



Tecnológico de Monterrey

Evidencia 2 : Entrega Intermedia

Jessica Odette Guizado Hernández A01643724

Santos Alejandro Arellano Olarte A01643742

Luz Patricia Hernández Ramírez A01637277

Carlos Alberto Mentado Reyes A01276065

Ángel Eduardo Esquivel Vega A01276114

Modelación de sistemas multiagentes con gráficas computacionales (Gpo 101)

Profesores:

Iván Axel Dounce Nava

Luis Raúl Guerrero Aguilar,

Obed Nehemías Muñoz Reynoso,

Carlos Johnnatan Sandoval Arrayga

Lunes 02 de septiembre, 2024

Evidencia 2: Entrega Intermedia

Conformación del equipo

Nuestro equipo está compuesto por los siguientes integrantes, cada uno con fortalezas únicas y áreas de oportunidad que contribuyen al desarrollo exitoso de nuestro proyecto:

1. Betito:

Fortalezas: Uso de AgentPy, con una sólida comprensión de la simulación de agentes. Posee habilidades técnicas que permitieron la implementación de modelos complejos.

Áreas de Oportunidad: Puede mejorar en la documentación detallada de sus procesos para facilitar la comprensión del equipo.

2. Santos:

Fortalezas: Gran habilidad en documentación y en proporcionar soporte técnico. Contribuyó significativamente en el desarrollo del código con AgentPy junto con Betito, y asegura que el código esté bien documentado para futuras referencias.

Áreas de Oportunidad: Podría fortalecer sus habilidades en la creación de diagramas y representaciones visuales de sistemas complejos.

3. Ángel:

Fortalezas: Creó los diagramas UML, tanto los de clase como los de protocolos de interacción. Su capacidad para visualizar y estructurar sistemas complejos es crucial para el entendimiento global del proyecto.

Áreas de Oportunidad: Podría mejorar en la implementación práctica de las soluciones técnicas que diseña, especialmente en programación.

4. Jess:

Fortalezas: Habilidad avanzada en la integración de sistemas, especialmente en la conexión entre Unity y Python. Logró la difícil tarea de mover objetos de Unity utilizando Python, lo que fue esencial para el proyecto.

Áreas de Oportunidad: Podría enfocarse en desarrollar más habilidades en la parte documental y en la explicación de sus procesos para que otros puedan replicar su trabajo con mayor facilidad.

5. Luz:

Fortalezas: Fuerte en la implementación de conexiones entre plataformas, trabajando en conjunto con Jess en la integración de Unity y Python. Su capacidad para colaborar y resolver problemas técnicos es una gran ventaja para el equipo.

Áreas de Oportunidad: Puede mejorar en la creación de diagramas y en la visualización de las interacciones entre sistemas para complementar su enfoque técnico.

Objetivos y Compromisos del Equipo

Lo que esperamos lograr y obtener como equipo:

Desarrollar una solución integrada donde todos los módulos (simulación, comunicación y respuesta) funcionen sincrónicamente con el modelo de visión.

Compromisos para lograrlo:

- Comunicación Continua: Mantener reuniones regulares.
- Colaboración: Aprovechar las fortalezas de cada miembro del equipo para apoyarnos mutuamente en nuestras áreas de oportunidad.
- Flexibilidad: Estar dispuestos a aprender y asumir nuevas tareas cuando sea necesario para el éxito del proyecto.

Con estos compromisos, estamos seguros de que alcanzaremos nuestros objetivos en este bloque y desarrollaremos una solución robusta y eficiente.

Propuesta Formal del Reto

Descripción del Problema

El objetivo principal del proyecto es desarrollar un sistema multiagente en un entorno de almacén o bodega, donde un dron y varios otros agentes (cámaras y personal de seguridad) trabajan juntos para mantener la seguridad del espacio. Este entorno simulado deberá permitir la interacción entre los diferentes agentes, optimizando la detección y respuesta ante situaciones sospechosas.

El proyecto contempla:

1. **Patrullaje del dron:** El dron despegará desde su estación y realizará un patrullaje rutinario por el almacén, siguiendo una ruta predeterminada y aterrizando de nuevo en su base una vez completado el recorrido.
2. **Detección de amenazas:** Las cámaras fijas ubicadas en diferentes puntos estratégicos del almacén estarán encargadas de detectar movimientos o figuras sospechosas. Si se detecta algo inusual, las cámaras notificarán al dron para que investigue.
3. **Inspección y respuesta:** Una vez notificado, el dron se dirigirá a la zona sospechosa para realizar una inspección más detallada. Si el dron detecta una amenaza, pasará el control al personal de seguridad, quienes decidirán la acción a seguir, como activar una alarma o desactivar una falsa alarma.

Identificación de los Agentes Involucrados:

- **Dron:** Responsable del patrullaje, detección de anomalías y comunicación con el personal de seguridad.
- **Cámaras Fijas:** Encargadas de monitorear el entorno del almacén, detectando movimientos o figuras sospechosas.

- **Personal de Seguridad:** Controla el dron en caso de amenaza y toma decisiones críticas para la seguridad del almacén.

Objetivos del Equipo:

1. **Desarrollar un entorno de simulación 3D altamente eficiente y realista**, donde todos los agentes interactúen de manera coherente y sincrónica.
2. **Integrar y coordinar eficazmente los diferentes agentes involucrados en el proyecto**, asegurando que cada uno cumpla su función dentro del sistema de seguridad del almacén.
3. **Asegurar que el sistema de seguridad del almacén funcione de manera óptima con el dron y el personal de vigilancia**, detectando y respondiendo a amenazas potenciales de manera eficiente.
4. **Implementar un sistema de comunicación robusto y eficiente entre los agentes**, priorizando la rapidez y la exactitud en la transmisión de información crítica.
5. **Documentar cada fase del proyecto**, asegurando que todas las decisiones y cambios sean registrados para futuras referencias y evaluaciones.

Diagramas de Clase

DronAgent
agentType : int currentPlan : List [Tuple [int, int]] currentPosition : Tuple [int, int] isPatrolling : bool detectedSuspicion : bool connectedToSecurity : bool controlledBySecurity : bool
setup(self) : None step(self) : None update(self) : None end(self) : None
brf(self, p : List[Agent]) : None brf_second(self, g : List[Agent]) : None brf_third(self, g : List[Agent]) : None options(self) : Dict[SecurityAgent, float] options_second(self) : Dict[TresspasserAgent, float] filter(self) : SecurityAgent or None filter_second(self) : TresspasserAgent or None plan(self) : List [Tuple[int, int]] plan_second(self) : List [Tuple[int, int]] BDI(self, p : List[Agent]) : None BDI_second(self, p : List[Agent]) : None execute(self) : None initBeliefs(self, initPos : Tuple[int, int]) : None initIntentions(self) : None takeOff(self) : Bool returnToBase(self) : Bool patrolArea(self) : Bool moveToArea(self) : Bool land(self) : Bool givePosition() : None requestPosition() : Tuple [int, int] confirmAlert() : Bool confirmAlertCamera() : Bool alertSecurity() : None
see(self, e : Environment) : List[SecurityAgent] see_second(self, e : Environment) : List[TresspasserAgent]
Capacidad para realizar patrullaje de áreas designadas; detección y análisis de actividades sospechosas; comunicación con el personal de seguridad; capacidad de ser controlado por el personal de seguridad.
El agente es responsable de patrullar un área, detectar actividades sospechosas, y coordinarse con el personal de seguridad para responder a potenciales amenazas
Protocolo de Comunicación "SecurityAgent - DronAgent" Protocolo de Comunicación "CameraAgent - DronAgent"
Drons

CameraAgent
agentType : int position : Tuple [int, int] isMonitoring : bool detectedMovement : bool trespasserDetected : bool controlledBySecurity : bool
setup(self) : None step(self) : None update(self) : None end(self) : None
brf(self, p : List[Agent]) : None brf_second(self, g : List[Agent]) : None brf_third(self, g : List[Agent]) : None options(self) : Dict[DronAgent, float] options_second(self) : Dict[TrespasserAgent, float] options_third(self) : Dict[SecurityAgent, float] filter(self) : DronAgent or None filter_second(self) : TrespasserAgent or None filter_third(self) : SecurityAgent or None plan(self) : List [Tuple[int, int]] plan_second(self) : List [Tuple[int, int]] plan_third(self) : List [Tuple[int, int]] BDI(self, p : List[Agent]) : None BDI_second(self, p : List[Agent]) : None BDI_third(self, p : List[Agent]) : None execute(self) : None initBeliefs(self, initPos : Tuple[int, int]) : None initIntentions(self) : None detectMovement() : bool alertDrone() : None givePosition() : None
see(self, e : Environment) : List[DronAgent] see_second(self, e : Environment) : List[TrespasserAgent] see_third(self, e : Environment) : List[SecurityAgent]
Capacidad para monitorear constantemente un área designada, detección de actividades sospechosas, comunicación con el DronAgent.
El agente es responsable de monitorear un área específica, detectar y analizar movimientos, y colaborar con el dron para la inspección de actividades sospechosas, proporcionando video en vivo cuando sea necesario.
Protocolo de Comunicación "CameraAgent - DronAgent"
Cameras

SecurityAgent
agentType : int name : string position : Tuple [int, int] isOnDuty : bool isControllingDron : bool linkedDron : DronAgent
setup(self) : None step(self) : None update(self) : None end(self) : None
brf(self, p : List[Agent]) : None brf_second(self, g : List[Agent]) : None brf_third(self, g : List[Agent]) : None options(self) : Dict[DronAgent, float] options_second(self) : Dict[TresspasserAgent, float] filter(self) : DronAgent or None filter_second(self) : TresspasserAgent or None plan(self) : List [Tuple[int, int]] plan_second(self) : List [Tuple[int, int]] BDI(self, p : List[Agent]) : None BDI_second(self, p : List[Agent]) : None execute(self) : None initBeliefs(self, initPos : Tuple[int, int]) : None initIntentions(self) : None controlDron() : None connectedToDron() : Bool requestPosition() : Tuple[int, int] confirmAlert() : Bool endControl() : None
see(self, e : Environment) : List[DronAgent] see_second(self, e : Environment) : List[TresspasserAgent]
Capacidad de monitorear las operaciones de Camaras y Drons; responder alertas y controlar el dron cuando sea necesario.
El agente es responsable de supervisar la seguridad de las instalaciones, respondiendo a alertas, controlando el dron cuando sea necesario, y tomando decisiones sobre la activación de alarmas.
Protocolo de Comunicación "SecurityAgent - DronAgent"
SecurityCrew

TresspasserAgent
agentType : int position : Tuple[int, int]
setup(self) : None step(self) : None update(self) : None end(self) : None
move() : None
El agente representa a un intruso en la simulación, este unicamente tendrá movimiento aleatorio.
Tresspassers

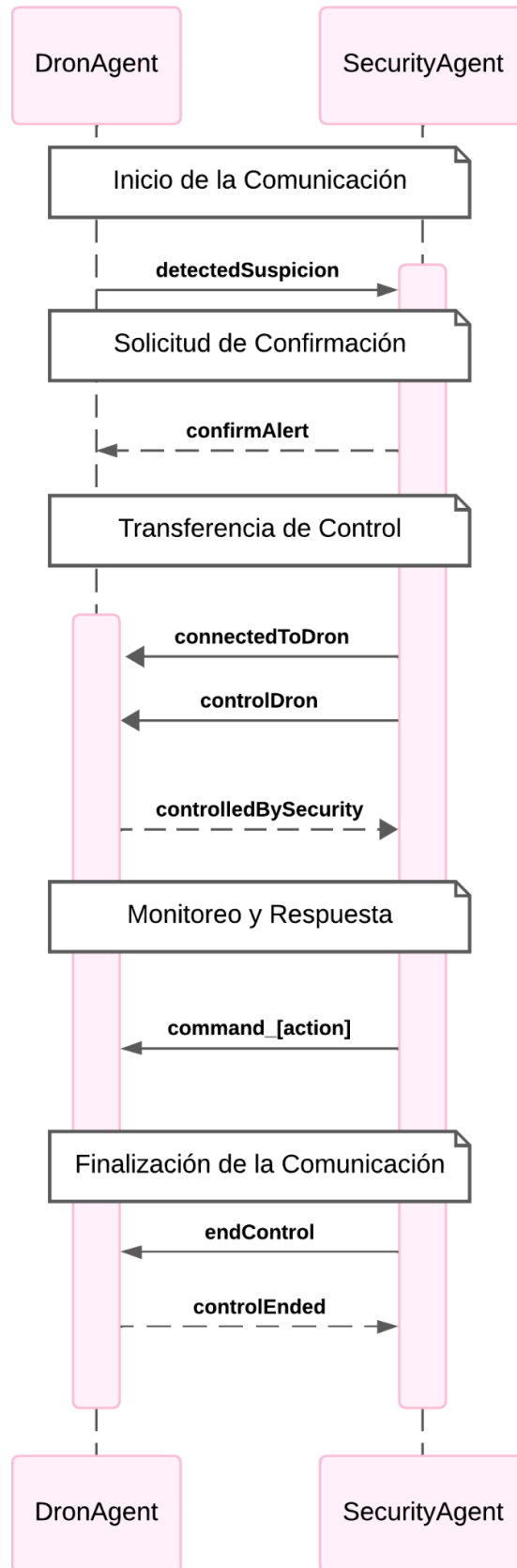
Medición de la Utilidad o Éxito

El éxito del sistema se medirá a través de los siguientes criterios:

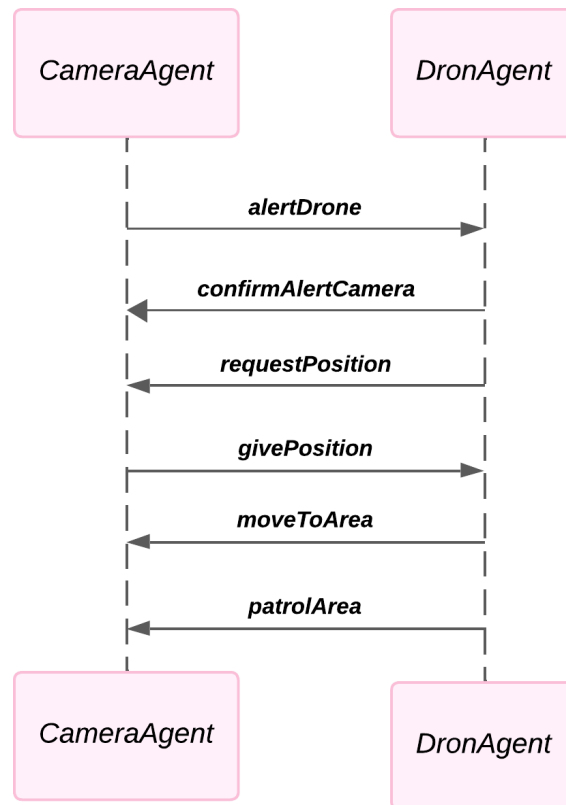
- **Tasa de detección de amenazas:** Proporción de amenazas detectadas y manejadas correctamente por el sistema.
- **Eficiencia del patrullaje:** Tiempo que tarda el dron en realizar su patrullaje completo y en responder a una notificación de amenaza.
- **Interacción entre agentes:** Evaluación de la eficacia y velocidad de la comunicación y coordinación entre el dron, cámaras y personal de seguridad.
- **Tiempo de respuesta:** Velocidad con la que el personal de seguridad puede tomar el control del dron y responder a una amenaza detectada.
- **Precisión de la detección:** Evaluación de la tasa de falsas alarmas generadas por el sistema.

Diagramas de Protocolos de Interacción

Protocolo de Comunicación "SecurityAgent - DronAgent"



Protocolo de Comunicación "CameraAgent - DronAgent"



Plan de Trabajo

Jira

Tu trabajo ▾

Proyectos ▾

Filtros ▾

Paneles ▾

Equipos ▾

Planes ▾

Aplicaciones ▾

Crear

Mejorar

C

📄

drones

Proyecto de software

PLANIFICACIÓN

Cronograma

Backlog

Tablero

Objetivos

+ Añadir vista

DESARROLLO

Código

Páginas del proyec...

Configuración del pro...

Proyectos / drones

Evidencia 2

🔍 Buscar

LH

CR

JH

👤

+2

👥

AGRUPAR POR Nada ▾

🔄 Importar t

POR HACER 3

implementar modelo de vision

✓ SCRUM-11

LH

lógica de patrulleo drones

✓ SCRUM-12

CR

análisis de la solución

✓ SCRUM-13

👤

+ Crear incidencia

EN CURSO 3

generar cambios en script Agentpy para simulación con dron

✓ SCRUM-2

CR

modificar script csharp

✓ SCRUM-3

JH

terminar diagramas

✓ SCRUM-7

👤

LISTO 4 ✓

agregar dron asset

✓ SCRUM-4 ✓

LH

terminar escena unity

✓ SCRUM-6 ✓

JH

Definición de steps y mejorar interacción entre agentes

✓ SCRUM-8 ✓

👤

utilidad y éxito

✓ SCRUM-9 ✓

LH

Repositorio de GitHub: <https://github.com/Santos-Arellano/Evidencia-Agentes/tree/main>