

# Modelación de sistemas multiagentes con gráficas computacionales (Gpo 523)

M5. Revisión de avance 1

Profesores:

Sergio Ruiz Loza

David Christopher Balderas Silva

17 de noviembre del 2021

# Conformación del equipo

## **Diego Rodriguez**

Fortalezas: Responsabilidad, resiliencia, liderazgo

Áreas de oportunidad: No hay un buen conocimiento concreto de la mayoría de cosas que se utilizarán en el curso.

Espero con este curso retomar varios conocimientos que ya había olvidado debido a que los vimos hace varios semestres y poder adquirir este nuevo conocimientos que suena muy interesante que podría llegar a servirme mucho en mi futuro profesional.

## **Manuel Hernandez**

Fortalezas: Conocimiento amplio en Python, curiosidad, trabajo en equipo

Áreas de oportunidad: Administración del tiempo, puntualidad, poco conocimiento de C# y Unity

## **Santiago Orozco Quintero**

Fortalezas: Buena capacidad de análisis, puntualidad y responsabilidad

Áreas de oportunidad: Falta de conocimiento sobre el uso de agentes y las representaciones gráficas en python

Para este bloque espero llevarme una mejor comprensión de la programación de agentes dinámicos así como también de la representación de información mediante medios gráficos como imágenes, tablas o gráficas.

Como equipo esperamos concluir de manera satisfactoria el reto cumpliendo con lo que se nos pide, sabemos que esto va de la mano con comprender los temas que se imparten por lo que también deseamos lograr mejorar nuestras habilidades como programadores añadiendo a nuestro catálogo de capacidades el mostrar información de manera más visual mediante la implementación de gráficos computacionales.

Para lograr esto nos comprometemos a asistir puntualmente a todas las sesiones así como también a dedicar tiempo después de clases al progreso de las actividades necesarias a realizar y por ende a solicitar asesorías cuando sea necesario.

# Creación de herramientas de trabajo colaborativo

Para lograr un resultado satisfactorio emplearemos las siguientes herramientas de comunicación:

Whatsapp: Para mantenernos en contacto constante durante el trabajo

Correo electrónico y Slack: Para estar en contacto con los profesores

Zoom: Por si en algún momento requerimos reunirnos en persona, así como también para tomar asesorías y las clases

Herramientas de Google Drive: Para realizar la documentación pertinente del proyecto

GitHub: Para tener un repositorio de los códigos que se vayan trabajando (<https://github.com/MovilidadUrbana>)

# Propuesta del reto

## Ruta menos congestionada

Para nuestro reto resolveremos el problema de hacer que los autos tomen una ruta menos congestionada, toda persona que lleve un tiempo conduciendo en una ciudad grande como lo es la capital sabe lo molesto y tedioso que puede ser el tráfico, es por ello que desarrollaremos un sistema que modele la forma en como muchos conductores suelen afrontar esta problemática que es la de buscar la ruta con mayor flujo vehicular aunque en ocasiones sea más larga.

Esta estrategia como ya dijimos es la que muchas personas implementan ya que en muchas ocasiones una ruta más larga pero donde los autos avanzan a un ritmo constante suele ser más eficiente que una ruta corta pero que no tiene mucha movilidad.

Los agentes involucrados para esta propuesta en particular son los autos y los semáforos, esto debido a que es en referencia a la cantidad de estos en un determinado lugar que nuestro vehículo decidirá si toma o no esta ruta, de igual manera los semáforos juegan un papel importante en esto ya que detienen la circulación si se encuentran en rojo, por lo que nuestros autos podrían también intentar evitar calles con semáforos para reducir la posibilidad de terminar en un embotellamiento otro objeto involucrado en este reto que no puede ser considerado como un agente en sí debido a su naturaleza estática y pasiva son las calles ya que este será el terreno por el cual se desplazaran los vehículos y que por ende les marca la pauta para poder realizar su función primaria.

La interacción entre los factores involucrados se podría definir de la siguiente manera, los autos van a viajar sobre la calle “observando” a otros autos, cuando ellos noten que en un determinado camino hay demasiados vehículos los agentes, representados en este caso por los vehículos, tomarán una calle alterna priorizando el seguir avanzando aunque la nueva ruta requiere recorrer una mayor distancia, esto debido a que partimos del supuesto de que una persona considera mejor el transitar por una vía sin congestión aunque la distancia que lleve recorrerla sea mayor, los semáforos por su parte detienen el paso de los vehículos cuando se encuentran en rojo y la permiten continuar cuando está en verde, como los semáforos están ligados de cierta forma a los embotellamientos, los automóviles podrían también elegir calles sin semáforos como posibles rutas para evitar el tráfico, finalmente la calle forma parte del ambiente por el cual los autos circulan, estas mismas tendrán la característica de definir en qué sentido pueden transitar los vehículos (esto para delimitar la movilidad de los agentes y no permitir que circulen como quieren y que entren en sentido contrario a una calle).

Diagrama de clase presentando los distintos agentes involucrados.

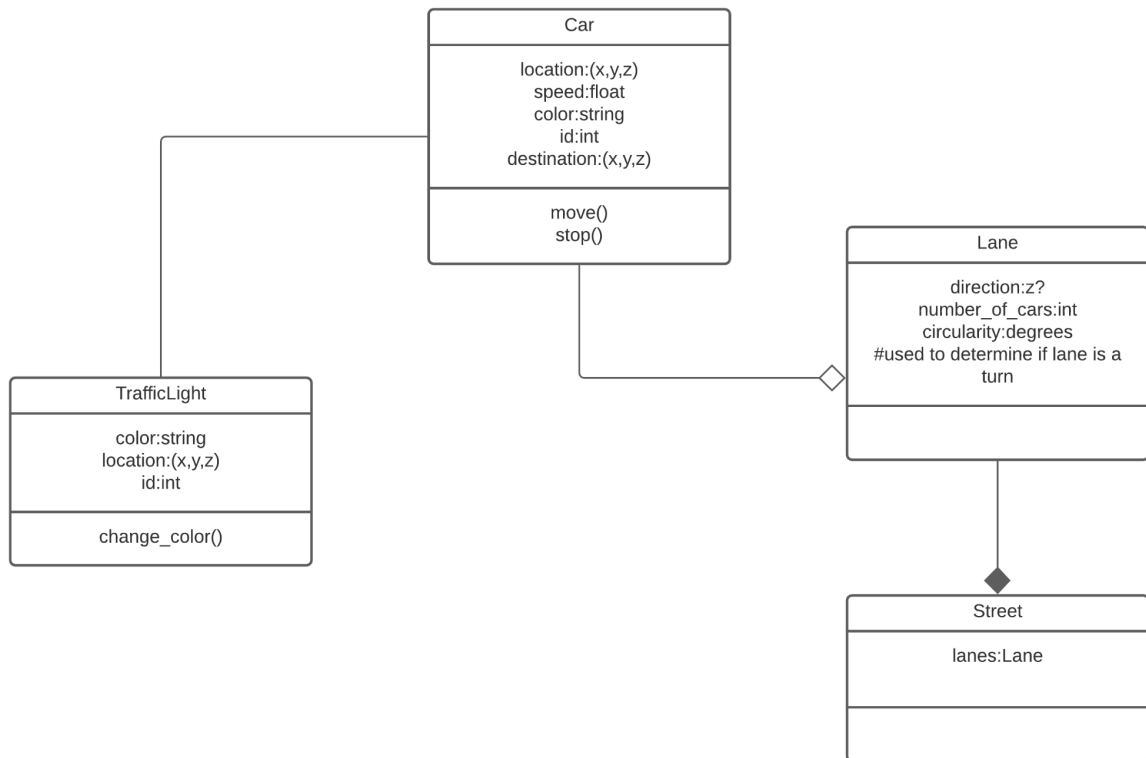
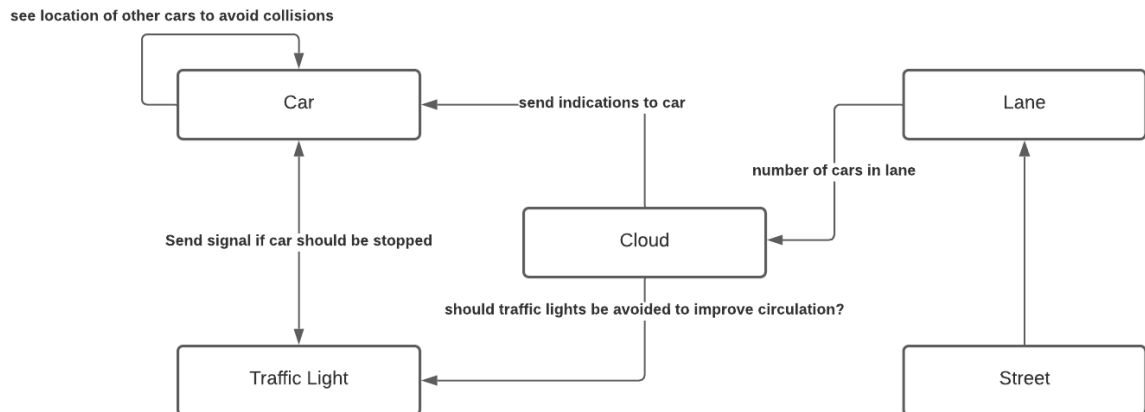


Diagrama de protocolos de interacción.

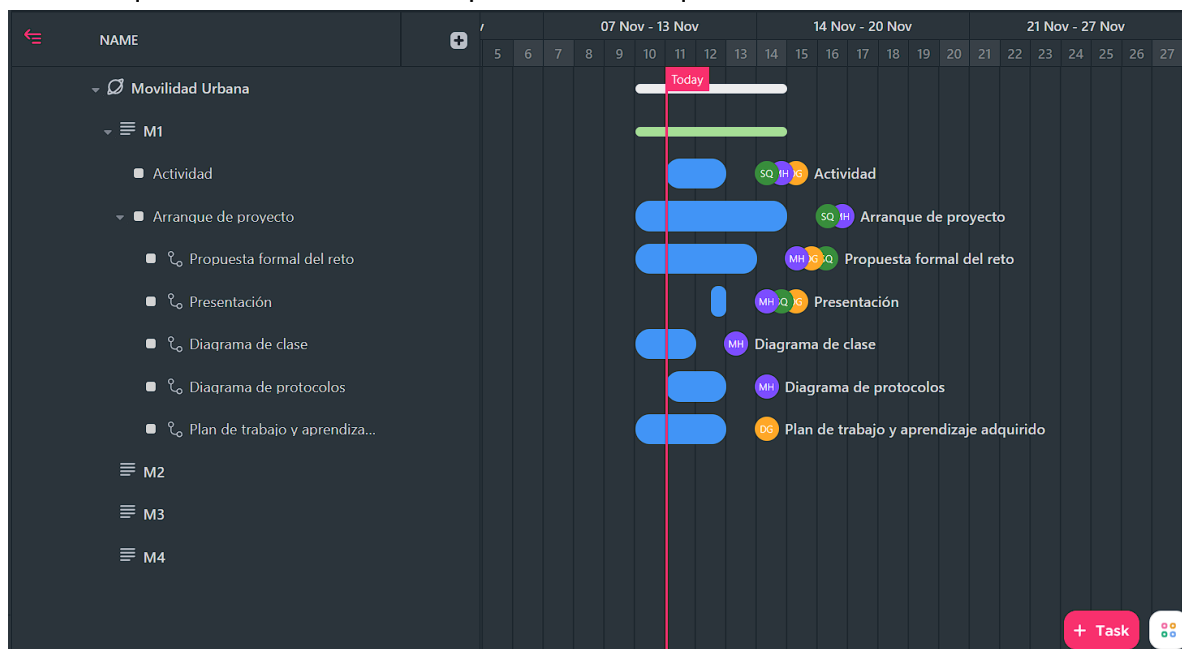


# Plan de trabajo y aprendizaje adquirido

La organización de nuestro trabajo se realizará por medio de la herramienta en línea llamada click up donde distribuimos los trabajos y los tiempos de una mejor forma ya que cada avance queda registrado aquí y se le mandan correos como recordatorios a los encargados de cada trabajo para que no hay problema de olvidos, igualmente la herramienta se puede conectar con github para tener una mejor administración de nuestro trabajo (<https://app.clickup.com/14180406/v/I/6-110349746-1>).


Esta primera etapa del trabajo consiste primero en crear la documentación pertinente para el reto, esto va desde plantear el proyecto hasta realizar los diagramas necesarios, posteriormente se crearán los diversos objetos que se usarán durante el reto y finalmente se les dará la funcionalidad requerida, así mismo cada semana se actualizará este plan de trabajo para incluir toda nueva información y avances conseguidos.

Como equipo obtuvimos aprendizajes sobre el uso de ciertos aspectos de diseño en Unity y sobre los agentes en Python, si bien de ambas cosas aun nos hace falta profundizar más creemos que es un buen comienzo para realizar lo que se nos solicita.



# Cuenta en IBM

Otro avance que aunque menor es importante fue la creación de las cuentas institucionales para utilizar los servicios de IBM ya que los servicios que ofrece son vitales para poder realizar el reto de manera satisfactoria



IBM Academic Initiative


Topics ▾

Usage terms ▾

Additional Resources

FAQ


Search




a01658308@te...

Logout


Most popular topics covered




AI




Capstone




Data Science




IBM Automation




IBM Cloud




Healthcare



Quantum



Red Hat Academy

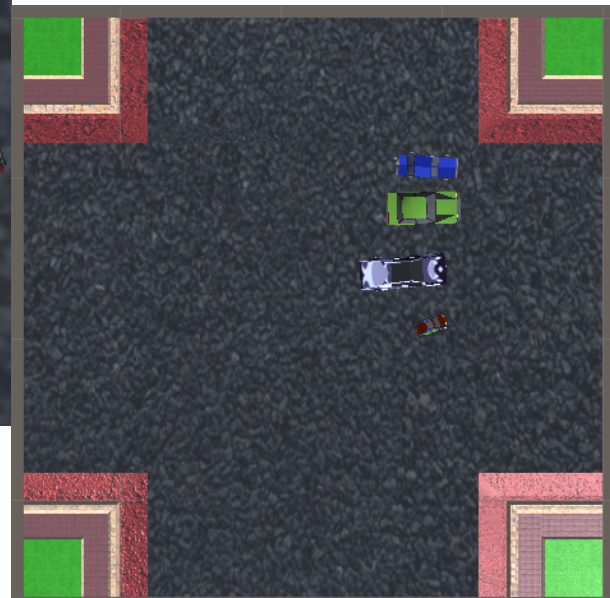


View All

## Modelos 3d

Para comenzar decidimos realizar un par de modelos 3d en Unity con el fin de avanzar el aspecto estético del reto, para ello cada integrante creó sus propuestas de vehiculos para usar, estos vehículos fueron pintados y se les agregaron texturas en ciertas zonas para hacerlos más vistosos, de igual manera se diseñó un modelo preliminar de la calle.

Era importante tener un par de propuestas diferentes para poder así elegir la que encaja mejor, a su vez las propuestas que no sean elegidas pueden después incorporarse como vehículos aparte, lo relevante de esto fue a su vez practicar con las distintas herramientas de Unity para dar textura, color y forma a los agentes.



## Avance del código

```
1  from mesa import Agent, Model, time
2  import math
3  import random
4
5
6  > class Car(Agent): ...
60
61
62  > class TrafficLight(Agent): ...
61
62
63  > class TrafficModel(Model): ...
65
66
67  > class RandomActivation(Model): ...
60
61
62  > class RandomActivationSchedule(object): ...
60
61
62  def main():
63      model = TrafficModel(100, 100, 1)
64      model.generate_model(100, 100, 1, 1, 1)
65      model.step()
```

Resumen de clases, código completo disponible en el Github del equipo.