



Tecnológico de Monterrey

Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Monterrey

Modelación de sistemas multiagentes con gráficas computacionales (Gpo 302)

Profesor:

Luis Alberto Muñoz Ubando

Raúl Valente Ramírez Velarde

Avance 3

Integrantes:

Alonso Abimael Morales Reyna A01284747

Marco Ottavio Podesta Vezzali A00833604

Ernesto Poisot Avila A01734765

Sergio Ortiz Malpica A01284951

Equipo 4

Fecha de entrega: 28/11/2023

Introducción

Durante este bloque nosotros aprendimos sobre la modelación de sistemas multiagentes para simular situaciones y procesos reales, con el propósito de encontrar una forma de optimizarlos, debido a que llegar a optimizar el resultado de procesos es de interés para muchas empresas y organizaciones debido a que les permite ah

Descripción del Reto

La industria de la agricultura es esencial para la subsistencia de la sociedad, debido a que los cultivos que genera son una fuente importante de alimentos. Sin embargo, los procesos de recolección de cultivos pueden ser complicados, haciendo que los procesos cuesten más y sean menos productivos, por lo que nuestro socio formador John Deere nos pidió crear una simulación de la cosecha de trigo con el fin de encontrar la forma más eficiente de cosechar este cultivo. De esta forma no solo se reducen los costos de recursos como la gasolina, también se reduce la cantidad de dióxido de carbono generado al reducir el tiempo que se utilizan los tractores.

Agentes Identificados

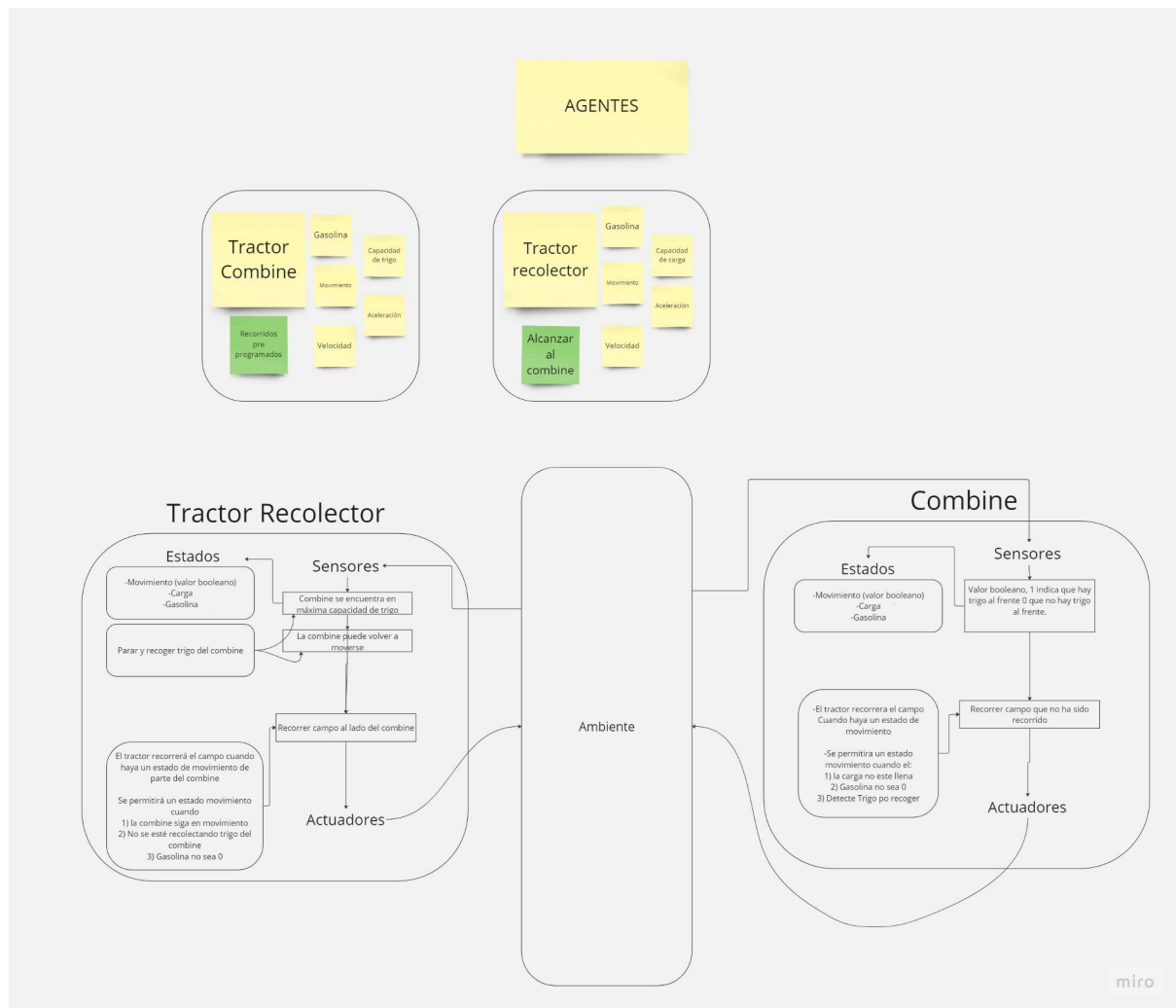
Cosechadora

El agente recolecta trigo y es capaz de seguir un camino para recolectar el trigo que se encuentre en el camino. El agente tiene un límite de capacidad de trigo y de gasolina. Cada paso que hace, consume gasolina y cada trigo que recoge, es acumulado en su capacidad de carga. Si se llega a una capacidad de carga mayor al 75%, o su capacidad de gasolina se reduce del 25%, este se detiene para esperar al tractor recolector, ya sea para que recolecta su trigo o recargue su gasolina.

Tractor Recolector

El agente representa un tractor que recolecta el trigo que ha procesado la cosechadora. Además, ofrece la posibilidad de descargar el combine y rellenar su gasolina. El agente tiene un límite de capacidad de almacenamiento de trigo y gasolina. Cada paso que hace, consume gasolina. Si se llega a una capacidad de carga mayor al 75%, o su capacidad de gasolina se reduce del 25%, este se detiene para esperar al tractor recolector. El agente está en un monitoreo constante de los estados de la cosechadora, si determina que la cosechadora necesita ayuda, este va a ir a hacer las acciones necesarias (descargar trigo, rellenar gasolina).

Diagrama de clase y protocolo de interacción entre agentes



Pantallas

Explicación de la solución

En nuestro proyecto, creamos un código en Python que tiene la lógica del movimiento de los agentes, los cuáles son la cosechadora y el tractor recolector. El código controla el comportamiento de los agentes y dicta qué movimientos deben de tomar. Para esto se toma en cuenta el ambiente que será el campo, su tamaño y las recompensas que ofrece dependiendo de la acción que el agente realice.

En Unity, simulamos un campo de una granja el cuál es un espacio cuadrado que se encuentra rodeado por árboles y en el centro se encuentra un campo de trigo que se genera una vez que se inicia el programa. El tamaño del campo puede ser programado desde Unity.

Unity utiliza exclusivamente C# para los scripts que dictan el comportamiento de objetos, por lo que unimos la lógica de Python con Unity utilizando una conexión cliente-servidor mediante una api de javascript, que detecta el resultado que se obtenga del código de la simulación de python, enviándolo al cliente de unity.

Código de implementación entre agentes

Código de implementación gráfica

Plan de trabajo

Actividades Pendientes

Mejora de gráficos en Unity - 2 días

Aumentar complejidad/efectividad del algoritmo - 2 días

- **Para las actividades planeadas para la primera revisión, los responsables de llevarlas a cabo, la fecha en las que las realizarán y el intervalo de esfuerzo estimado.**

Actividades Completadas

1. Identificación de agentes - 1 día

2. Planeación de la solución - 2 días

3. Creación de primer algoritmo para la solución del reto - 3 días

4. Creación del ambiente en Unity - 8 días

5. Conexión entre Python y Unity - 3 días

Aprendizaje adquirido

Como equipo tuvimos la oportunidad de aprender sobre el modelado de sistemas multiagentes, al igual que de su funcionamiento. Al trabajar en este proyecto también mejoramos nuestro conocimiento de fundamentos matemáticos para gráficas, el cuál nos fue de utilidad para generar el modelado en Unity. También tuvimos la oportunidad de aprender más de machine learning y aplicarlo para intentar optimizar un proceso real, al igual que ver cómo es el desarrollo profesional de software en base a lo que nos pide un socio.

Alonso

Aprender de los sistemas multiagentes me ayudó a pensar más en las interacciones entre agentes en un sistema, ya que el comportamiento de un agente en un ambiente puede cambiar el comportamiento de otros agentes de formas inesperadas. Me pareció muy interesante poder utilizar machine learning para

resolver problemas reales y ver cómo se comporta el modelo después de entrenarlo. Siento que lo que aprendí en esta materia será importante para mi carrera.

Ernesto

La perspectiva única que se obtiene a base de trabajar en un sistema de múltiples agentes ha cambiado mi forma de solucionar problemas al abrirse nuevas puertas de conocimiento. Comprender el paradigma de resolución de problemas teniendo múltiples agentes en un ambiente que interactúen con este y eso traiga cambio a lo que se trabaje es toda una fuente de conocimiento que al utilizar herramientas como agent.py, unity, javascript y demás se puede dar soluciones que se implementen igual en la vida real

Marco

El trabajar en el desarrollo del algoritmo me ha ayudado a obtener un mayor conocimiento sobre el funcionamiento de Machine Learning y sus usos aplicados en la vida real; el poder ver los efectos directamente en la simulación ha sido una experiencia de aprendizaje sumamente valiosa para refinar mis habilidades de programación e implementación de algoritmos.

Sergio

La modelación de agentes para la realización de una simulación definitivamente es una forma nueva que no conocía antes de atacar un problema. En nuestro caso teníamos que ser capaces de simular un entorno en el que a su vez pudiéramos resolver una eficiente manera de recolectar el trigo. A lo largo de este proyecto se fue posible poner en práctica lo aprendido en la teoría en cuanto a machine learning y modelación de estructuras de agentes para nuestra simulación, y soy de los que piensan que la teoría se refuerza mucho mejor en la práctica, por lo que fui nutritivo aplicar la teoría a la práctica a través de este proyecto.

Repositorio en Github

<https://github.com/marcopod/multiagentes-TC2008B>

Video del Proyecto

<https://youtu.be/7Keh9mlzYHk>