permite que los agentes aprendan lo cual es más realista para una simulación

¿Qué variables se consideraron en la toma de decisiones? Estado de semáforo= posición/orientación= tipo de celda= recompensas=

¿Cómo interactúan estas variables con los resultados de la simulación? Epsilon controla la exploración= recompensas guían rutas= orientación penaliza= semáforos regulan cruces

¿Por qué se eligió este diseño gráfico? Colores por cada elemento para una clara visualización= hay semáforos= cruces= direcciones

¿Cuáles son las ventajas de la solución final? Aprendizaje adaptativo= Múltiples agentes = Reglas realistas y Visual claro

¿Cuáles son las desventajas? Largo entrenamiento = Baja escalabilidad

¿Qué modificaciones podrían mejorar o eliminar estas desventajas? Optimizar entrenamiento

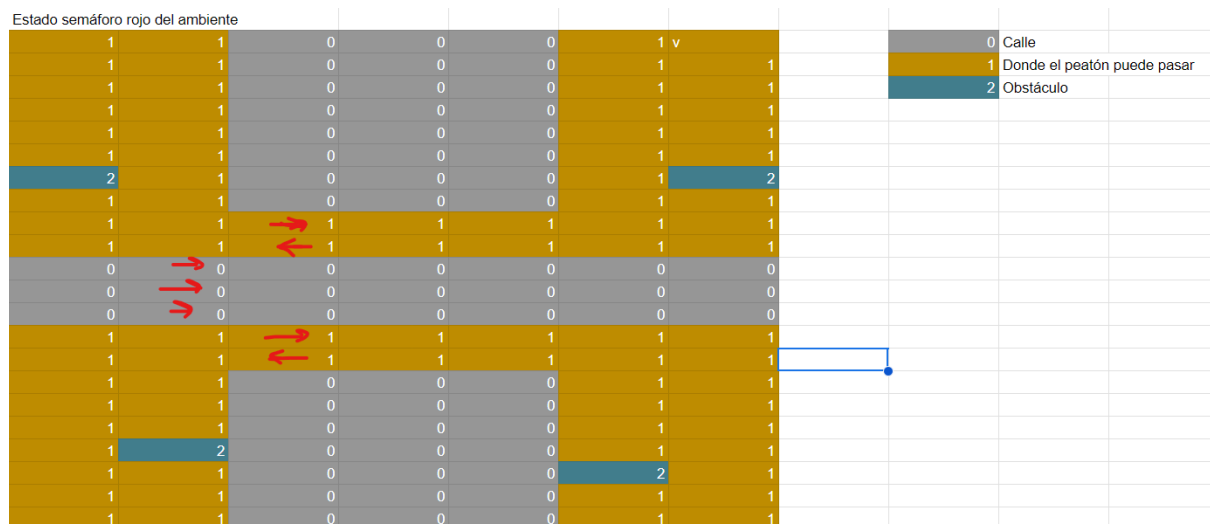
7.3 MODELO DEL ENTORNO

Estado semáforo verde del ambiente													
1	1	0	0	0	0	1	v			0	Calle		
1	1	0	0	0	0	1	1	1		1	Donde el peatón puede pasar		
1	1	0	0	0	0	1	1	1		2	Obstáculo		
1	1	0	0	0	0	1	1	1					
1	1	0	0	0	0	1	1	1					
1	1	0	0	0	0	1	1	1					
1	1	0	0	0	0	1	1	1					
2	1	0	0	0	0	1	1	2					
1	1	0	0	0	0	1	1	1					
1	1	0	0	0	0	1	1	1					
0	0	0	0	0	0	0	0	0					
0	0	0	0	0	0	0	0	0					
0	0	0	0	0	0	0	0	0					
0	0	0	0	0	0	0	0	0					
1	1	0	0	0	0	1	1	1					
1	1	0	0	0	0	1	1	1					
1	1	0	0	0	0	1	1	1					
1	1	0	0	0	0	1	1	1					
1	1	0	0	0	0	1	1	1					
1	2	0	0	0	0	1	1	1					
1	1	0	0	0	0	2	1	1					
1	1	0	0	0	0	1	1	1					
1	1	0	0	0	0	1	1	1					

Este es un ejemplo de nuestro ambiente, tal cual es un grid que no tiene agentes por el momento, los 1's marcan el espacio donde los peatones pueden caminar, al inicio solo podrán caminar por la banqueta, donde habrá casos donde tengan que tener una comunicación para ver quién pasa primero al momento de encontrar un obstáculo. Los 0's representan la calle donde los vehículos transitan.

Estado semáforo rojo del ambiente													
1	1	0	0	0	0	1	v			0	Calle		
1	1	0	0	0	0	1	1	1		1	Donde el peatón puede pasar		
1	1	0	0	0	0	1	1	1		2	Obstáculo		
1	1	0	0	0	0	1	1	1					
1	1	0	0	0	0	1	1	1					
1	1	0	0	0	0	1	1	1					
2	1	0	0	0	0	1	1	2					
1	1	0	0	0	0	1	1	1					
1	1	0	0	0	0	1	1	1					
1	1	0	0	0	0	1	1	1					
0	0	0	0	0	0	0	0	0					
0	0	0	0	0	0	0	0	0					
0	0	0	0	0	0	0	0	0					
1	1	0	0	0	0	1	1	1					
1	1	0	0	0	0	1	1	1					
1	1	0	0	0	0	1	1	1					
1	1	0	0	0	0	1	1	1					
1	1	0	0	0	0	1	1	1					
1	2	0	0	0	0	1	1	1					
1	1	0	0	0	0	1	1	1					
1	1	0	0	0	0	1	1	1					
1	1	0	0	0	0	1	1	1					

El segundo ejemplo es cuando el semáforo cambia a color rojo, los cruces peatonales podrán marcarse como 1, permitiendo así que los agentes (peatones) puedan caminar. Además, es importante mencionar que tanto los vehículos que van hacia el norte, como los peatones que van al norte y sur, que necesitan cruzar, podrán hacerlo una vez el semáforo convierta el estado en el que se encuentra el ambiente.



Al igual que la imagen anterior, el semáforo ahora permitirá que los peatones y vehículos que vayan hacia el oeste y este, puedan avanzar sin ningún problema.

7.4 DESCRIPCIÓN DEL AMBIENTE

Descrito de manera formal, el ambiente con el que trabajamos es discreto por ser un Grid de AgentPy, es accesible ya que todos los agentes tienen acceso a la matriz de información tal como la posición de los demás, determinístico porque el ambiente no cambia de estados en base a circunstancias externas y es predecible en sí mismo, episódico porque correr la simulación no es influenciada por ejecuciones previas ni influencia en futuras, y estática por no ser cambiante.

e r g w

100, 10, 3, 2,

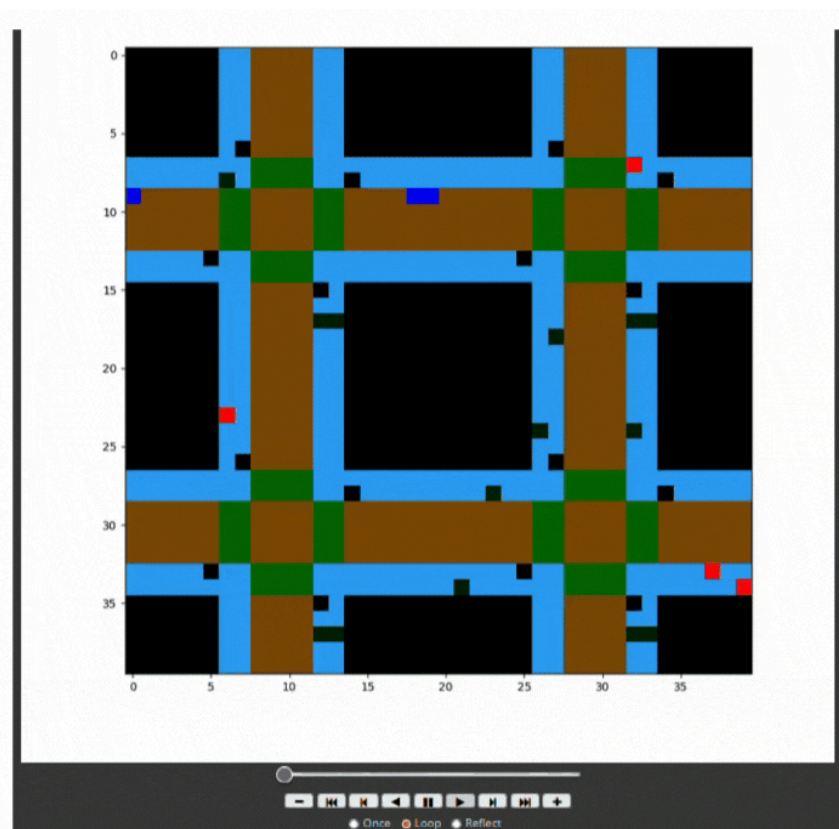
explorer goal car

-1, -2, -3

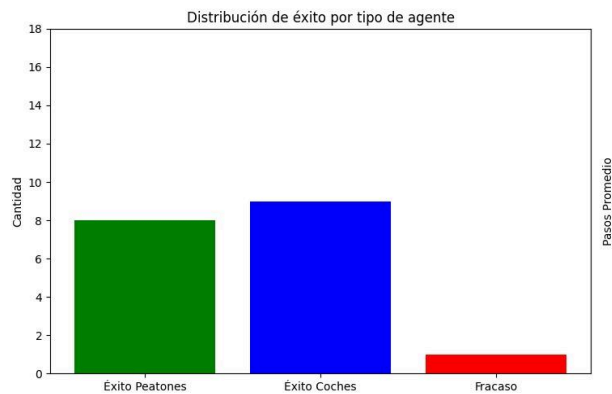
n,s,z,x,p,b,l

[illegible]

A continuación una demostración de cómo se maneja el ambiente con los múltiples agentes descritos anteriormente.

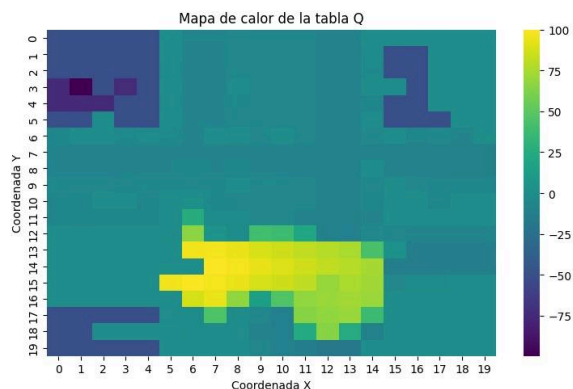


7.5 METRICAS OBTENIDAS



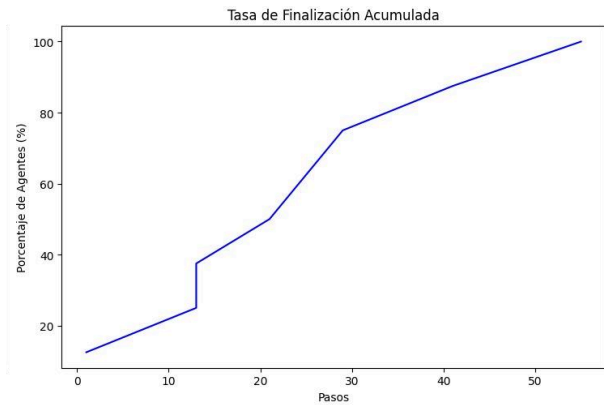
1. Distribución de éxito por tipo de agente:

- Este gráfico de barras muestra los resultados de la simulación separados por tipo de agente
- La barra verde representa aproximadamente 8 peatones que tuvieron éxito
- La barra azul muestra cerca de 9 coches que lograron su objetivo
- La barra roja indica un número muy bajo de fracasos (1)
- Esto sugiere que la mayoría de los agentes fueron exitosos en alcanzar sus metas



2. Mapa de calor de la tabla Q:

- Representa visualmente los valores aprendidos por el algoritmo Q-learning
- Las zonas amarillas indican áreas de alta recompensa (valores cercanos a 100)
- Las zonas azul oscuro representan áreas a evitar (valores negativos)
- Las zonas verdes/turquesa son áreas neutras o de transición
- Se puede observar un camino claro (amarillo) que representa la ruta óptima aprendida



3. Tasa de finalización

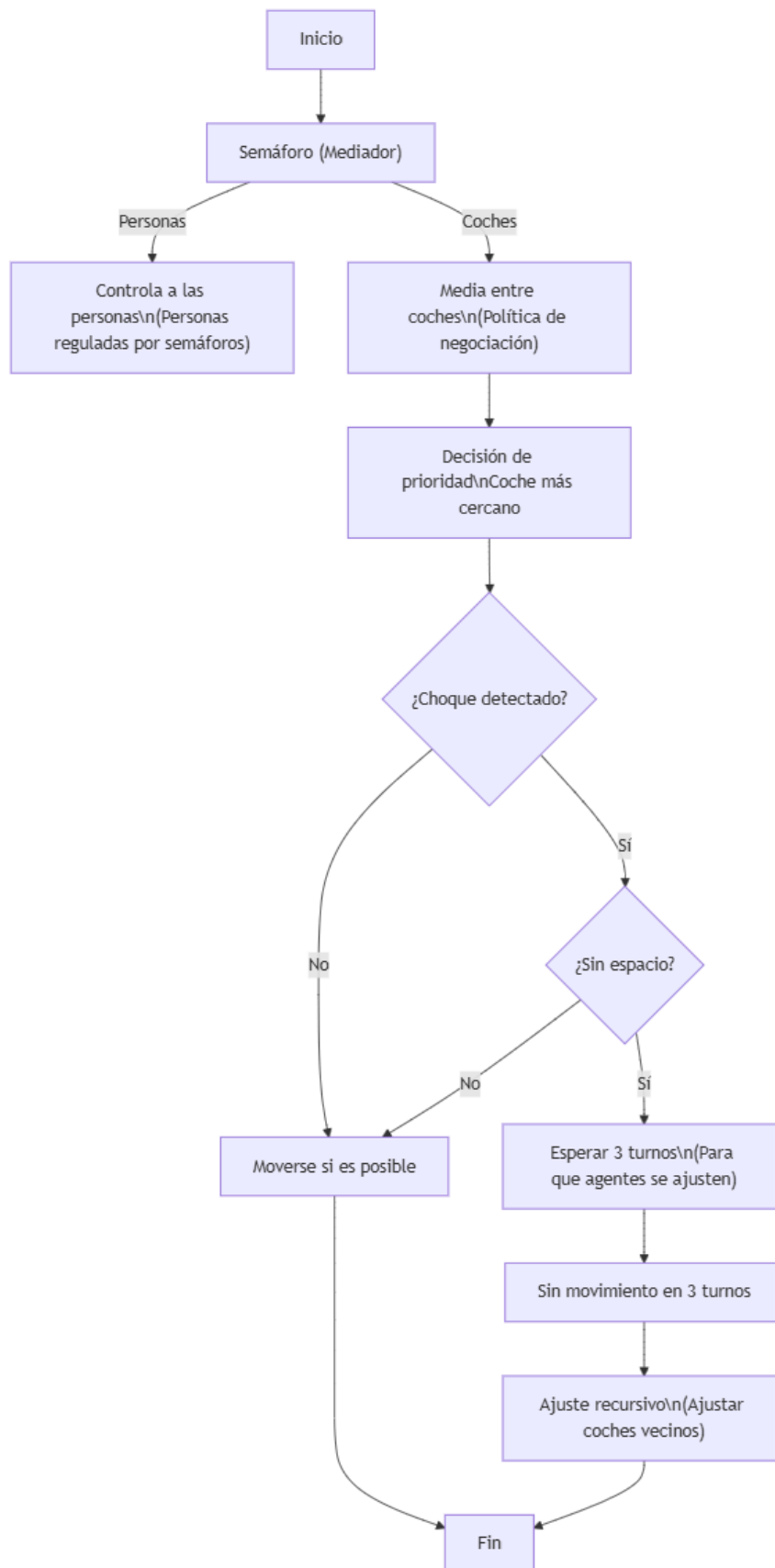
- Gráfico que muestra el progreso de los agentes a lo largo del tiempo
- La línea azul aumenta gradualmente hasta alcanzar el 100%
- Indica que todos los agentes eventualmente completaron sus objetivos

Conclusión sobre modelado

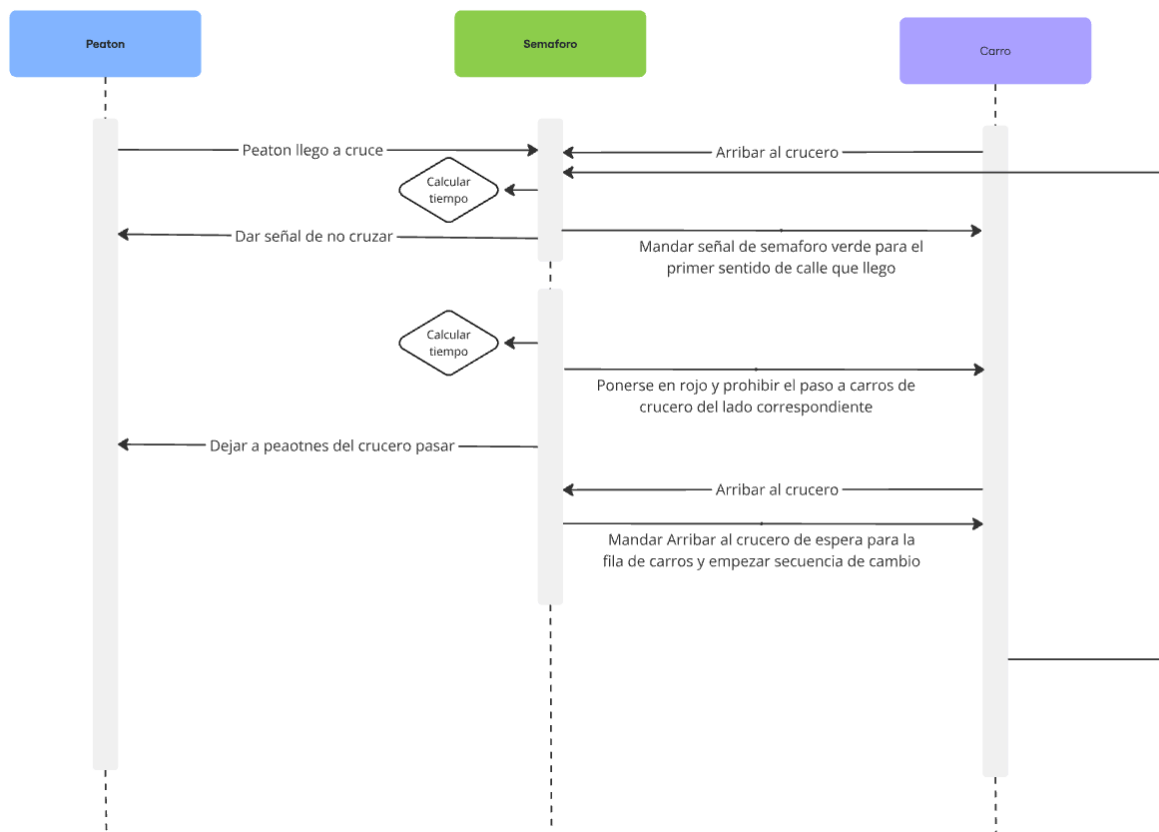
Fuimos capaces de observar que Q-learning puede llegar a ser una mejor solución y aquella que implementaremos por necesitar de experimentación o entrenamiento para funcionar de manera adecuada.

7.5 MODELO DE LA NEGOCIACIÓN

La manera en la que se negocia primero que nada es mediante el semáforo, él es el mediador entre coches y personas. Claro esto no es una negociación directa sino más bien una forma de controlar que el tráfico sea adecuado. Fuera de eso, los coches no necesitan negociar puesto que el tráfico lo regulan los semáforos y los sentidos de las calles. Para las personas implementaremos una política donde la persona que está más cercana a su destino será la que obtendrá el paso y aquel que no, se moverá de su camino si estos dos chocan, si no puede moverse, se intentará que él tenía preferencia se mueva, de lo contrario se esperará a que se abra paso de los agentes adyacentes por 3 turnos y si no se han movido aún se iniciará una operación recursiva con las personas que lo rodean para poder lograr que estos se abran espacio para el paso.



7.6 MODELO DE LA INTERACCIÓN



El orden de interacciones se basa en 2 tipos de estados por cruce que todos serán regulados por el semáforo o sistema de semáforos en medio. El primer estado explica lo que pasa cuando llega un carro al cruce y nadie más tiene prioridad. En este caso el semáforo calcula cuántos mensajes de carros llegaron para calcular cuánto durará y se pone en verde y manda un mensaje para que los carros que van llegando puedan pasar. A la par en el caso de

que se mande un mensaje de un peatón que quiera cruzar se le prohíbe y se manda de regreso una respuesta de que se espera al siguiente estado. En sistemas complejos de semáforos (aún no hemos contemplado si los vamos a incluir) los mismos semáforos se estarían mandando mensajes entre ellos también para estar sincronizados que solo uno esté en verde y los demás en rojo para no crear tráfico. El atributo de color cambiaria a amarillo pero no se realizarían interacciones entonces no aparece en el protocolo; simplemente indica que se está acabando el tiempo calculado. Después de esto, se pasa al segundo estado donde se envía a los peatones que el cruce peatonal se puede cruzar. Así mismo, el semáforo estará en color rojo, por lo que si un carro llega a ese cruce, envía mensaje de que llegó y recibe la respuesta que se espera hasta que agote el tiempo y de nuevo envía mensaje de regreso al estado verde pidiendo pasar.

8. DESCRIPCIÓN DEL MODELADO GRÁFICO

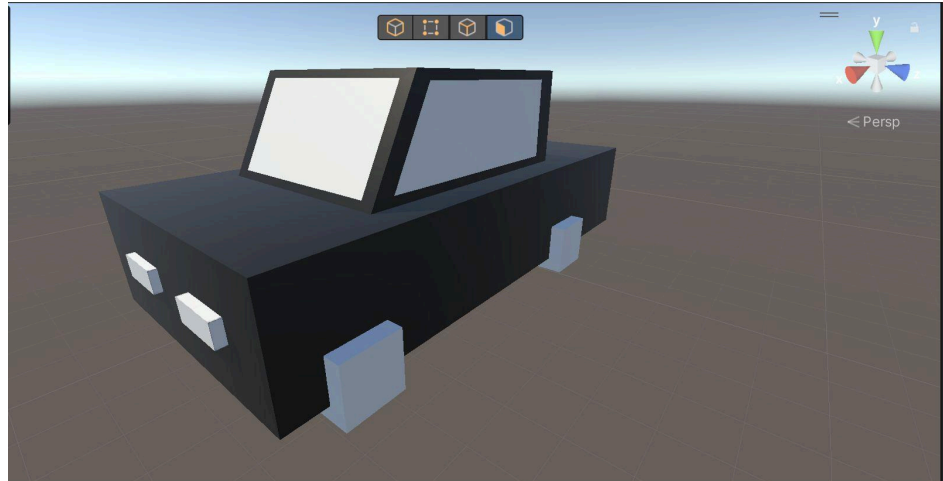
8.1 ESCENA A MODELAR

Aquí es necesario agregar de inicio un *draft* de la escena y explicarla. Posteriormente cuando se tenga diseñada en Unity agregar una imagen del resultado final. Una especie de expectativa vs. realidad.

8.2 COMPONENTES GRÁFICOS

Nombre	Carro
Descripción	Automóvil en 3D que funcionará para representar al agente Vehículo, fue creado con pro-builder

Imagen



Fuente

Pedro Sánchez

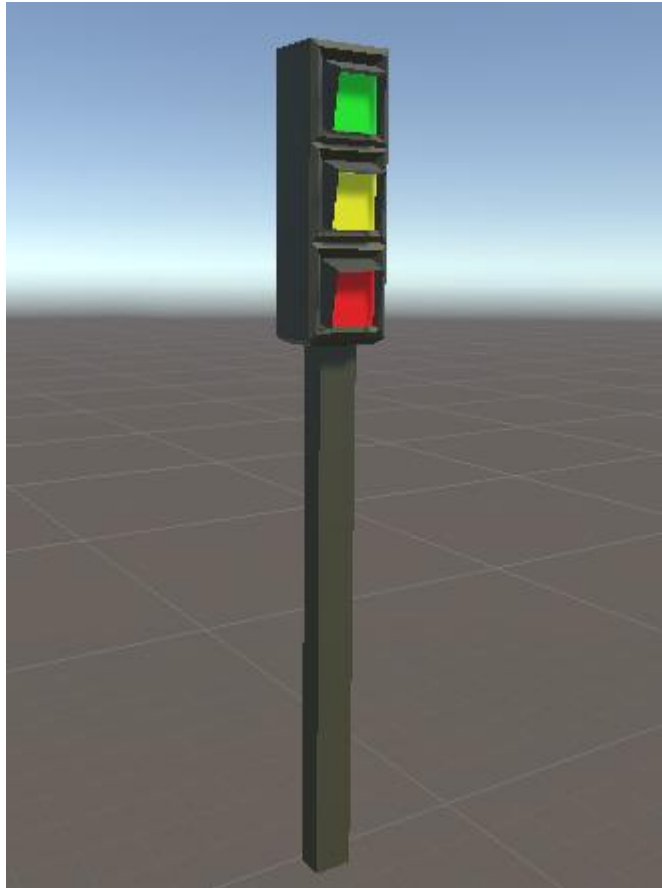
Nombre

Semáforo

Descripción

Semáforo en 3D que funcionará para representar al agente Semáforo, utilizando pro-builder.

Imagen



Fuente

Daniel Rubies

8.3 PREFABS

Nombre

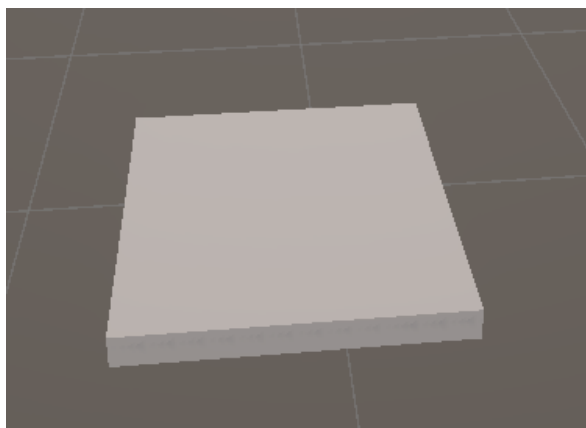
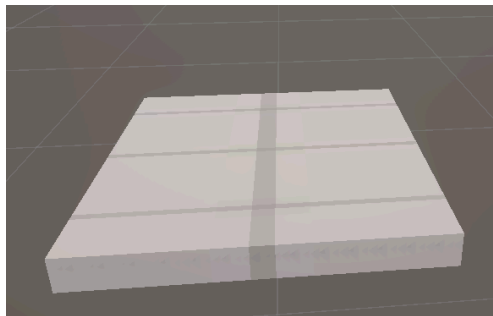
Generador de calles

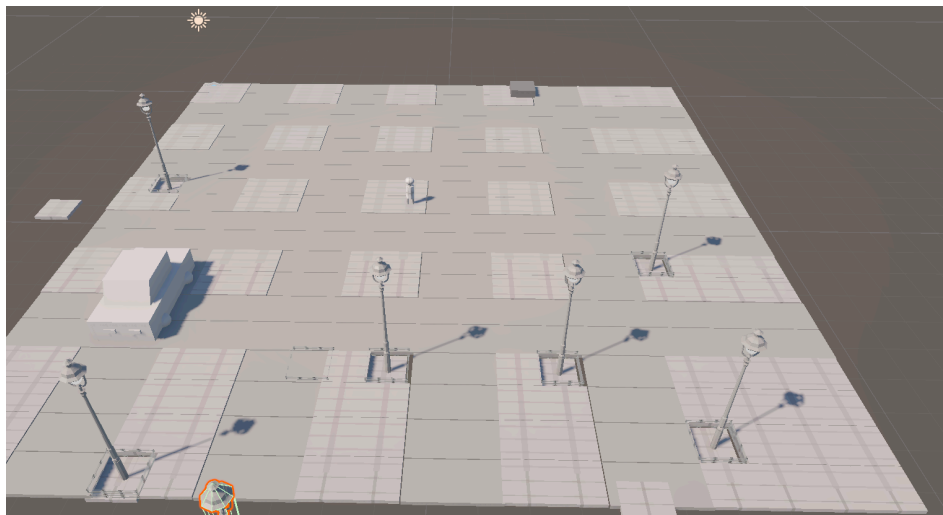
Descripción

Este prefab se encarga de generar calles con base en los prefabs de **persona**, **poste**, **banqueta y calle**. Usa el script que se detalla debajo para generar una calle que después consumirá datos de nuestra simulación en agentPy a través de TCP para ser capaz de hacer render de cualquier simulación generada en

Python. Un ejemplo básico del comportamiento se puede ver en el siguiente vídeo. [aquí](#)

Imagen





Scripts


Scripts que utiliza el prefab

8.4 SCRIPTS

Nombre	Generate simulation
Descripción	<p>Este script genera un entorno urbano en Unity utilizando una matriz 2D (floorMatrix) para definir calles, banquetas y obstáculos. Crea automáticamente los elementos del suelo (con los prefab del punto anterior), coloca obstáculos en las banquetas, y genera un personaje caminante (con otro prefab) que se mueve aleatoriamente por las celdas caminables. La posición del caminante se calcula en base a las celdas disponibles y se ajusta con un pequeño offset para que parezca que está sobre el suelo. También se incluye lógica para posicionamiento preciso.</p> <p>Se está trabajando para obtener la matriz de la ciudad (floorMatrix) y la posición del caminante desde un WebSocket, en lugar de generarlas de forma aleatoria, lo que permite reflejar datos externos en tiempo real en el entorno, por lo tanto toda la información será recibida a través de lo que se va creando en la simulación de agentPy.</p>
Interacciones	Este script interactúa con los prefabs detallados anteriormente a su vez que a futuro con los multiagentes de AgentPy.

9. ENTREGABLES DE ADMINISTRACIÓN DE PROYECTO

Repositorio: [santigugon/reto-multiagentes](https://github.com/santigugon/reto-multiagentes)

Q-Learning :  qlearnign_reto_final.ipynb

Unity:

https://1drv.ms/u/c/a283235c35be309e/EQe0YXDyy8RGur3il59_SclB-ko-LgOFugBHaE3-CFkpeg?e=3NBAB9

10. REFERENCIAS

Krajewicz, D., et al. (2012). "Recent Development and Applications of SUMO – Simulation of Urban MObility."

Balmer, M., et al. (2008). "MATSim-T: A Framework for Integrating Activity-Based Models and Multiagent Simulations."

Batty, M. (2013). "The New Science of Cities."