“Revisión de avance 2"

Instituto tecnológico y de estudios superiores de Monterrey

Campus Monterrey

TC2008B

Grupo 302

Modelación de sistemas multiagentes con gráficas computacionales

Profesores:

Raul V. Ramirez Velarde

Luis Alberto Muñoz Ubando

Equipo 3:

Ainhize Legarreta García	A01762291
Enrique Uzquiano Puras	A01762083
Christopher Pedraza Pohlenz	A01177767
David Cavazos Wolberg	A01721909
Fausto Pacheco Martínez	A01412004

ÍNDICE

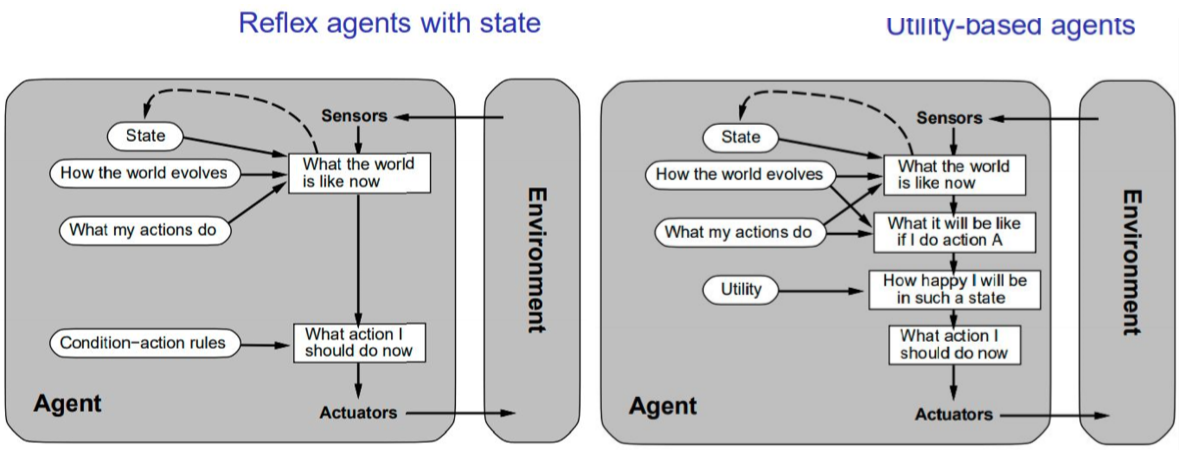
1. DESCRIPCIÓN DEL RETO

El reto que se nos presenta está relacionado con el entorno del campo y la agricultura, donde se nos dice que "*Optimizar las operaciones agrícolas a través de un sistema inteligente no sólo mejora la eficiencia en la producción de alimentos, sino que también puede reducir costos, lo que podría traducirse en precios más bajos para los consumidores. Además, al tener una logística más eficiente, disminuimos la congestión en las carreteras, lo que beneficia a todos los que viven y trabajan en áreas urbanas.*". Por lo anteriormente mencionado, el equipo ha desarrollado la propuesta de crear una simulación de un sistema multiagente donde se tendrá un campo de dimensiones variables, donde una serie de tractores y cosechadoras interactúan entre sí, buscando minimizar el tiempo de cosecha (lo que se traduce a la reducción de costos) buscando la ruta más óptima.

2. AGENTES INVOLUCRADOS

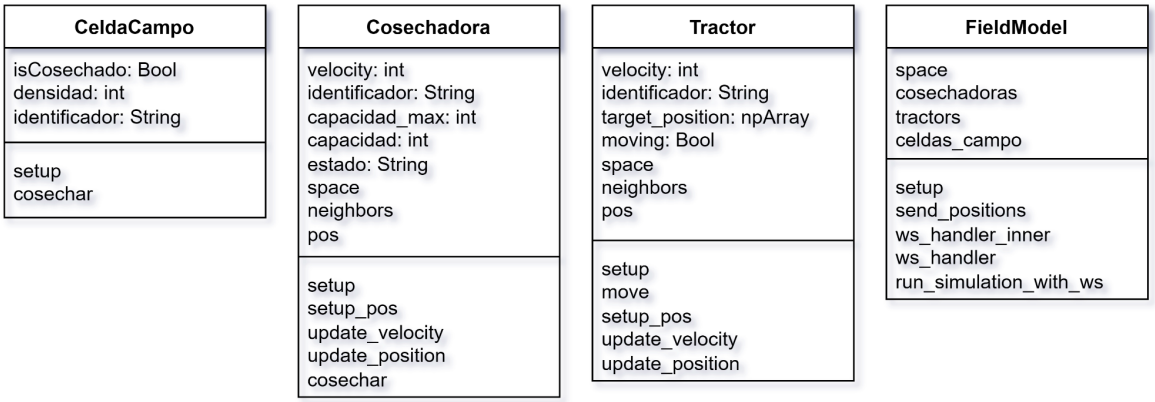
El sistema multiagente consta de 3 agentes: cosechadora, tractor y celda de campo. El agente "cosechadora" se encargará de ir cosechando el campo hasta llenar su capacidad máxima. Una vez que su capacidad se llena, un agente "tractor" va en su dirección para descargar el contenido de la cosechadora. Luego de descargar el contenido, el tractor vuelve a su posición de reposo esperando a que otra cosechadora se llene para volver a descargarla. Asimismo, una vez que la cosechadora ha sido descargada, continúa cosechando las celdas de campo. Los agentes "celda campo" se encargan de llevar registro de las partes del campo que ya han sido cosechadas y la densidad de trigo que hay en cada una, para que cuando una cosechadora pase sobre ellas, estas le transmitan la densidad que está cosechando.

Por el momento, se tiene 2 tipos de agentes: Agentes reflexivos con estados, y Agentes basados en utilidad. Siendo la celda de campo y el tractor el reflexivo con estados, y la cosechadora basada en utilidad. Esto se debe a que tanto el tractor como la celda cuentan con estados que cambian su comportamiento, y reaccionan a cambios del ambiente, sin embargo, no hacen sus acciones con un objetivo o para conseguir una utilidad. En cambio, la cosechadora tiene el objetivo de cosechar todo el campo (objetivo) en el menor tiempo (utilidad). A continuación dos diagrama describiendo a los agentes:



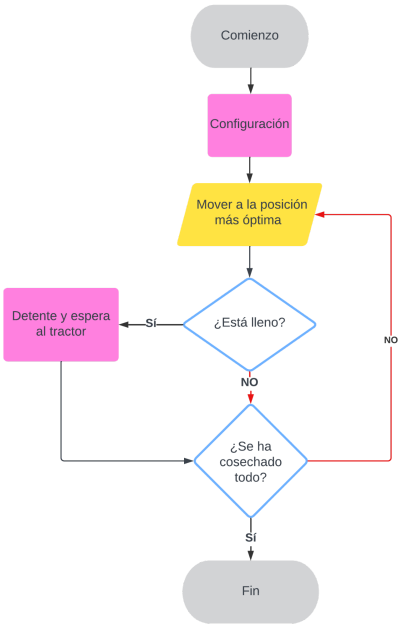
Además, los agentes interactúan entre sí por medio del cambio de estados. En el caso de la celda campo, inicialmente comienza con el estado de isCosechado como falso, significando que cuando una cosechadora pase por encima, podrá cosechar la densidad de esa celda. Tras esto, su estado cambiaría a verdadero para que no pueda ser cosechada otra vez. En el caso de la cosechadora, cuenta con 3 estados: cosechando, llena, y esperando. Cuando tiene todavía capacidad, su estado es cosechando, lo que le permite seguir moviéndose y cosechando las celdas por las que pase. Sin embargo, una vez que llega a su capacidad máxima, esta cambia al estado de llena. Aquí es donde se da la interacción con los tractores. Estos en todo momento buscan alguna cosechadora con estado lleno, mientras tanto, permanecen en su estado de reposo. Una vez que detectan una cosechadora llena, cambian el estado de la cosechadora a esperando, su estado propio a movimiento, y se empiezan a dirigir a la cosechadora para descargarla. El cambio de estado a esperando es una manera de decirle a otros tractores que esa cosechadora ya tiene un tractor asignado para ser descargada, por lo que no van a ir múltiples tractores a descargarla.

Diagramas de clase

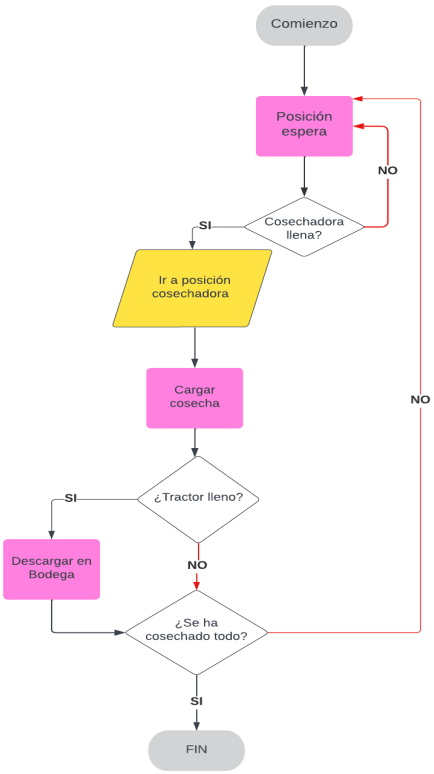


Diagramas de protocolos de interacción

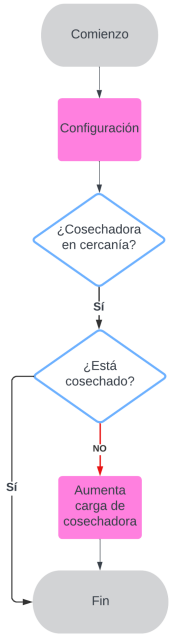
Cosechadora



Tractor



Campo



3. VIDEO DE LA SIMULACIÓN

https://youtu.be/0q4Wyd9vReg

4. IMPLEMENTACIÓN DE Q-LEARNING

El algoritmo Q-learning es una técnica fundamental en el campo del aprendizaje por refuerzo dentro del ámbito de la inteligencia artificial. Se utiliza para que una inteligencia artificial (IA) aprenda a tomar decisiones óptimas en un entorno basado en la teoría de recompensas y acciones. Nos hemos decidido por este algoritmo para optimizar la toma de decisiones de la cosechadora y así poder reducir el tiempo de cosecha, se le ha asignado una recompensa a cada una de las posibles acciones que el agente puede tomar siendo estas recompensas peores cuando menos deseable sea esa acción, obteniendo recompensas desde -100 en el caso menos deseable a -1 en el caso más deseable. El algoritmo se ejecutará una serie de episodios para poder mejorar su ruta y su toma de decisiones obteniendo así un Q value cada vez menor y una mejor solución.

Celda fuera del grid – reward = -100

Chocar con otros objetos – reward = -100

Campo opuesto – reward = -50

Celda cosechada – reard = -30

Celda sin cosechar – reward = -1

5. PLAN DE TRABAJO Y APRENDIZAJES ADQUIRIDOS

Planeación de Actividades

Actividad	Esfuerzo	Fecha estimada	Encargado
Crear/conseguir el asset para la cosechadora	M	11/20/2023	David
Crear/conseguir el asset para el tractor	M	11/20/2023	David

Crear/conseguir el asset para el trigo	M	11/20/2023	David
Integrar Q-Learning en el movimiento de las cosechadoras	M	11/20/2023	TODOS
Modelar una vista en primera persona para los tractores/cosechadoras	L	11/27/2023	Christopher
Llevar un contador del trigo cosechado	XS	11/20/2023	Christopher
Integrar obstáculos en el campo y el comportamiento para evitarlos	L	11/23/2023	Fausto
~~Conectar Unity con AgentPy usando un websocket~~	XL	11/14/2023	Christopher
~~Identificar las cosechadoras llenas y mandar un tractor a descargarlas~~	M	11/14/2023	TODOS
Pasar todos los datos necesarios de la simulación a Unity	M	11/20/2023	Christopher

6. REPOSITORIO DE GITHUB

El siguiente enlace abre el Github del equipo:
https://github.com/christopher-pedraza/multiagentes_john_deere

Como herramienta de comunicación entre el equipo, se cuenta con un grupo de Whatsapp por el cual nos organizamos y discutimos las ideas y propuestas.