

Avance Semanal

Equipo 3

A01656937 Adriana Inés López Sánchez

A01369117 Lidia Paola Díaz Ramírez

A01251505 Sotero Osuna Gómez

Principales Avances

Esto fue lo trabajado durante la semana:

- Diagramas de clase y protocolos de interacción actualizados
- Código de implementación de los agentes
- Código de la implementación de la parte gráfica de la solución

Diagrama de clases

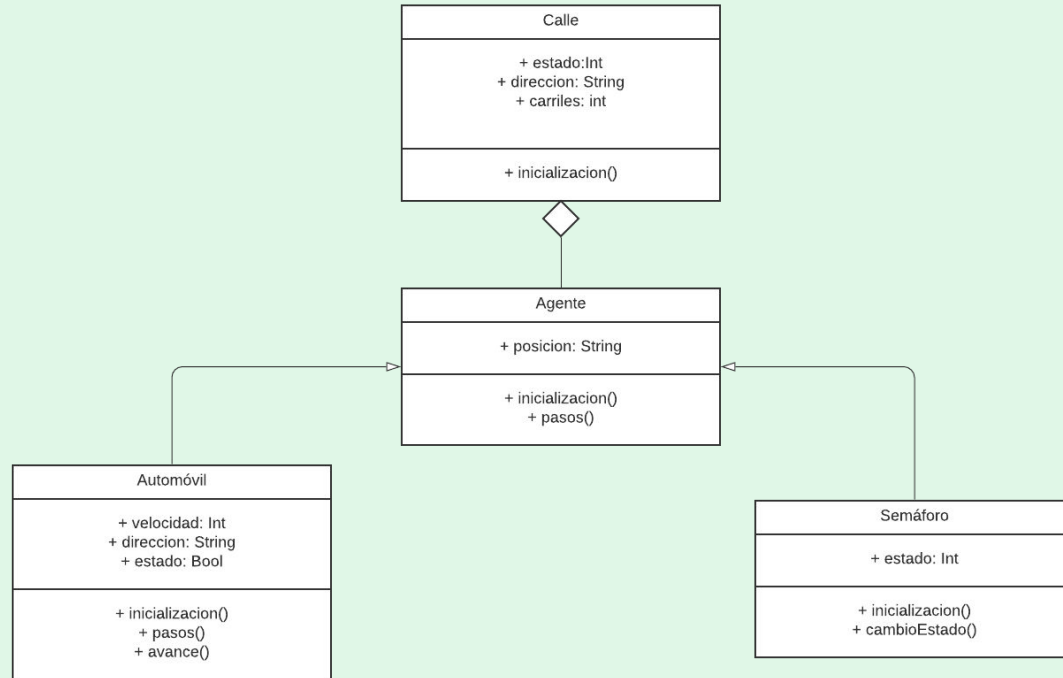
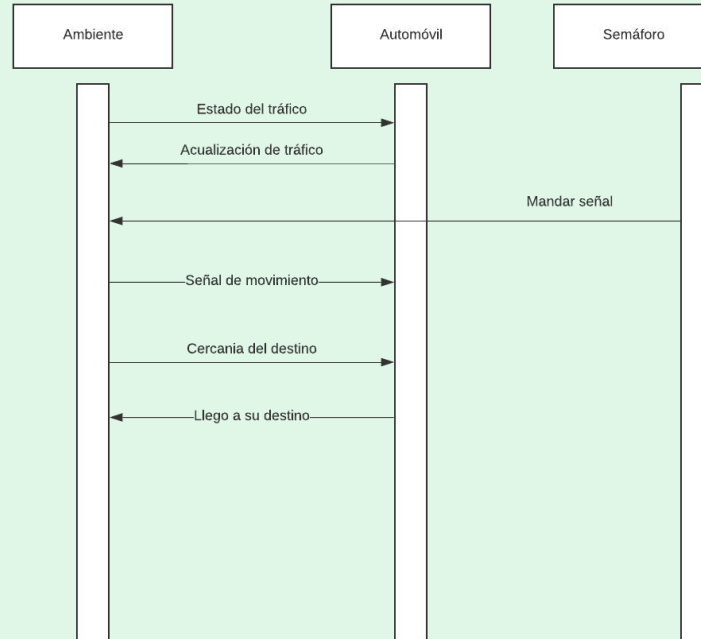


Diagrama de protocolos de interacción



Python

```
agentes.py
# TC20008, Sistemas Multiagentes y Gráficas Computacionales
from Mesa import Agent, Model
import pandas as pd
import numpy as np

class Coche(Agent):
    def __init__(self, id_unico, modelo):
        super().__init__(id_unico, modelo)
        self.id_unico=id_unico
        self.nueva_posicion = None
        self.movimientos = 0
        self.sentidoCOCHE=None

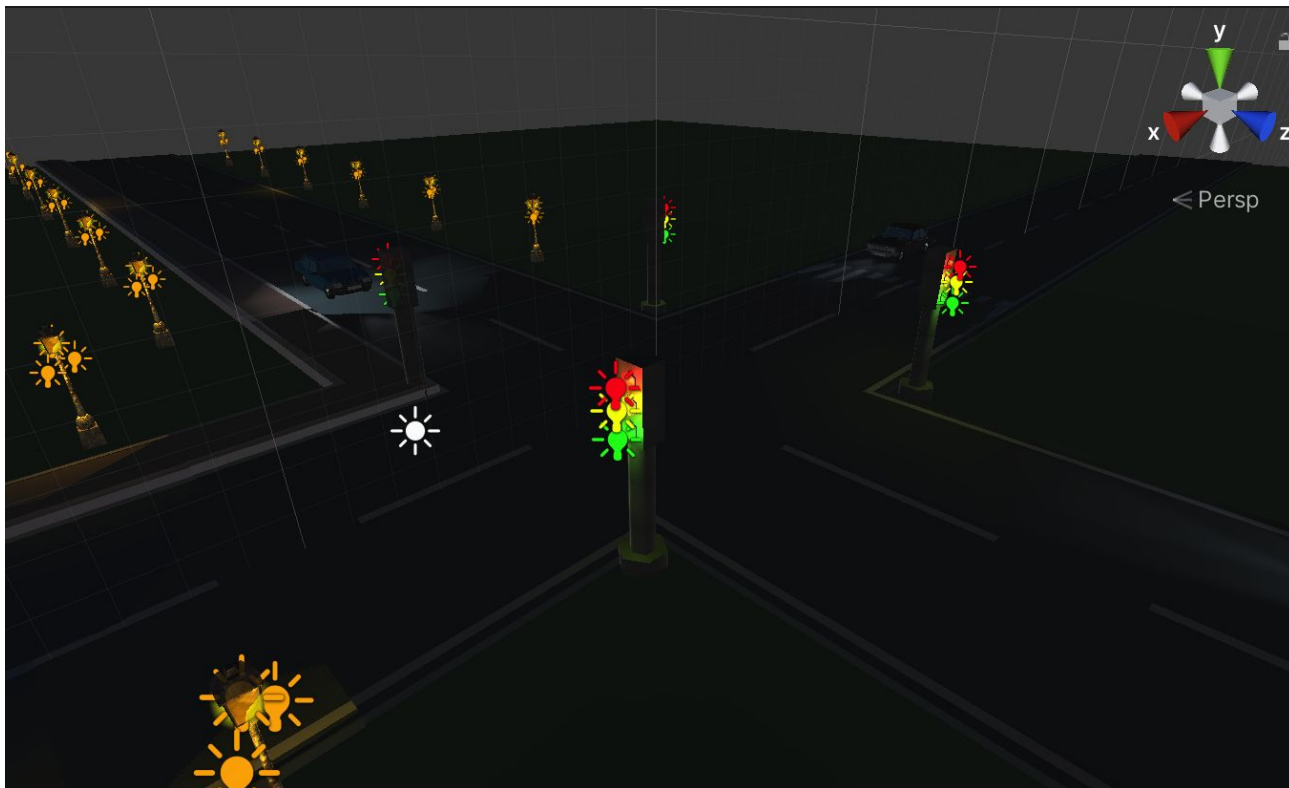
    def step(self):
        global randomDireccion
        global auxSwitch

        vecinos = self.model.grid.get_neighbors(
            self.pos,
            moore=False,
            include_center=True)
        for vecino in vecinos:
            if isinstance(vecino, Coche) and self.sentidoCOCHE==vecino.sentidoCOCHE:
                self.nueva_posicion=self.pos
                for vecino in vecinos:
                    if isinstance(vecino, Semaforo):
                        if (vecino.estado==vecino.ROJO or vecino.estado==vecino.AMARILLO) and vecino.sentidoSEMAFORO==self.sentidoCOCHE:
                            self.nueva_posicion=self.pos
                            break
                        if vecino.estado==vecino.VERDE and vecino.sentidoSEMAFORO==self.sentidoCOCHE:
                            randomDireccion=random.randint(1,2)
                            vecinos2 = self.model.grid.get_neighbors(
                                self.pos,
                                moore=False,
                                include_center=True)
                            for vecino2 in vecinos2:
                                if isinstance(vecino2, Piso) and vecino2.pos==self.pos:
                                    vecino2.siguiente_estado = vecino2.estado
                                    if vecino2.siguiente_estado==vecino2.CALLE:
                                        #Se moverá del semáforo dependiendo que sentido tiene la calle
                                        if vecino2.SENTIDO_CALLE==vecino2.SENTIDO_Este:
                                            self.nueva_posicion= tuple(sum(x) for x in zip(self.pos,(0,1)))
                                            self.sentidoCOCHE="E"
                                            break
                                        if vecino2.SENTIDO_CALLE==vecino2.SENTIDO_Oeste:
                                            self.nueva_posicion= tuple(sum(x) for x in zip(self.pos,(0,-1)))
                                            self.sentidoCOCHE="O"
                                            break
                                        if vecino2.SENTIDO_CALLE==vecino2.SENTIDO_Norte:
                                            self.nueva_posicion= tuple(sum(x) for x in zip(self.pos,(1,0)))
                                            self.sentidoCOCHE="N"
                                            break
                                        if vecino2.SENTIDO_CALLE==vecino2.SENTIDO_Sur:
```

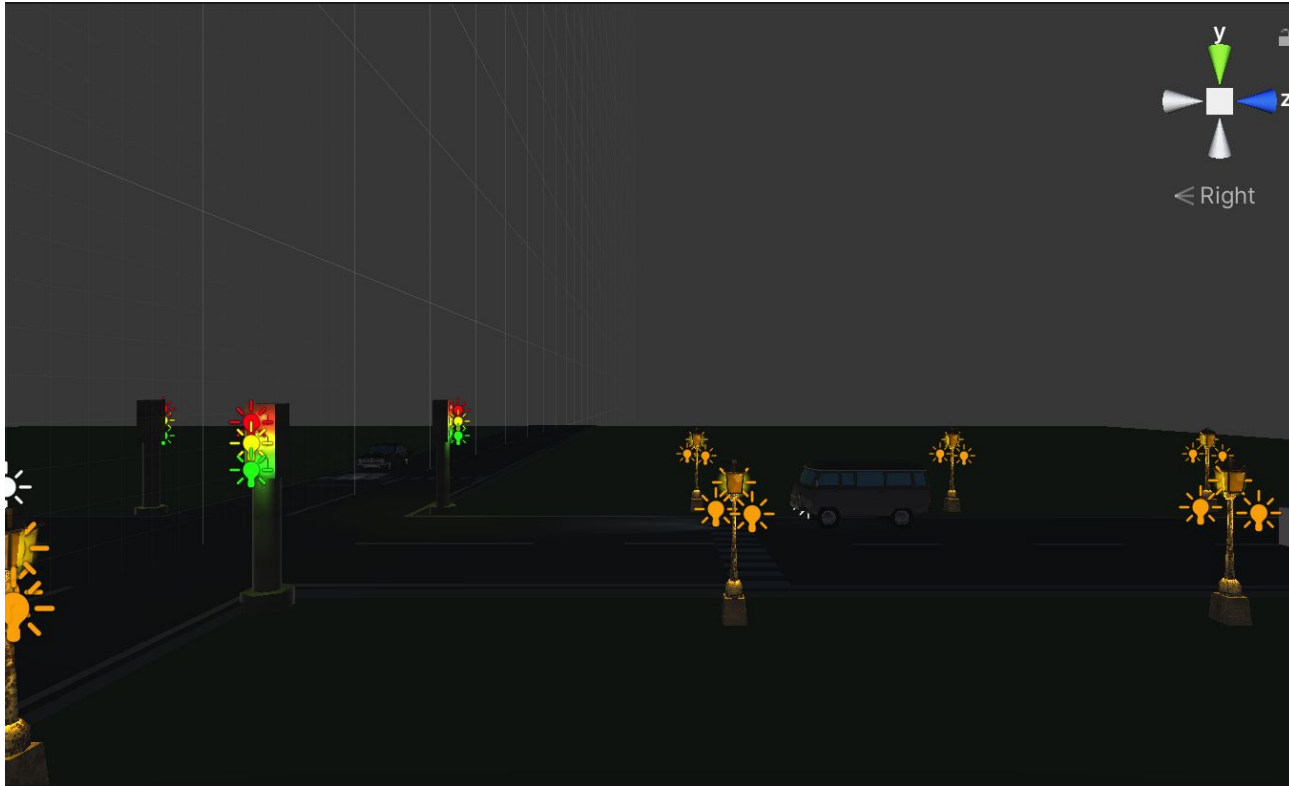
Python

```
81
82
83 class Semaforo(Agent):
84     ROJO=2
85     AMARILLO=3
86     VERDE=4
87
88     def __init__(self,pos,modelo,estadoS=ROJO):
89         super().__init__(pos,modelo)
90         self.x,self.y=pos
91         self.estadoS=estadoS
92         self.siguienteEstadoS=None
93         self.sentidoSEMAFORO=None
94
95     def step(self):
96         global switchSemaforo
97
98         vecinos = self.model.grid.get_neighbors(
99             self.pos,
100             moore=True,
101             include_center=False)
102         for vecinoSem in vecinos:
103             if isinstance(vecinoSem,Coche) and self.sentidoSEMAFORO==vecinoSem.sentidoCOCHE and not switchSemaforo:
104                 auxSwitch=False
105                 switchSemaforo=True
106                 self.siguienteEstadoS=self.VERDE
107                 break
108             else:
109                 vecinosR = self.model.grid.get_neighbors(
110                     self.pos,
111                     moore=True,
112                     include_center=False)
113                 for vecinoR in vecinosR:
114                     if not isinstance(vecinoR,Coche) and not switchSemaforo:
115                         self.siguienteEstadoS=self.AMARILLO
116
117                 else:
118                     self.siguienteEstadoS=self.ROJO
119
120     def advance(self):
121         self.estadoS=self.siguienteEstadoS
122
123
```

Unity



Unity



Unity



Plan de trabajo: Semana 1 y 2

Tarea	Inicio	Fin	Tiempo	Quien realiza	Porcentaje	Diferencia de estimación
Crear carpeta en Google Drive para ir almacenando los trabajos del equipo	04/11/21	4/11/21	15 min	Todos los miembros del equipo	100%	0%
Crear repositorio en Github	04/11/21	4/11/21	15 min	Paola	100%	0%
Crear cuentas de IBM Cloud	05/11/21	10/11/21	20 min	Todos los miembros del equipo	100%	0%
Finalizar entrega de arranque de proyecto	08/11/21	11/11/21	2 h	Todos los miembros del equipo	100%	0%
Crear proyecto y subirlo a Github	08/11/21	11/11/21	1 h	Todos los miembros del equipo	100%	100%

Plan de trabajo: Semana 3

Tarea	Inicio	Fin	Tiempo	Quien realiza	Porcentaje
Actualizar diagramas de clase y protocolos de interacción	14/11/21	13/11/21	30 min	Sotero	100%
Crear código de agentes (semaforos y coches)	15/11/21	11/21	3 h	Paola	40%
Crear implementación gráfica (unity)	16/11/21	18/11/21	3 h	Adriana	40%
Finalizar entrega de semana 3 del proyecto	15/11/21	19/11/21	2 h	Todos los miembros del equipo	100%
Subir actualización a GitHub	19/11/21	19/11/21	15 min	Todos los miembros del equipo	100%

Actividades planeadas

Tarea	Inicio	Fin	Tiempo	Quien realiza	Porcentaje
Seguir desarrollando el entorno en Unity	20/11/21	-	-	Todos los miembros del equipo	0%
Crear conexión de IBM Cloud a Unity	20/11/21	-	-	Todos los miembros del equipo	0%
Desarrollar el ambiente en Python	20/11/21	-	-	Sotero	0%
Continuar desarrollo de código en python	20/11/21	-	-	Todos los miembros del equipo	0%

Aprendizajes adquiridos

Para esta semana los aprendizajes que más nos llevamos fue respecto a unity y python, ya que estuvimos trabajando más en estas áreas.

Aprendimos acerca del modelado, la iluminación, Frustum y la comunicación entre la nube y python, los cuales son conocimientos que nos sirven para lograr llegar a una solución del reto