



Instituto Tecnológico de Estudios Superiores de Monterrey

Modelación de sistemas multiagentes con gráficas computacionales

Grupo 101

Equipo Druids

Arturo Ramos Martínez A01643269

Adolfo Hernández Signoret A01637184

Bryan Ithan Landín Lara A01636271

Diego Enrique Vargas Ramírez A01635782

Luis Fernando Cuevas Arroyo A01637254

Campus Guadalajara

Viernes 6 de septiembre de 2024

Tabla de contenidos

| | |
|---|-----------|
| Tabla de contenidos | 2 |
| 1.0. Conformación del equipo de trabajo (Fortalezas y debilidades) : | 3 |
| Arturo Ramos Martínez: | 3 |
| Adolfo Hernández Signoret: | 3 |
| Bryan Ithan Landín Lara: | 3 |
| Diego Enrique Vargas Ramírez: | 4 |
| Luis Fernando Cuevas Arroyo: | 4 |
| 2.0. Listado de objetivos y compromisos: | 5 |
| 2.1.0. Compromisos: | 6 |
| 3.0. Herramientas de trabajo colaborativo: | 6 |
| 4.0. Propuesta formal del reto: | 6 |
| 4.1. Agentes involucrados | 7 |
| 4.1. Propiedades de Agentes involucrados | 7 |
| 5.0. Diagramas. | 9 |
| 5.1. Diagrama de clases de agentes involucrados. | 9 |
| 5.2. Diagrama de protocolos de interacción finales | 9 |
| 6.0. Utilidad o éxito de los agentes, con gráficos | 10 |
| 6.1. Gráficos | 10 |
| 7.0. Propiedades, con justificación | 10 |
| 8.0. Reflexiones: | 11 |
| Arturo Ramos Martínez: | 11 |
| Adolfo Hernández Signoret: | 11 |
| Bryan Ithan Landín Lara: | 11 |

Diego Enrique Vargas Ramírez: 11

Luis Fernando Cuevas Arroyo: 11

1.0. Conformación del equipo de trabajo (Fortalezas y debilidades) :

Arturo Ramos Martínez:

Fortalezas:

- ***Conocimiento Técnico:*** Experiencia en programación, con lenguajes como python o c++
- ***Resolución de Problemas:*** Habilidad para encontrar soluciones a problemas complejos.

Áreas de oportunidad:

- ***Gestión del Tiempo:*** Necesita mejorar en la planificación y distribución de tareas para evitar retrasos en la entrega de sus responsabilidades.
- ***Delegación:*** Asumir demasiadas responsabilidades, lo que puede llevar a una sobrecarga de trabajo.

Adolfo Hernández Signoret:

Fortalezas:

- ***Desarrollo del Endpoint:*** Sabe utilizar fast API.
Control del estado de las variables compartidas entre unity y python

Áreas de oportunidad:

- ***Tiempo de dedicación.*** Podría dedicar más tiempo para obtener los resultados no tan cercanos a la fecha de entrega.

Bryan Ithan Landín Lara:

Fortalezas:

- ***Diseño de Interfaces de Usuario:*** Sabe utilizar Unity para crear escenas realistas y que reflejan la simulación adecuadamente.

- ***Creatividad:*** Habilidad para agregar componentes que resuelven conflictos de una manera que no a todos parece obvia.
- ***Colaboración:*** Facilidad para trabajar en equipo, aportando ideas y ayudando a otros miembros a superar obstáculos.

Áreas de Oportunidad:

- ***Conocimiento Técnico:*** Podría beneficiarse de una mayor familiarización con los conceptos avanzados de programación de sistemas multiagente.
- ***Priorización:*** Necesita mejorar en la identificación de las tareas más críticas para enfocarse en ellas primero.

Diego Enrique Vargas Ramírez:

Fortalezas:

- ***Analítica:*** Habilidades para el análisis de datos y la interpretación de patrones, lo que es clave para mejorar la eficiencia del sistema de patrullaje.
- ***Optimización de Algoritmos:*** Capacidad para optimizar algoritmos, logrando una mejor performance del sistema en tiempo real. Implementando estrategias como el star path finding.

Áreas de Oportunidad:

- ***Documentación:*** Necesita mejorar en la documentación de su trabajo para que otros miembros del equipo puedan entender y seguir su progreso.
- ***Multitasking:*** Puede tener dificultades para manejar múltiples tareas simultáneamente, lo que podría afectar su rendimiento.

Luis Fernando Cuevas Arroyo:

Fortalezas:

- ***Gestión de Proyectos:*** Capacidad para planificar y gestionar proyectos, asegurando que el equipo se mantenga en línea con los objetivos y plazos acordados.
- ***Comunicación efectiva:*** Habilidad para comunicar ideas y estrategias de manera clara, tanto al equipo como a otros participantes como el profesor.
- ***Negociación y Resolución de Conflictos:*** Experiencia en manejar conflictos dentro del equipo y llegar a soluciones que beneficien a todos.

Áreas de Oportunidad:

- ***Profundización Técnica:*** Se puede mejorar los conocimientos técnicos para entender más a fondo los desafíos que enfrenta el equipo en términos de programación y desarrollo.
- ***No ceder:*** A veces cede demasiado aunque sabe que él está en lo correcto con tal de evitar un conflicto.

2.0. Listado de objetivos y compromisos:

- Entender y aplicar conocimientos sobre los agentes, así como de los sistemas multiagentes.
- Lograr comunicar las distintas tecnologías, en este caso la comunicación entre python y unity con C#, por medio de una api con Fast API para manejar las peticiones de unity.
- Mejorar nuestra planeación de proyectos sobre agentes, aprendiendo a diseñar los distintos diagramas (poner los tipos de diagramas).
- Mejorar nuestra organización como equipo.
- Presentar una simulación que cumpla con lo que se nos pide. sobre todo apegándose lo más posible a la programación orientada a los agentes.

- Y sobre todo aprender todo lo que podamos en el proceso.

2.1.0. Compromisos:

- Comunicarnos lo mejor posible, es importante que cualquier duda o propuesta sea expresada, para estar claros con nuestra propuesta para el reto, fechas de entrega y avances. Esto por medio del grupo de whatsapp y en las reuniones de equipo.
- Reunirnos todos los días por lo menos 3 horas hasta la fecha de entrega. Además de cada quien trabajar por su cuenta.
- Avisar al equipo cuando se haga alguna actualización en el código (cuando se sube algún cambio al repositorio de github)
- Comprometernos y responsabilizarnos con las partes que a cada integrante del equipo le toca desarrollar.
- Crear un ambiente ideal para el trabajo colaborativo, dejando de lado actitudes que puedan afectar la comunicación.

3.0. Herramientas de trabajo colaborativo:

- El repositorio de github donde guardaremos la información es el siguiente:
https://github.com/ArturoRM22/drone_agents
- Para la comunicación entre los participantes optamos por un grupo de WhatsApp para una comunicación más clara y directa.

4.0. Propuesta formal del reto:

Descripción de la problemática:

La problemática que enfrentamos es la falta de seguridad en lugares de grandes dimensiones , esto más que nada por la dificultad que implica el estar vigilando cada parte, además del tiempo que toma por ejemplo desplazarse a estos mismos sitios. Esta falta de coordinación entre sistemas de

seguridad genera brechas de vigilancia y aumenta los tiempos de respuesta y esto provoca incidentes no detectados y mayores pérdidas económicas e incluso pone en riesgo la vida de las personas.

- **Descripción.**

El proyecto consiste en diseñar e implementar un sistema multiagentes que simule un patrullaje coordinado en una fábrica/mega-almacén. El objetivo principal es lograr una cobertura efectiva de vigilancia, por medio de un dron, cámaras y un guardia de seguridad. El flujo será el siguiente:

4.1. Agentes involucrados

- **Dron:** Tendrá la tarea de seguir una ruta de patrullaje predefinida, con el objetivo de vigilar la zona; en caso de detectar algo sospechoso, el dron notificará/alertará al guardia, para que el delibere si se trata o no de un peligro o de alguna situación que no debería estar pasando.
- **Cámaras:** Tendrán un rol muy parecido al del dron, pero estas en lugar de comunicarse con el guardia al detectar que algo anda mal, por ejemplo si ven algún intruso, lo que harán será comunicarse con el dron, para que este se acerque a la zona y decida si comunicarse con el guardia o no.
- **Guardia:** Tendrá la capacidad de tomar control del dron cuando este se comunique con él, para hacer una inspección más detallada y decidir si encender o no la alarma del lugar.
- **Intruso/Empleado:** El funcionamiento de este agente es demostrar la comunicación de los agentes encargados de la vigilancia, agregamos un 4 agente, el cual actuará como intruso o empleado; esté estará rondando por el lugar, hasta que el dron, o alguna cámara lo detecte, en ese momento, tanto el guardia, el dron y las cámaras, establecerán una comunicación y decidirán qué hacer.

4.1. Propiedades de Agentes involucrados

- Dron:

- **Autonomía:** El dron patrulla por sí solo e incluso puede iniciar la lógica de mensajes al detectar un intruso o ser alertado primero por las cámaras.
- **Reactividad:** El dron reacciona a los eventos de vigilancia y posibles intrusos detectados por él mismo o por las cámaras.
- **Proactividad:** Realiza patrullajes de manera autónoma siguiendo una ruta predefinida, anticipándose a posibles incidentes.
- **Socialización:** Se comunica con las cámaras y el guardia, notificando sobre posibles amenazas y colaborando para tomar decisiones conjuntas.

- Cámaras:

- **Autonomía:** Las cámaras si bien actúan por sí cuenta al detectar intrusos, este es el límite de su autonomía. El resto de acciones dependen de detectar a un intruso.
- **Reactividad:** Detectan la presencia de intrusos y responden de manera inmediata, alertando al dron para que verifique la situación.
- **Proactividad:** Aunque las cámaras no patrullan, están activamente monitoreando la zona y pueden detectar actividades sospechosas de manera continua.
- **Socialización:** Se comunican con el dron, solicitando que inspeccione áreas donde se detecta una posible intrusión.

- Guardia:

- **Autonomía:** El guardia y toda su lógica es dependiente de que se detecte a un intruso dentro del mapa, sino este no es capaz de actuar.
- **Reactividad:** Responde a las notificaciones del dron o cámaras y decide si activar o no las alarmas en función de la información recibida.

- **Proactividad:** Aunque su papel es principalmente reactivo, puede tomar control del dron para investigar más a fondo cuando lo considere necesario.
- **Socialización:** Se comunica tanto con el dron como con las cámaras, recibiendo y procesando la información antes de tomar una decisión sobre las alarmas.
- **Intruso:**
 - **Autonomía:** El intruso tiene una serie de objetivos en el mapa a los cuales llegar, este se mueve a ellos de manera independiente.
 - **Reactividad:** No responde de manera directa a los eventos de vigilancia, pero su presencia desencadena las acciones de los otros agentes.
 - **Proactividad:** Se mueve de manera autónoma por el área vigilada, simulando el comportamiento de un intruso o empleado.
 - **Socialización:** No se comunica directamente con los otros agentes, pero su detección provoca que los demás agentes colaboren para gestionar la situación.

5.0. Diagramas.

5.1. Diagrama de clases.

Drone / Agent

state-description:

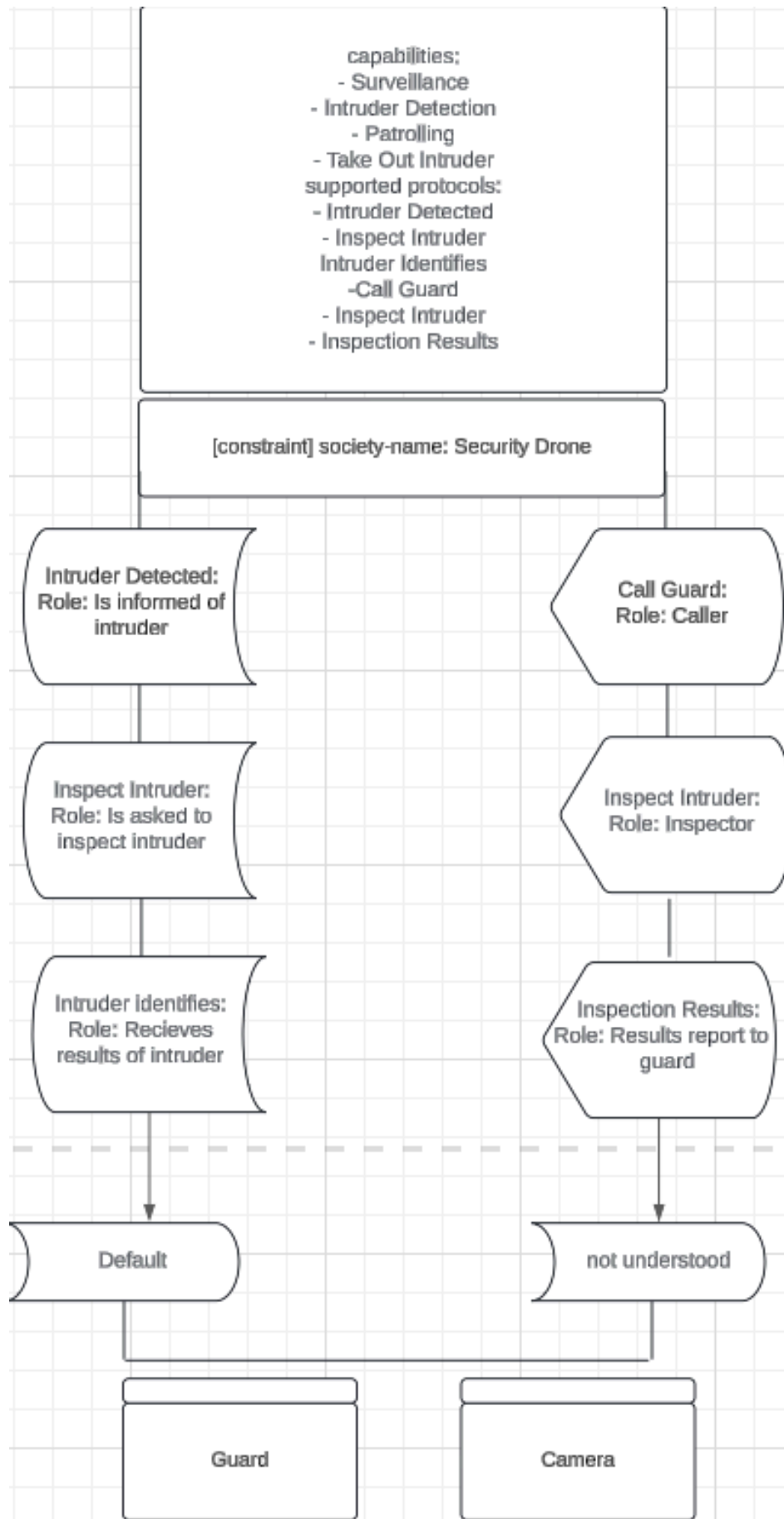
- agentType: int.
- firstStep: bool.
- path_counter: int.
- path:Array<(int,int)>.
- pathfinding_path: Array<(int,int)>.
- target: (int,int).
- currentMessage: string.

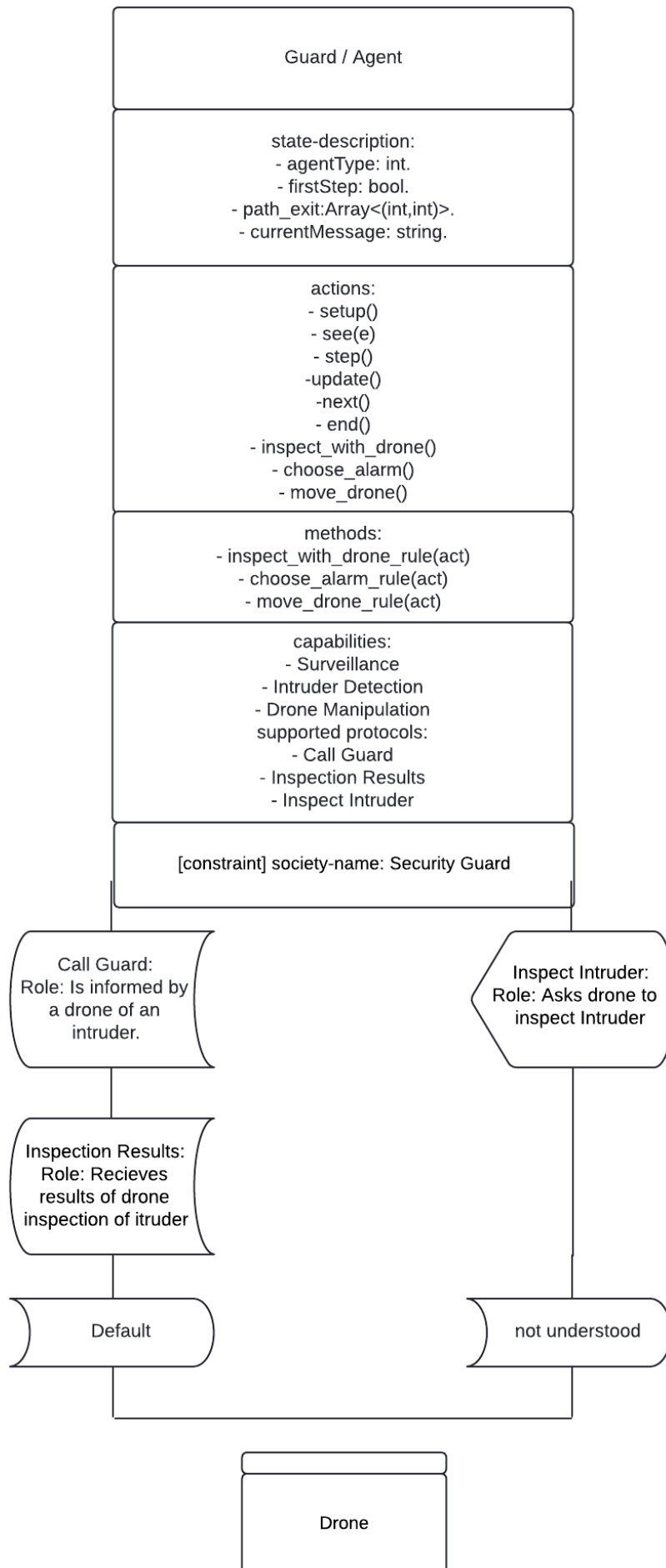
actions:

- setup()
- see(e)
- step()
- update()
- next()
- end()
- check_for_intruder()
- patrol()
- path_to_target()
- move_to_target_intruder()
- call_guard()
- inspect_intruder()
- report_to_guard()
- move_exit()

methods:

- check_for_intruder_rule(act)
- patrol_rule(act)
- path_to_target_rule(act)
- move_to_target_intruder_rule(act)
- call_guard_rule(act)
- inspect_intruder_rule(act)
- report_to_guard_rule(act)
- move_exit_rule(act)





Wall / Agent

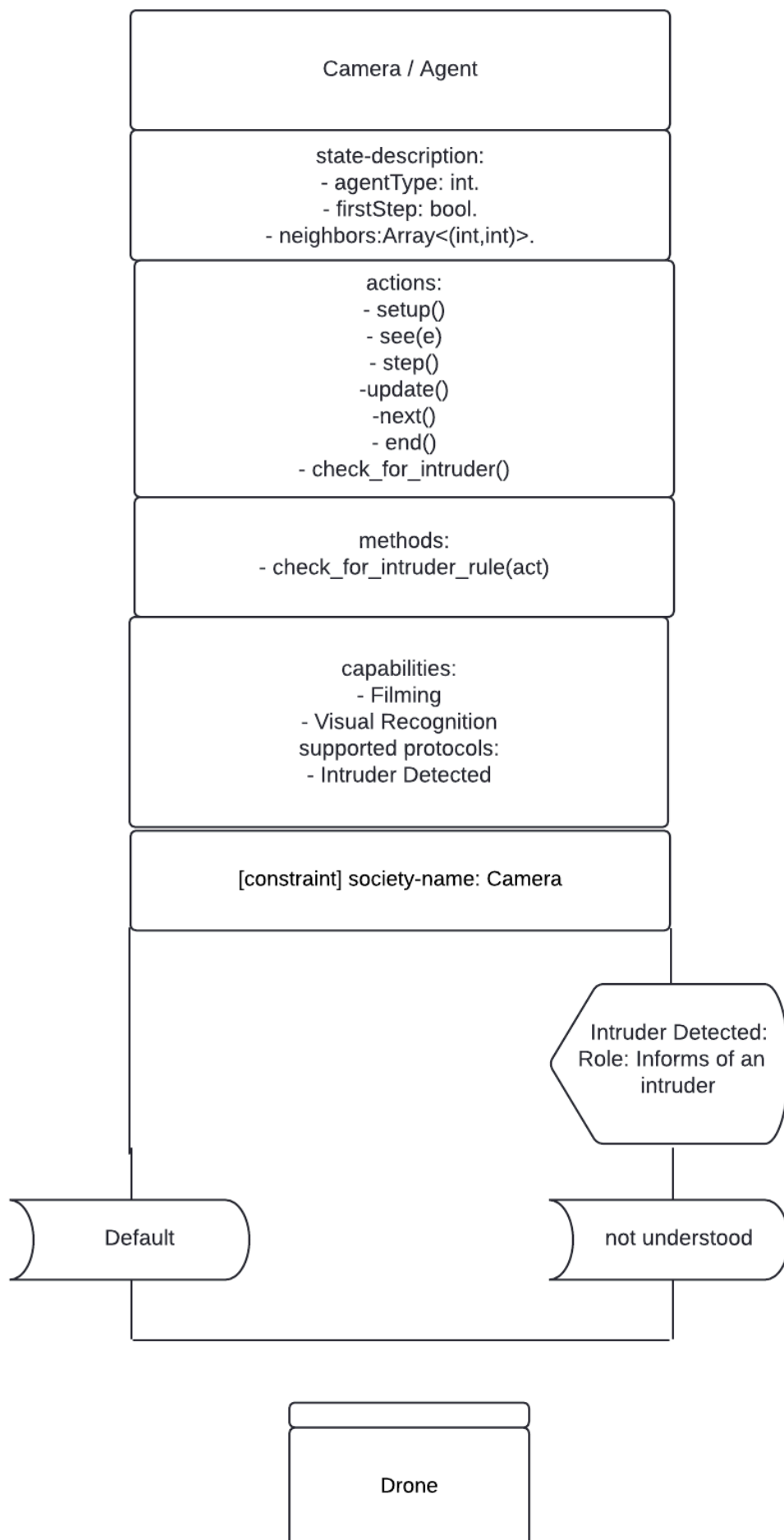
state-description:
- agentType: int.

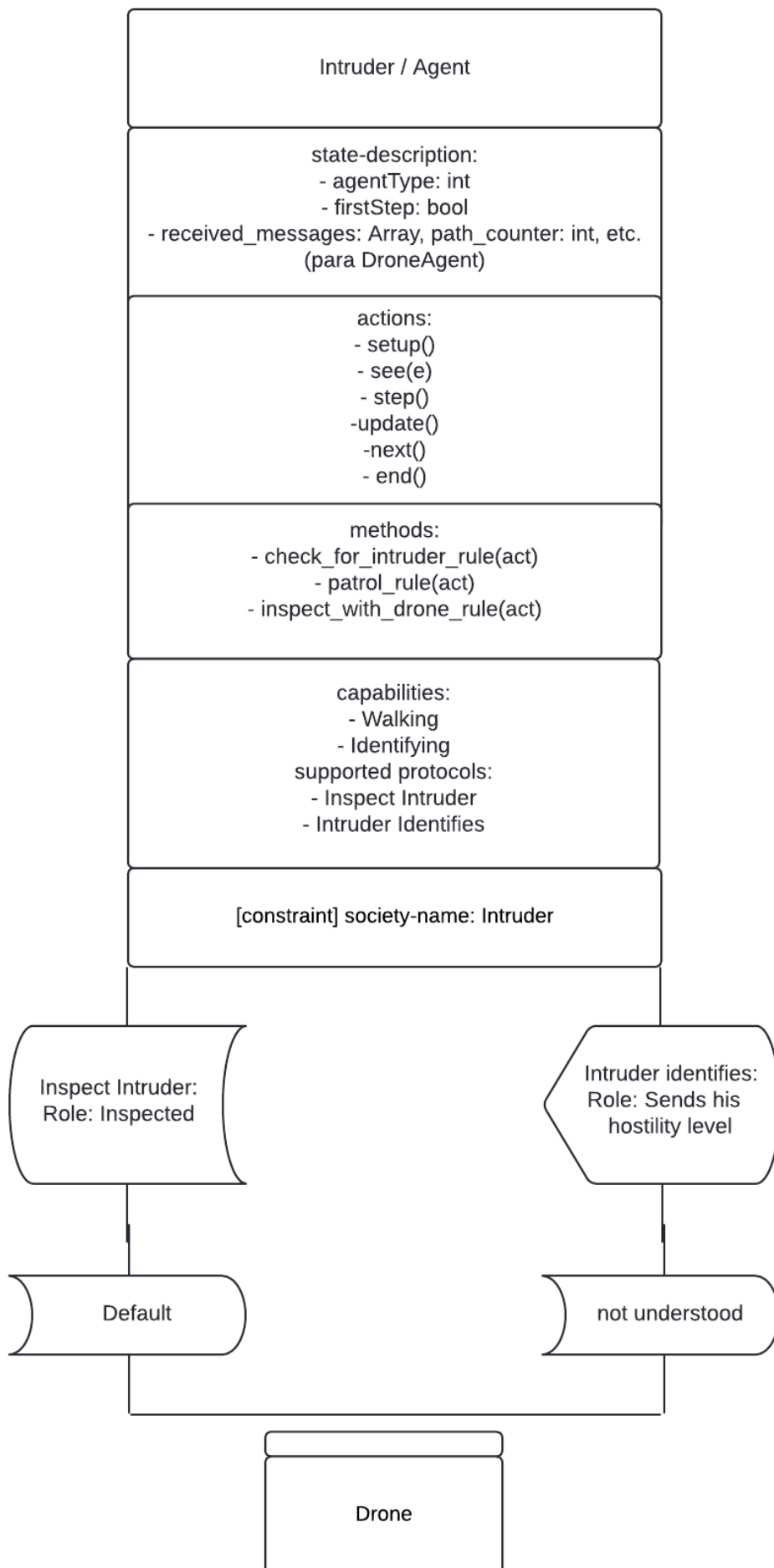
actions:
- setup()
- see(e)
- step()
-update()
-next()
- end()

methods: none

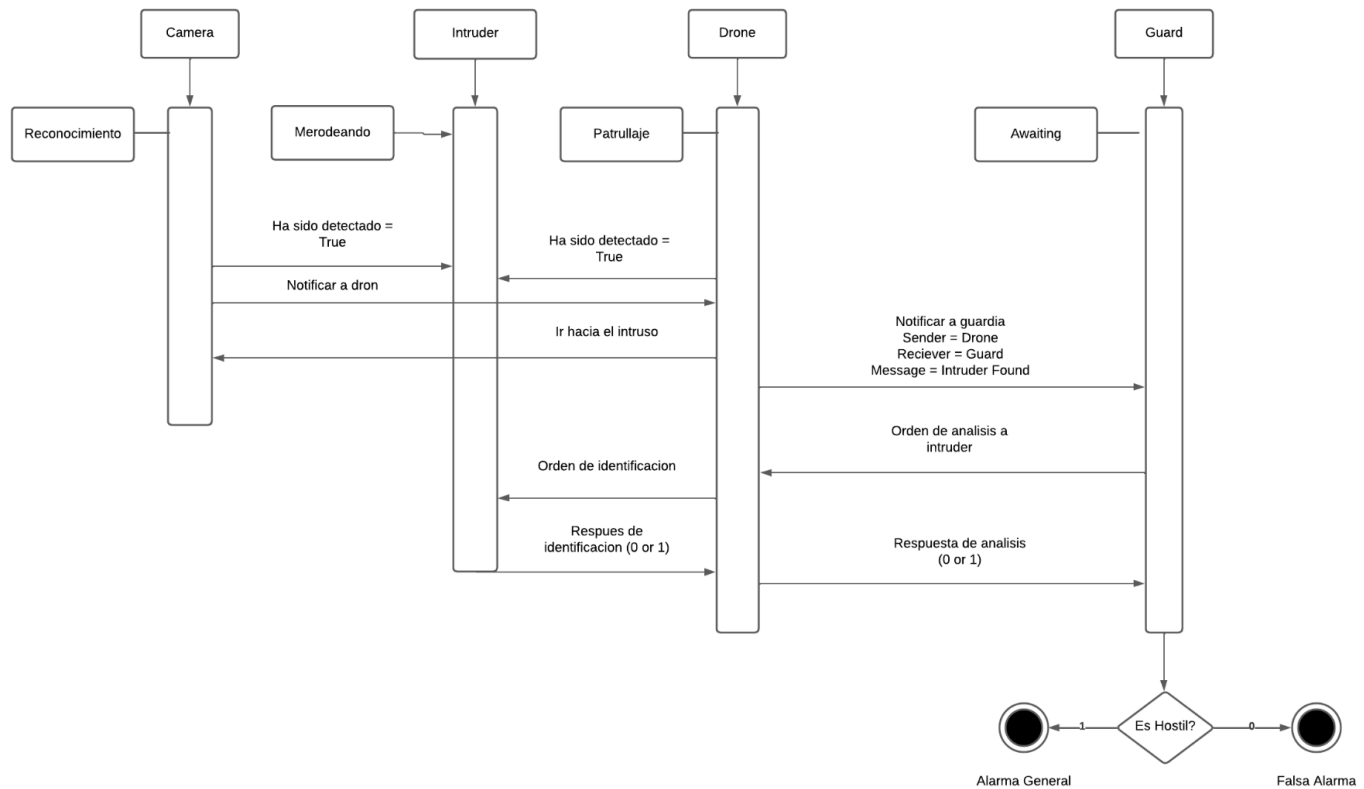
capabilities: none
supported protocols:none

[constraint] society-name: Wall





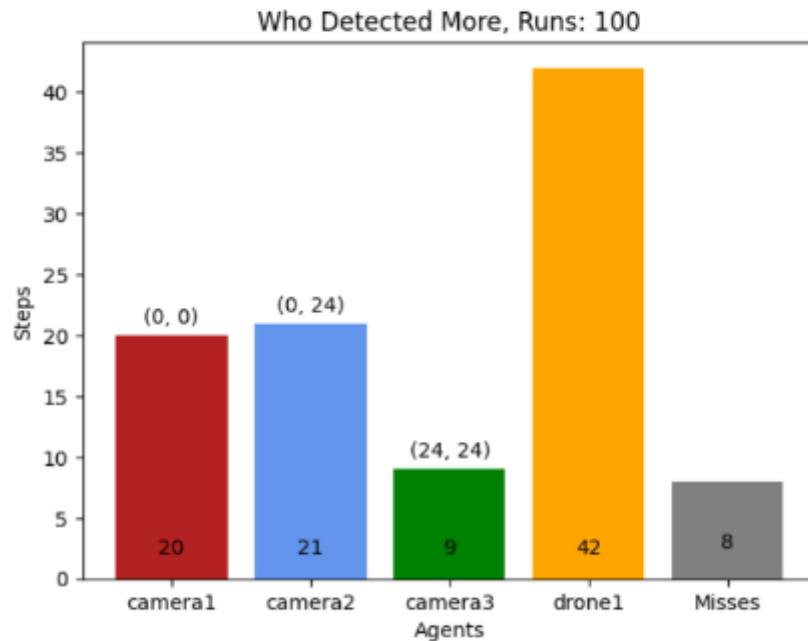
5.2. Diagrama de protocolos de interacción finales



6.0. Utilidad o éxito de los agentes, con gráficos

Para identificar si la simulación ha tenido éxito, se realiza la comparación de si al final de los pasos indicados, la posición del dron y la del intruso son la misma. Desde un punto de vista de cada agente, tiene mayor éxito el que detecta más intrusos. Desde el punto de vista de todos los agentes como comunidad, fracasan si ninguno detecta al intruso.

6.1. Gráficos



7.0. Propiedades, con justificación

1. Ambiente

- a. **Determinista:** El entorno no cambia ni se transforma a lo largo del tiempo, sino que actúa como un campo de juego predeterminado en el cual el agente intruso y el dron se desplazan. Así como los agentes, guardia y cámara simplemente mantienen comunicación.
- b. **Estático:** El ambiente tiene campos ejecutados por pasos, teniendo cada agente su momento específico para analizarlo sin riesgo de que este cambie en plena deliberación. Además, el entorno tiene cambios muy ligeros en temas de posiciones y agentes disponibles.
- c. **Accesible:** Si bien hay paredes que limitan el espacio, nuestro entorno es accesible para todos los agentes. Aquellos específicamente que tengan capacidad de pathfinding.

- d. **Discreto:** Al estar trabajando en una cuadrícula, el entorno de la simulación es discreto. Además, tenemos conjuntos específicos de reglas y acciones que dictan el comportamiento de los agentes. Estos generan un entorno que tendrá posiciones y eventos muy específicos.
- e. **Episódico:** Existe una lógica muy marcada a la hora de cómo funcionan nuestros agentes. A cada momento esperamos que ejecuten cierta parte específica de su código. Esto es notorio especialmente a la hora de comunicarse, teniendo cada mensaje que ser mandado y recibido en cada paso.

8.0. Reflexiones:

Un análisis conciso de la solución desarrollada, deberás enfocarte en contestar estas preguntas:

¿Por qué seleccionaron el modelo multiagentes utilizado? ¿Cuáles fueron las variables que se tomaron al momento de tomar la decisión? ¿Cuál es la interacción de esas variables con respecto al resultado de la simulación? ¿Por qué seleccionaron el diseño gráfico presentado? ¿Cuáles son las ventajas que encuentras en la solución final presentada? ¿Cuáles son las desventajas que existen en la solución presentada? ¿Qué modificaciones podrías hacer para reducir o eliminar las desventajas mencionadas?

Una reflexión corta sobre tu proceso de aprendizaje.

Arturo Ramos Martínez:

Aprender un nuevo enfoque de programación es un desafío que exige no solo adquirir nuevos conocimientos, sino también desaprender temporalmente lo que ya conocemos. Este proceso me ha permitido ampliar mi perspectiva sobre la programación, especialmente en el contexto de los modelos multiagente. Elegimos este modelo debido a su capacidad para simular de manera efectiva la interacción y coordinación entre agentes en tiempo real, lo cual era crucial

para nuestro proyecto. Al vincular estos modelos con simulaciones en Unity y observar cómo pequeños datos de sistemas reales como cámaras pueden influir significativamente, me hizo reflexionar sobre el impacto que puede tener la información en el éxito de un sistema. La idea de que cualquier mundo dinámico puede representarse en un mundo estático con el nivel de precisión que el desarrollador desee me inspira a seguir empujando los límites de lo que podemos lograr con la tecnología.

Adolfo Hernández Signoret:

Al aprender un nuevo paradigma de programación, siempre es difícil adaptarse y lleva un tiempo de estudio. Uno intenta siempre asociar cosas con procedimientos aprendidos previamente y entonces no solo debes de aprender cosas nuevas sino desaprender al menos momentáneamente algunas otras. En este proceso fue en el que más adquirí el conocimiento, cuando me dediqué a tratar de descifrar cómo funciona la programación de modelos de multiagentes. Me sirvió para ampliar mi manera de ver la programación. Además, el procedimiento de externar el modelo a una simulación con datos “reales” en la simulación de Unity, me hizo ver que los sistemas embedded (cómo las cámaras) pueden llegar a ser de una gran utilidad incluso si sólo alimentan al modelo con pequeños datos de información. Y que esta pequeña información puede ser utilizada para lograr grandes cosas. Mi frase favorita del libro de multiagentes es que, cualquier mundo dinámico se puede representar en un mundo estático al nivel de similitud que el desarrollador desee, esto abre las puertas a hacer sistemas que comprendan la realidad cada vez de una mejor manera en la que los límites son solo nuestros y de nuestras tecnologías. El hecho de que este proceso de mejora sea siempre eterno me genera ganas de empujar esos límites.

Bryan Ithan Landín Lara:

Durante el proyecto, elegimos un modelo multiagente basado en razonamiento deductivo porque nos resultaba familiar y permitía crear una lógica clara. Sin embargo, notamos que

esta simplicidad podría limitar la escalabilidad, especialmente si quisiéramos manejar más drones e intrusos. En el futuro, sería importante hacer la lógica más flexible. También aprendimos el valor de integrar sistemas reales, como cámaras, en la simulación. Aunque parecía que aportarían poca información, resultaron ser clave para mejorar la toma de decisiones de los agentes. Este proceso me recordó que siempre hay margen para seguir aprendiendo y superando límites.

Diego Enrique Vargas Ramírez:

Primero y primordialmente, escogimos el modelo de un agente con razonamiento deductivo debido a que con el que nos sentíamos mas familiarizados. Además, este modelo nos permite crear una logica compleja la cual se compone de muchas acciones regidas por reglas específicas. Esperando así crear una secuencia de acciones conforme el estado del agente cambiara. Si bien nuestro modelo usa una cantidad de agentes relativamente pequeña, dos de ellos se esperaba que tuvieran un diseño bastante complejo. Siendo estos el intruso y el dron. El hecho de que estos dos elementos de nuestra simulacion fueran “mas inteligentes” que los otros agentes del guardia y el dron hacia que nuestra simulacion fuera mas compleja, dando lo que consideramos mejores resultados. Esta solucion en el esquema de las cosas es sencilla pero poderosa a mi opinion, aunque esta sencillez a su vez puede ser un detrimento. Por ejemplo, que tal si quisieramos lidiar con mas drones e intrusos, como manejamos una sobrepoblación. Pienso que en una futura iteración de este proyecto se podria considerar mejorar esta logica. E incluso desligarla a un grupo tan extensivo de reglas.

Luis Fernando Cuevas Arroyo: Elegimos este modelo multiagente, debido a que se nos hizo el mejor para desarrollar en 5 semanas. Al mismo tiempo, seleccionamos el modelo multiagente por su capacidad para simular la interacción y coordinación entre agentes de seguridad en tiempo real, enfocándonos en variables como la cobertura de vigilancia y el tiempo de respuesta. Por otro lado, el diseño gráfico facilita la visualización de las rutas y decisiones de los agentes, con ventajas como la optimización de la vigilancia, aunque con la desventaja de ser menos adaptable a escenarios complejos. Por último, a lo largo del proyecto, aprendí a integrar tecnología multiagente para resolver problemas de seguridad y mejoré mis habilidades en la gestión de proyectos colaborativos. Aprendí sobre las propiedades de un agente, el ambiente y las bases de lo que es la inteligencia artificial.