Avance semanal

Equipo 1

Diego Enrique Jiménez Urgell	A01652617
Agustín Abreu Callejas	A01653126
Yusdivia Molina Román	A01653120

Plan de trabajo semana 3

Etapa 1.2	11/15	11/19	7	4	21	Todos	85%
Diseño del sistema 3d (calles)	11/15	11/17	7	2	13	Agustín y Yus	90%
Simulación en Unity	11/16	11/18	8	4	13	Diego y Agustín	20%
Simulación de coches en línea recta	11/17	11/19	9	2	8	Todos	100%
Simulación coches dando la vuelta	11/17	11/19	9	2	8	Agustín	100%
Simulación de semáforos en funcionamiento	11/15	11/18	7	3	8	Yus y Diego	50%
Diseño de un coche por nosotros	11/17	11/19	9	2	13	Diego	100%
Entregable .py	11/17	11/19	9	2	8	Todos	40%
Definición clases con atributos y métodos	11/18	11/19	10	1	8	Yus	40%

Aprendizaje obtenidos: Gráficas computacionales

Radar VFC -> Optimización

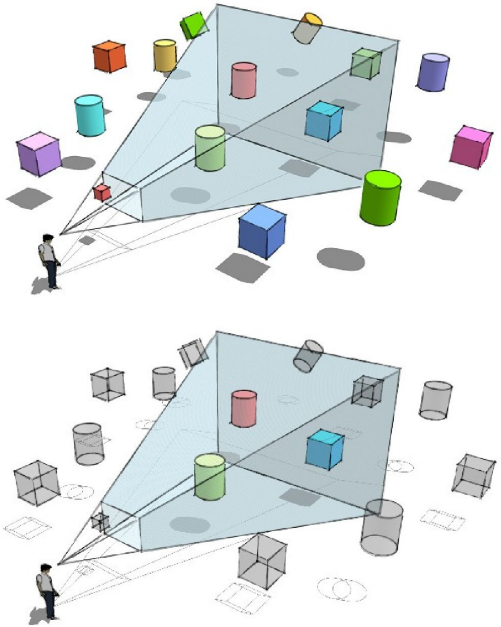
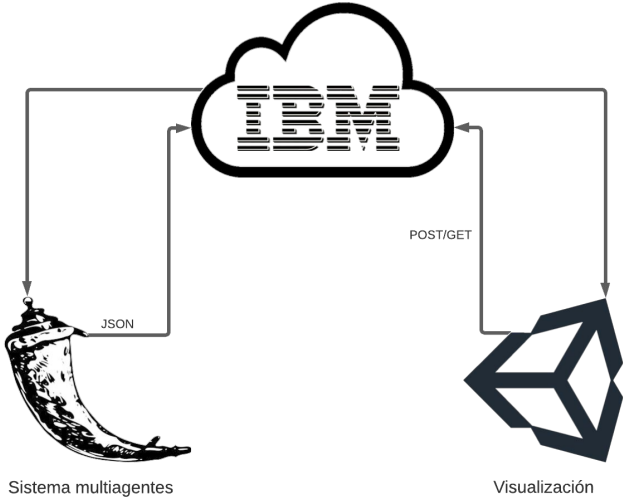


Diagrama de la arquitectura

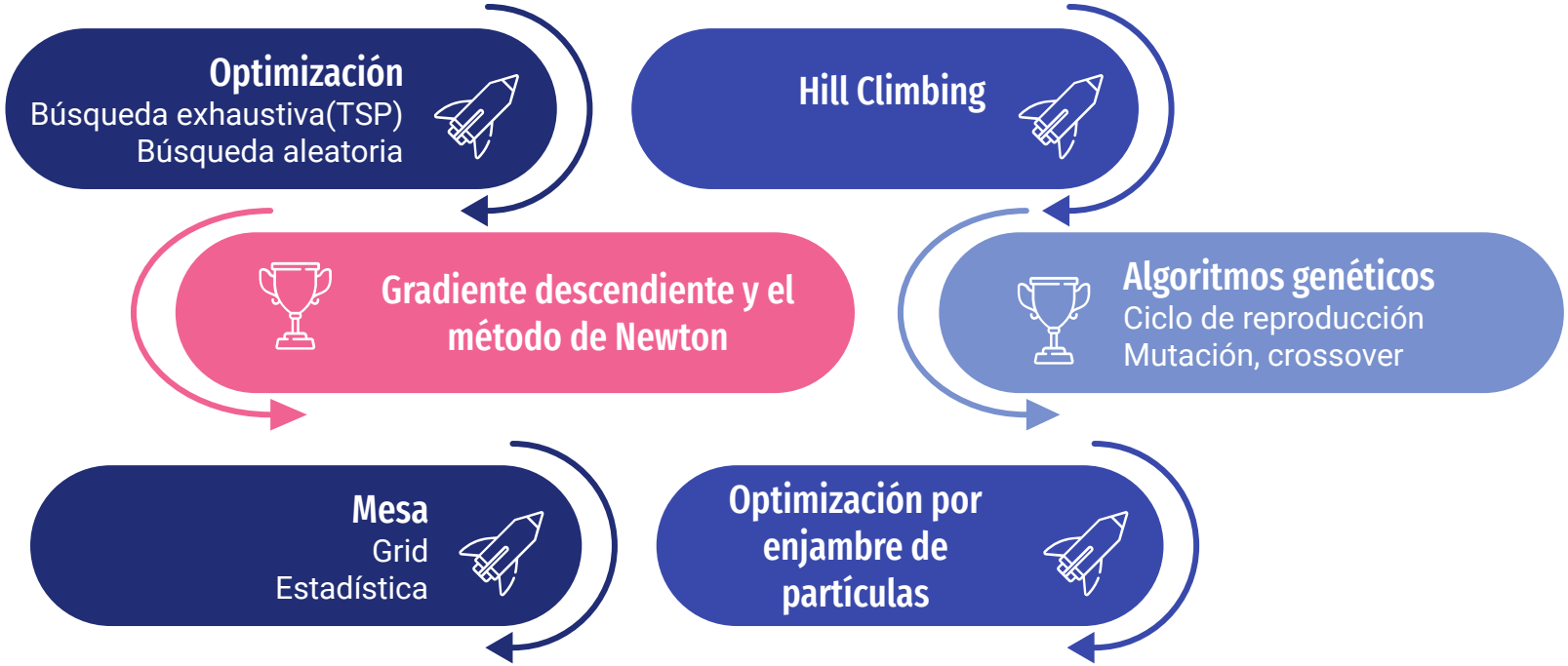


Ejemplo de conexión con localhost

```
class Server(BaseHTTPRequestHandler):

    def _set_response(self):
        self.send_response(200)
        self.send_header('Content-type', 'text/html')
        self.end_headers()

    def do_GET(self):
        logging.info("GET request,\nPath: %s\nHeaders:\n%s\n", str(self.path), str(self.headers))
        positions = updatePositions()
        self._set_response()
        resp = "{\"data\":" + positionsToJSON(positions) + "}"
        self.wfile.write(resp.encode('utf-8'))

    def do_POST(self):
        content_length = int(self.headers['Content-Length'])
        post_data = json.loads(self.rfile.read(content_length))
        logging.info("POST request,\nPath: %s\nHeaders:\n%s\n\nBody:\n%s\n",
                     str(self.path), str(self.headers), json.dumps(post_data))

        positions = updatePositions()
        self._set_response()
        resp = "{\"data\":" + positionsToJSON(positions) + "}"
        self.wfile.write(resp.encode('utf-8'))
```

```
IEnumerator SendData(string data)
{
    WWWForm form = new WWWForm();
    form.AddField("bundle", "the data");
    string url = "http://localhost:8585";
    //using (UnityWebRequest www = UnityWebRequest.Post(url, form))
    using (UnityWebRequest www = UnityWebRequest.Get(url))
    {
        byte[] bodyRaw = System.Text.Encoding.UTF8.GetBytes(data);
        www.uploadHandler = (UploadHandler)new UploadHandlerRaw(bodyRaw);
        www.downloadHandler = (DownloadHandler)new DownloadHandlerBuffer();
        //www.SetRequestHeader("Content-Type", "text/html");
        www.SetRequestHeader("Content-Type", "application/json");

        yield return www.SendWebRequest();          // Talk to Python
```

Aprendizajes obtenidos: Sistemas multiagentes



Actividades pendientes

- Seguir con la simulación en Unity, específicamente en la parte de los semáforos y los coches dando vuelta en las intersecciones. Los encargados son Diego y Agustín.
- Por la parte de python, aún debemos seguir codificando la lógica. Hay una parte faltante en la definición de las clases, la encargada es Yus. Del resto, todos somos responsables.
- Ambas actividades se realizarán el día sábado y en la tarde del viernes.

Diagrama de clases de la solución propuesta

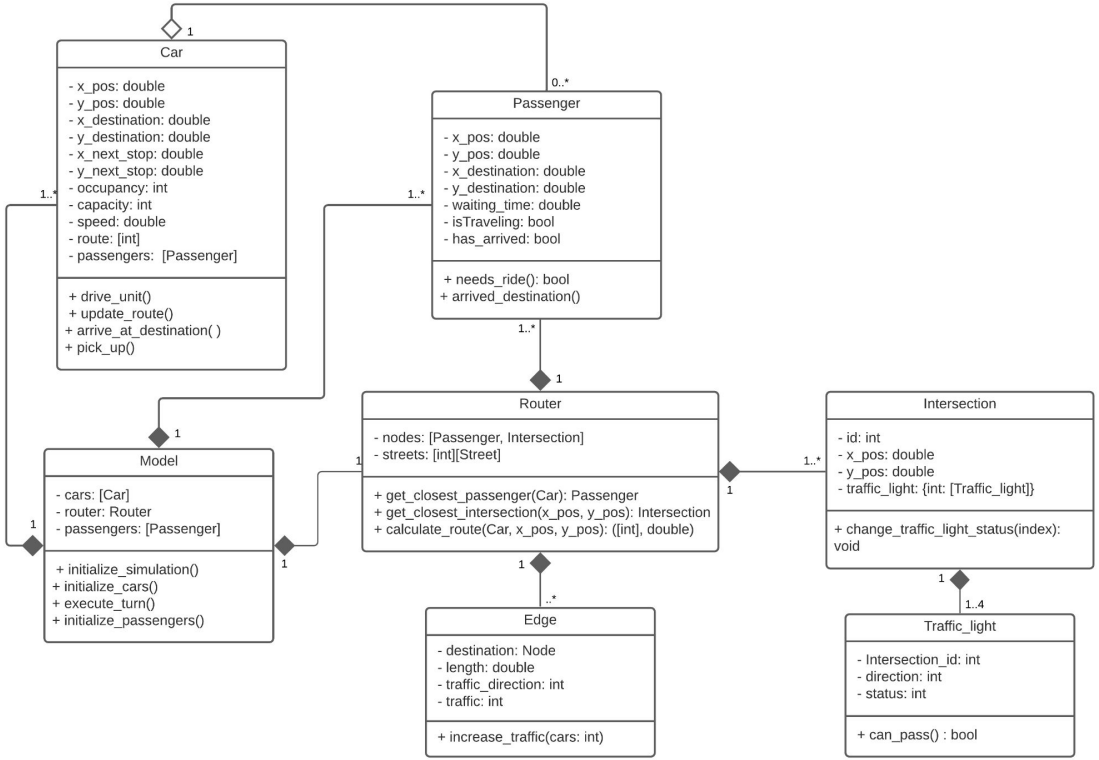
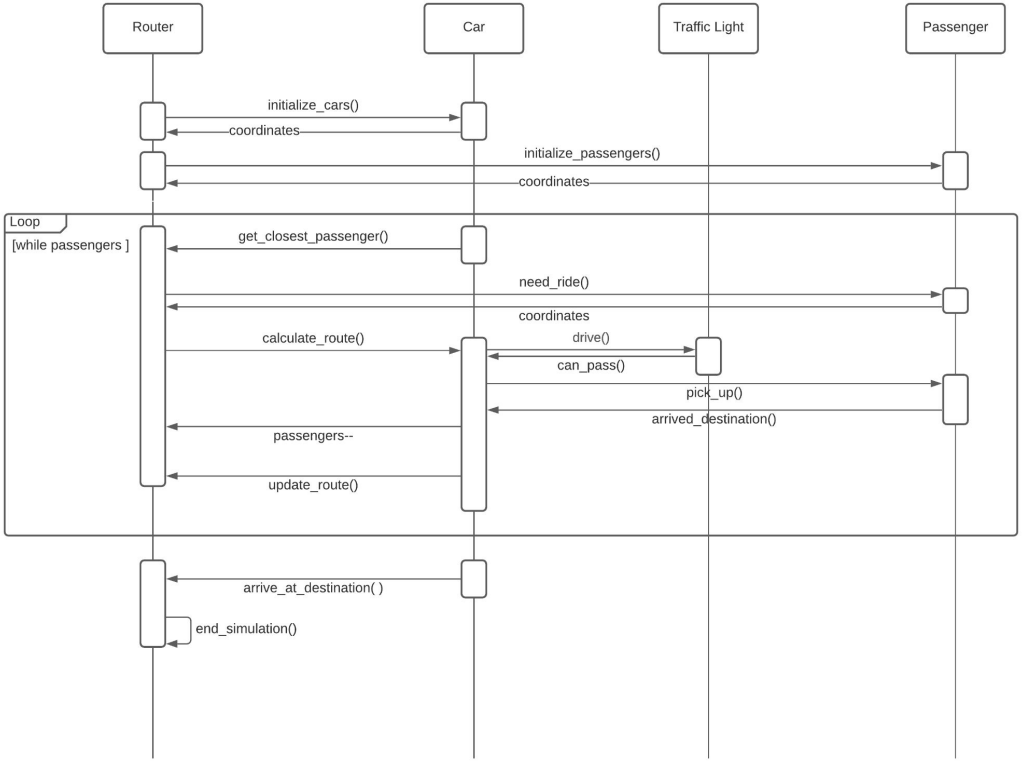


Diagrama de interacción de los agentes



Plan de trabajo completo:

https://docs.google.com/spreadsheets/d/1VUu2x0GdWkeFjR5Wqz8ogRXiVcVxM9z49T1Q9Nsx73w/edit?usp=sharing

Evidencias de trabajo