



INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

Campus Estado de México

TC2008B.302

Modelación de sistemas multiagentes con gráficas computacionales (Gpo 302)

Equipo 2

| | |
|---------------------------------|-----------|
| Víctor Alejandro Morales García | A01749831 |
| Aislinn Ruiz Sandoval | A01750687 |
| Miguel Ángel Galicia Sánchez | A01750744 |
| David Sánchez Báez | A01798202 |

Profesor

Jorge Adolfo Ramírez Uresti

Mauricio Bezares Peñúñuri

Fecha de entrega:

21 de noviembre de 2024

| | |
|--|-----------|
| Descripción del Medio Ambiente..... | 3 |
| Diagramas de Agente AUML..... | 4 |
| • Automóvil..... | 4 |
| • Peatón..... | 4 |
| • Autobús..... | 4 |
| • Ambulancia..... | 5 |
| • Semáforo:..... | 5 |
| Diagrama de organización SMA..... | 6 |
| Diagrama de interacción..... | 7 |
| Plan de trabajo..... | 8 |
| Cronograma General..... | 8 |
| Fase 1: Creación de la Ciudad en Blender..... | 9 |
| Fase 2: Conexión de Unity con Python..... | 9 |
| Fase 3: Creación de Agentes (Ambulancia y Grúa)..... | 10 |
| Fase 4: Pruebas Finales y Ajustes..... | 10 |
| Aprendizaje adquirido..... | 11 |
| Victor..... | 11 |
| Miguel..... | 11 |
| David..... | 11 |
| Aislinn..... | 11 |

Descripción del Medio Ambiente

| | | | |
|--------------|------|-----------------|-----|
| Accesible | 75% | No accesible | 25% |
| Determinista | 40% | No determinista | 60% |
| No episódico | 80% | Episódico | 20% |
| Dinámico | 80% | No dinámico | 20% |
| Discreto | 100% | Continuo | 0% |

| | |
|----------------------|--|
| Accesibilidad | Todos los agentes tienen acceso parcial a la información del entorno limitado solo a sus proximidades. |
|----------------------|--|

| | |
|---------------------|---|
| Determinista | Debido a que el comportamiento de algunos agentes como la ambulancia en situaciones de emergencia puede alterar el flujo de tráfico. |
| No episódico | Las acciones de un agente afectan el comportamiento de otros. |
| Dinámico | Con los constantes cambios en las posiciones y acciones de los agentes. |
| Discreto | Debido a que se trata de un entorno virtual, los datos utilizados son finitos, pues incluso los agentes se encuentran en un mapa con un tamaño delimitado |

Diagramas de Agente AUMML

- Automóvil
 - Diagrama de Agente

| |
|---|
| Grupo: Vehículo Rol: Transporte |
| Servicio: Traslado |
| Protocolo: Optimización de ruta |
| Eventos: Obstáculo en la vía, cambio de semáforo |
| Metas: trasladarse a un edificio destino Planes: Ajustar velocidad y ruta rápida Acciones: Detenerse en los semáforos |

| |
|---|
| y esquivar obstáculos |
| Conocimiento: Mapa de la ciudad, estado del tráfico, localización de destinos |

- Peatón

- Diagrama de Agente

| |
|--|
| Grupo: Persona Rol: Atravesar la ciudad |
| Servicio: Traslado |
| Protocolo: Interacción peatonal |
| Eventos: Obstáculos en su camino, cambio de semáforo |
| Metas: Llegar a un edificio destino Planes: Escoger la mejor ruta Acciones: Mantenerse a salvo |
| Conocimiento: Mapa de la ciudad, uso de autobuses, uso de la ambulancia |

- Autobús

- Diagrama de Agente

| |
|--|
| Grupo: Vehículo Rol: Transporte público |
| Servicio: llevar peatones a su destino |
| Protocolo: Obstáculo en la vía, cambio de semáforo, solicitud de parada |
| Eventos: cambio de semáforo, obstáculo en la vía, ingreso de peatón, egreso de peatón |
| Metas: Transitar todas las paradas establecidas Planes: Completar la ruta en tiempo y forma. Acciones: Ingresar peatones y |

| |
|------------------------------------|
| egresar peatones |
| Conocimiento: Ruta preestablecida. |

- Ambulancia

- Diagrama de Agente

| |
|--|
| Grupo: Vehículo Rol: Emergencia |
| Servicio: llevar al peatón al hospital |
| Protocolo: Asistencia de emergencia |
| Eventos: Accidentes detectados, solicitud de auxilio |
| Metas: llevar al hospital al peatón Planes: ruta mas rapida al hospital Acciones: dar prioridad en las intersecciones, activar sirena |
| Conocimiento: Ruta al hospital, mapa de la ciudad |

- Semáforo:

- Diagrama de Agente

| |
|--|
| Grupo: Vehículo Rol: Regular el tráfico |
| Servicio: Flujo vehicular |
| Protocolo: Coordinar con semáforos |
| Eventos: detección de vehículos |
| Metas: flujo ordenado y seguro Planes: adaptar el tiempo de cambio Acciones: cambiar la luz comunicación con semáforos ceranos |
| Conocimiento: estado del tráfico reglas de prioridad de cruce |

Diagrama de organización SMA

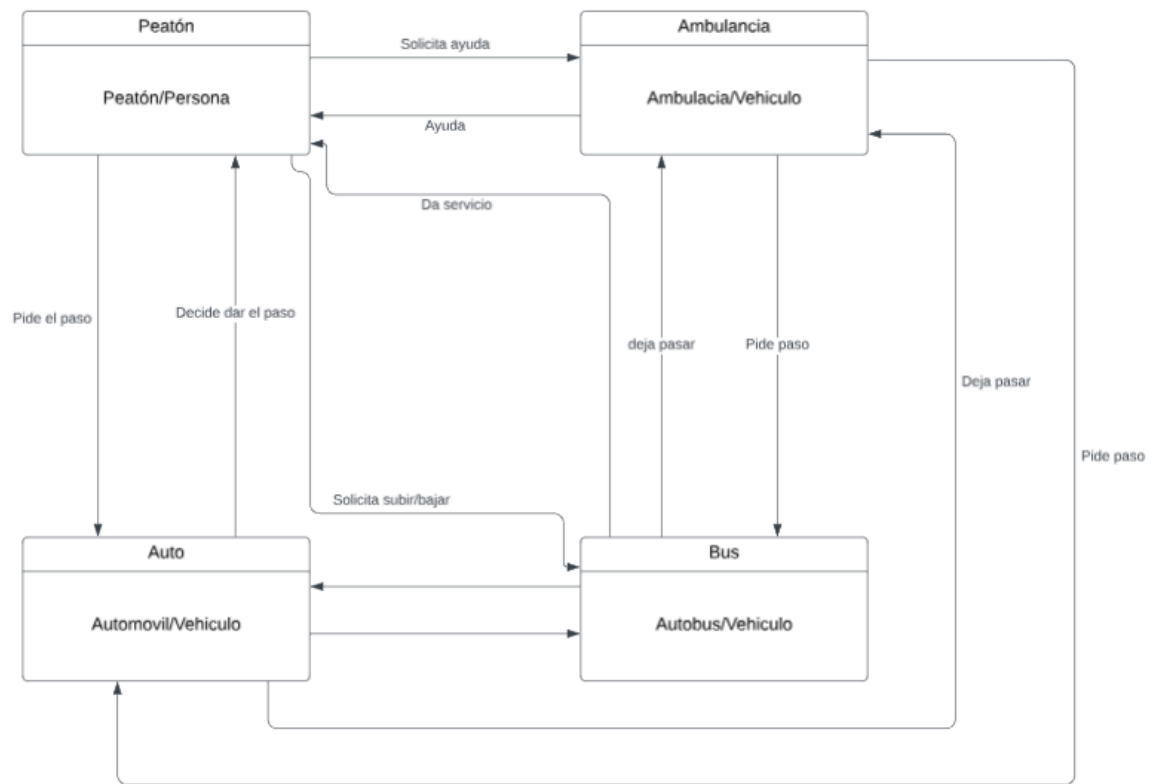
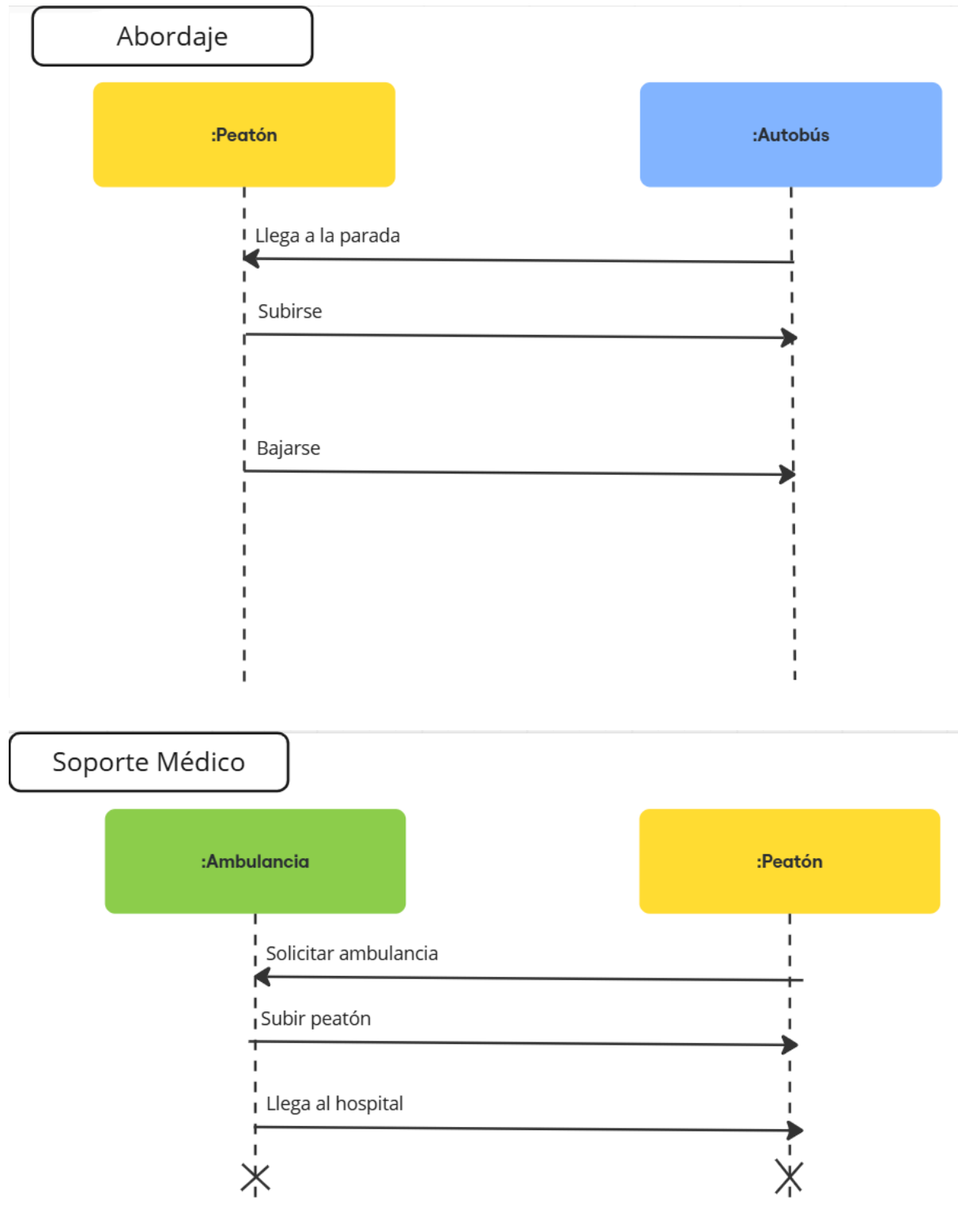
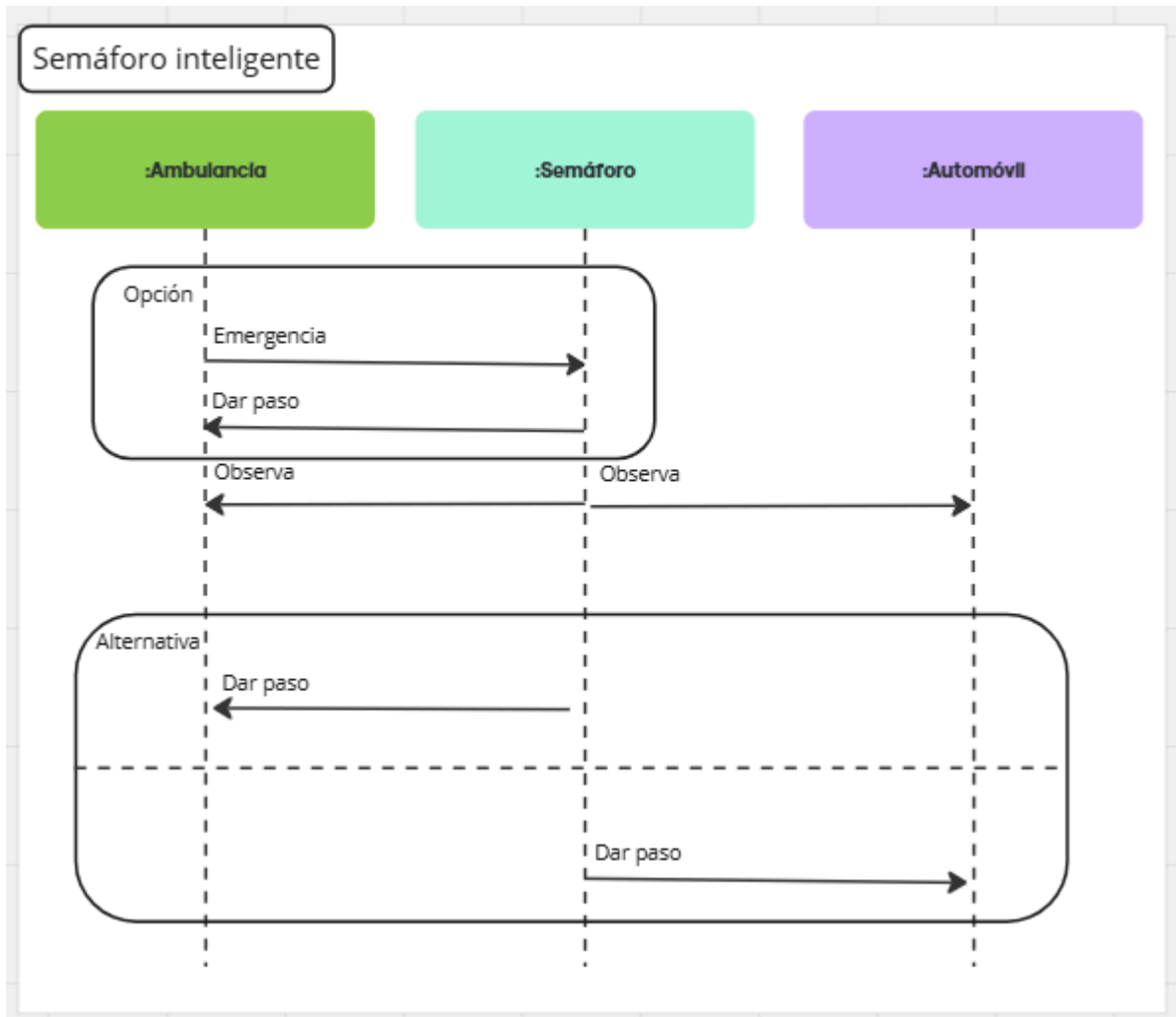


Diagrama de interacción





https://miro.com/welcomeonboard/dkpiVVZ6REN4WFN5bUVLSIRRSgdoNzd4b1FmeFhyYWN3bGRRZmZDV1kxNmtROHdKbDRWQnZTVVNmZ0dxeW11cnwzMDc0NDU3MzU0NmM3ODYzNTQ5fDI=?share_link_id=258724566456

Plan de trabajo

Cronograma General

| Fase | Duración | Fecha de Inicio | Fecha de Fin | Responsables |
|---------------------------|----------|-----------------|--------------|-----------------|
| Fase 1: Ciudad en Blender | 3 días | 18/11/2024 | 21/11/2024 | David y Aislinn |

| | | | | |
|----------------------------------|--------|------------|------------|--------------------|
| Fase 2: Conexión Unity-Python | 2 días | 21/11/2024 | 23/11/2024 | Víctor y Miguel |
| Fase 3: Creación de Agentes | 2 días | 23/11/2024 | 25/11/2024 | Todo el equipo |
| Fase 4: Pruebas y Ajustes | 3 días | 25/11/2024 | 28/11/2024 | Todo el equipo |

Fase 1: Creación de la Ciudad en Blender

Duración: 3 días

Tareas:

- Modelado en Blender**
 - Diseñar edificios, calles, señales de tráfico, y áreas específicas como hospitales y zonas de estacionamiento.
 - Buscar detalles como árboles, lámparas, y texturas para dar una mejor estética y realismo a nuestra ciudad.
- Exportación de Blender**
 - Exportar la ciudad en formato compatible (FBX o similar) y asegurarse de que las texturas se mantengan intactas.
- Integración en Unity**
 - Importar la ciudad a Unity y ajustar la escala y posiciones de los objetos.
 - Verificar que las calles tengan colisionadores y caminos navegables para agentes.

Entregable: Ciudad completa y funcional importada en Unity.

Fase 2: Conexión de Unity con Python

Duración: 2 días

Tareas:

- Configuración de Comunicación**
 - Implementar comunicación bidireccional entre Unity y Python utilizando bibliotecas como [Socket](#) ,[Unity-Python](#) o ([método del profe](#)).
 - Verificar el intercambio de datos en tiempo real (posiciones de agentes, evento de hospital ,semáforos y coche).
- Pruebas Iniciales**
 - Recibir datos de Unity, como las coordenadas de los agentes o eventos.

Entregable: Sistema funcional que permite la comunicación en tiempo real entre Unity y Python.

Fase 3: Creación de Agentes (Ambulancia y Grúa)

Duración: 2 días

Tareas:

1. **Diseño y Modelado de Agentes**
 - Crear los modelos 3D de la ambulancia y la grúa en Blender o usar modelos existentes.
 - Integrar animaciones básicas como movimiento de ruedas, luces de emergencia, y grúas operativas.
2. **Programación de Comportamiento**
 - Definir rutas de navegación utilizando el sistema de NavMesh de Unity.(ver si alguien mas tiene otros métodos)
 - Implementar lógica en Python o Unity para que los agentes respondan a eventos específicos:
 - Ambulancia: Dirigirse al hospital más cercano en respuesta a emergencias.
 - Grúa: Reubicar vehículos accidentados.
3. **Integración con Python**
 - Controlar los agentes desde Python mediante comandos para simular decisiones complejas o escenarios dinámicos.
 - Asegurar que los agentes interactúen correctamente con el entorno (respetar señales de tráfico, ruta más rápida o evitar crear otro accidente).
4. **Pruebas de Simulación**
 - Verificar el comportamiento autónomo de los agentes en múltiples escenarios.
 - Realizar ajustes según los resultados de las pruebas.

Entregable: Ambulancia y grúa funcionales, con comportamiento autónomo e interacción con Python.

Fase 4: Pruebas Finales y Ajustes

Duración: 3 días

Tareas:

1. **Simulación Completa**
 - Ejecutar pruebas integrales donde los agentes naveguen en la ciudad simulada, respondiendo a eventos generados por Python.
2. **Optimización**

- Mejorar el rendimiento del entorno en Unity si es necesario (ajustes de scripts).
- Corregir errores en la lógica de agentes o conexión entre Unity y Python.

Entregable: Simulación completa.

Aprendizaje adquirido

Victor

Aprendí sobre el uso de Mesa y poder integrar mejor los agentes y objetos al entorno para la simulación, por otra parte, en tanto a gráficas, aprendí más sobre el modelado en blender y los materiales que dan color al modelo en 3D, de igual forma aprendí a modificar el modelo, por ejemplo, a cortarlo y moldearlo.

Miguel

Aprendí más sobre los agentes y la librería Mesa para la interacción entre peatón. También aprendí sobre el uso de modelos 3D dentro de Unity para el modelado de la ciudad que será utilizada en la simulación.

David

Aprendí mucho más a cómo funciona unity (atajos y dominio de las vistas) y todas las posibilidades que se pueden hacer así como modelar los agentes que pronto tomarán las figuras que hicimos en blender o que tomamos de algún sitio web para terminar nuestro programa

Aislinn

Aprendí más a fondo el funcionamiento de Unity ya que al ir importando materiales y objetos que vamos necesitando al proyecto, más enriquecedor se vuelve, además, aprendí más sobre la escala de los objetos dentro del mapa y su comportamiento de un objeto dentro de otro para complementarlo.