

M3. Avance de Proyecto 1

Modelación de sistemas multiagentes con gráficas computacionales

Materia TC2008B

Integrantes del equipo

Martín Palomares García A01066569

Ana Fernanda Hernández Tovar A01411484

Brian Alberto Salomón Sevilla A00828826

Carlos G. del Rosal A01566719

Profesores

Luis Andres Castillo Hernández

Jorge Mario Cruz Duarte

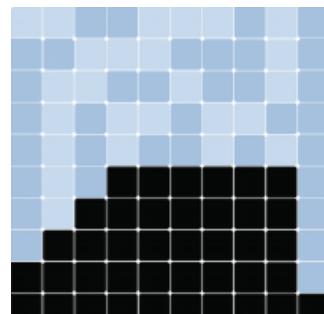
Fecha de entrega



17 de Noviembre de 2021

Introducción

Para este arranque de proyecto, empezamos con lo básico, que fueron los conceptos básicos y conocimiento de las funciones que maneja el lenguaje de programación Python, además de conocer también “Mesa”, el “ProBuilder” de Unity, entre otras cosas, y conociendo todas estas plataformas y herramientas, podemos hacer una buena modelación y simulación de sistemas multiagentes. Y ahora, en este documento, les mostraremos cómo fuimos desarrollando la primera etapa del proyecto del bloque de “**Modelación de sistemas multiagentes con gráficas computacionales**” .



Mesa Logo

Avances



Para el primer avance de este reto, hemos trabajado en 2 diferentes plataformas que se nos fueron explicadas en las sesiones anteriores del bloque para desarrollar el proyecto. A continuación mencionaremos las plataformas que usamos para este avance y que fue lo que elaboramos en cada una de ellas.

Unity



En Unity aprendimos a usar las siguientes herramientas para el desarrollo del modelo principal del proyecto:



ProBuilder: Esta herramienta es un híbrido único de modelado 3D y herramientas de diseño en niveles, optimizado para crear geometría simple pero capaz de lograr edición detallada y desenvoltura de UV también.

Con esta herramienta, aprendimos a crear rápidamente prototipos de estructuras, características de una calle, vehículos, etc. Y en este proyecto, usamos el **probuilder** para hacer los modelos de los automóviles y el modelo de la intersección de los automóviles.

También aprendimos a trabajar con los siguientes elementos:



Materiales: Para que los elementos tengan un color más opaco o más brillante dependiendo a las configuraciones que se les de.



Texturas: Para que se pueda ver con más realismo un elemento, haciéndola más creíble y lograr sensaciones de espacios.



Iluminación: Sirve para que la luz pueda disminuirse con la distancia desde la fuente y así le pueda dar un efecto más realista al ambiente de la situación.

A continuación se muestra una imagen del preview del modelo de Unity. Donde muestra al menos 1 de cada elemento mencionado anteriormente. →



Imagen 1.1 Imagen del Preview del modelo de la intersección de los automóviles y de la calle

Python

En el área de python, aprendimos a utilizar Google Colab, para trabajar en los programas desarrollados en lenguaje de programación Python, y así trabajar en el funcionamiento y la simulación del reto establecido. Ahora mencionaré lo que tuvimos que instalar y las librerías que tuvimos que importar para que se lograra el funcionamiento de la simulación.



Mesa: Es un marco modular para construir, analizar y visualizar modelos basados en agentes y este es muy adecuado para estudiar la cooperación y competencia de varios sujetos de simulación de los modelos de los agentes.



from mesa import Agent, Model: La clase `Model` se hace cargo de los atributos a nivel del modelo, maneja los agentes. Cada modelo puede contener múltiples agentes y todos ellos son instancias de la clase `Agent`.



from mesa.space import MultiGrid: Debido a que necesitamos un solo agente por celda elegimos `SingleGrid` que fuerza un solo objeto por celda.



from mesa.time import SimultaneousActivation: Con `SimultaneousActivation` hacemos que todos los agentes se activen de manera simultánea.



from mesa.datacollection import DataCollector: Vamos a hacer uso de `DataCollector` para obtener el grid completo cada paso (o generación) y lo usaremos para graficarlo.

Matplotlib lo usamos para graficar/visualizar cómo evoluciona el autómata celular:

- `%matplotlib inline`
- `import matplotlib`
- `import matplotlib.pyplot as plt`
- `import matplotlib.animation as animation`
- `plt.rcParams["animation.html"] = "jshtml"`
- `matplotlib.rcParams['animation.embed_limit'] = 2**128`

Definimos los siguientes paquetes para manejar valores numéricos.

- `import numpy as np`
- `import pandas as pd`

- `import random`

Y en el desarrollo del código en python, se llegó a esta simulación en python (El Jupyter Notebook se encuentra en el repositorio de Github). [→](#)

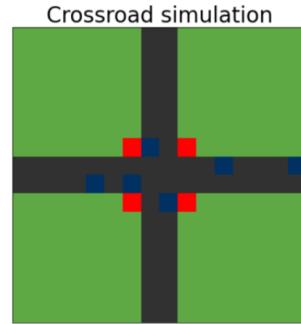


Imagen 2.1 Captura de una simulación obtenida en Colab

Diagramas

Y los diagramas que desarrollamos para entender el funcionamiento y la creación del proyecto en simulación, son los siguientes:

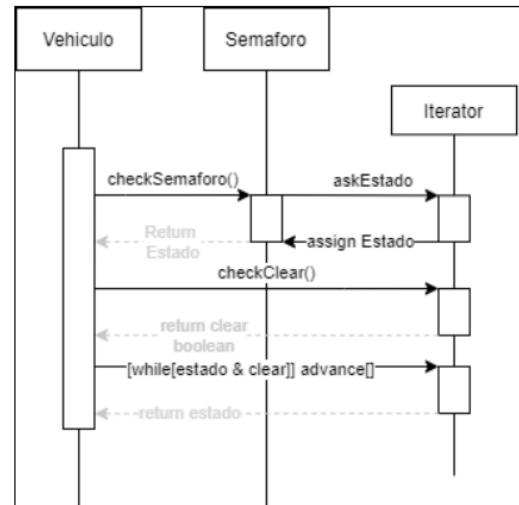
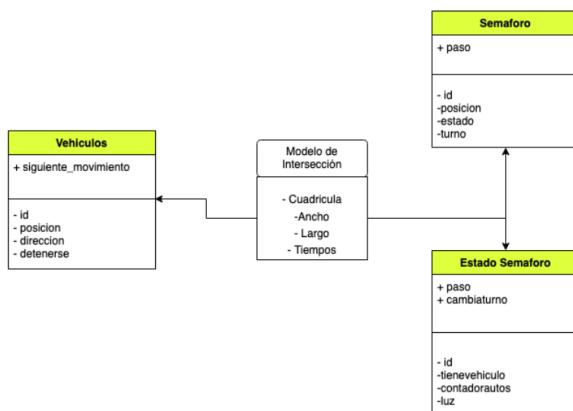


Imagen 1.2 Diagrama de Iteraciones Sequencial.

Extras

Liga de Repositorios

En este apartado se encuentran los links de los repositorios que hemos creado, donde encontrarán cada trabajo y/o avance que hemos realizado en equipo.

Slack

Slack

 https://join.slack.com/t/itesm-zfs6300/shared_invite/zt-yekqz1xf-YB6Vvd8T_5uPp0oohGWOvQ

GitHub - MartinPaGarcia/Proyecto_Sistemas_Inteligentes  PaGar
to_Sis

Martín Palomares García A01066569 Brian Alberto Salomón
Sevilla A00828826 Ana Fernanda Hernández Tovar A01411484
Carlos González del Rosal A01566719 La comunicación por
gente

 0 Issues

Conclusión

Ahora como primera fase del proyecto, sólo hemos avanzado hasta lo básico, en un modelado 3D en Unity y en entender el funcionamiento de un semáforo en una intersección con una N cantidad de automóviles. Aun nos falta conocer mucho si queremos llegar a una etapa final en este proyecto, pero al menos conocemos las bases que necesitamos para ya empezar con este proyecto e irlo desarrollando poco a poco.