Tecnológico de Monterrey

Reporte grupal

Hiram Israel Mendoza López A01613963

Santiago Yael Nieto Castillo A01612348

Gonzalo Morán Sánchez A01710494

Modelación de sistemas multiagentes con gráficas computacionales (Gpo 102)

Alejandro Fernández Vilchis

Denisse Lizbeth Maldonado Flores

Pedro Oscar Pérez Murueta

Fecha de entrega:
5 de septiembre del 2024

Tecnológico de Monterrey

Introducción

Narrativa:

Había una vez una leyenda: la casa de la calle "Armenia" estaba embrujada. Y, ¿saben qué? Tenían razón. En esta casa habitaban fantasmas que, desde hace mucho tiempo, la vigilaban para protegerla de cualquier intruso. Un día, uno de los fantasmas más jóvenes, mientras conectaba algunas decoraciones, provocó un cortocircuito que desató un incendio en la casa.

Los fantasmas, asustados y desesperados, gritaban en busca de ayuda, pero no tenían un teléfono para llamar a los bomberos. En ese preciso momento, la familia Pacman, que vivía al lado, escuchó los gritos y, sin pensarlo dos veces, acudió al rescate.

Antecedentes, Objetivos y Alcance

El proyecto consiste en el desarrollo de una simulación multiagente inspirada en el juego de mesa Flash Point: Fire Rescue.

El objetivo principal de esta simulación es recrear un escenario en el que seis bomberos deben cooperar para rescatar a las víctimas atrapadas en una casa mientras el fuego se propaga. A través del desarrollo de un modelo computacional, buscamos simular con precisión el comportamiento autónomo y coordinado de los agentes, implementando diversas estrategias para optimizar el rescate antes de que el fuego cause un colapso en la estructura.

El alcance de este proyecto incluye la creación de un tablero de simulación de 6x8 celdas, la implementación de agentes con capacidades como moverse, apagar incendios y rescatar víctimas, y la visualización en un entorno de Unity. La simulación sigue las reglas establecidas en el juego

original bajo el Family Game Setup, y considera tanto las dinámicas del fuego como las acciones de los bomberos.

Los objetivos son desarrollar habilidades en la implementación de algoritmos de sistemas multiagente, la visualización en 3D, y la evaluación del rendimiento del modelo, elementos esenciales para las conclusiones que se derivarán de esta experiencia

Metodología

Clase Building

La clase Building es el núcleo del modelo y representa el entorno de la casa donde ocurre la simulación. Hereda de la clase Model de la biblioteca Mesa, una herramienta que permite crear simulaciones de sistemas multiagentes.

Matriz de paredes y fuego: El edificio se modela mediante una matriz de celdas, donde las paredes y el fuego se representan con valores numéricos. Por ejemplo, las paredes están indicadas por un valor de 3, mientras que el fuego en diferentes niveles se representa por valores de 9 o 10. Los objetos y agentes (como puertas, entradas y bomberos) están integrados en esta matriz.

Agentes bomberos: Los bomberos son instancias de la clase Bombero. Son ubicados inicialmente en las entradas del edificio y su objetivo es moverse por el espacio para salvar víctimas y extinguir el fuego . El estado y las acciones de los agentes se gestionan a través de una clase Fila que permite llevar un registro de los agentes vivos y su turno.

Algoritmo de generación de paredes y fuego

El espacio del edificio se divide en celdas y, utilizando ciclos anidados, el código genera paredes verticales y horizontales. Las puertas y entradas también se definen en este proceso.
Para los fuegos, se utilizan posiciones predefinidas (almacenadas en la lista fuegos). Cuando el fuego avanza, un número aleatorio genera la siguiente posición en la que el fuego se propagará. Además, si una pared entre celdas está en llamas, su estado se modifica (se debilita o es destruida) conforme avanza el fuego.

Clase Bombero

La clase Bombero define las capacidades y restricciones de los bomberos dentro del edificio. Cada bombero puede moverse entre celdas adyacentes, interactuar con puertas (abrirlas o cerrarlas) y derribar paredes para llegar a áreas en llamas. Además, los bomberos tienen puntos de acción (action_points), que limitan cuántas acciones pueden realizar en cada turno.

Acciones principales: Los bomberos pueden realizar diferentes tipos de acciones: moverse, abrir puertas y derribar paredes. Dependiendo de las situaciones que enfrenten (como la cercanía de una puerta o una pared), elegirán entre realizar una acción u otra. La cantidad de puntos de acción también condiciona las acciones disponibles.

Interacción con el fuego y las paredes: Un aspecto interesante es cómo los agentes interactúan con las paredes y el fuego. Cuando se encuentran con paredes, pueden optar por derribarlas si es necesario para avanzar. Asimismo, pueden verse afectados por el avance del fuego.

Clase Fila y el manejo de agentes

El manejo de los agentes se realiza con la clase Fila, que utiliza una estructura circular para mantener el ciclo de los turnos. Cada bombero es representado por un nodo dentro de esta fila. Esto permite alternar entre los bomberos y asegurar que cada agente actúe de manera equitativa. La estructura circular asegura que, una vez que todos los agentes han actuado, el ciclo se reinicia.

Estructura circular: Los bomberos son añadidos a la fila mediante el método push. Al finalizar el turno de un bombero, la referencia al siguiente agente es actualizada, garantizando la rotación en los turnos. Si un bombero muere, el método pop se encarga de eliminarlo de la fila, manteniendo la consistencia en la rotación.

Metodología del avance del fuego

El avance del fuego en la simulación consiste en que se elige aleatoriamente una celda de la casa en cada turno para encenderla o intensificar su nivel de incendio y explotar. Esta metodología se basa en la probabilidad y la simulación de eventos aleatorios. Cuando el fuego alcanza una pared, ésta puede debilitarse y eventualmente destruirse, permitiendo que el fuego avance hacia otras áreas del edificio. El fuego también puede propagarse vertical y horizontalmente, afectando múltiples celdas cercanas.

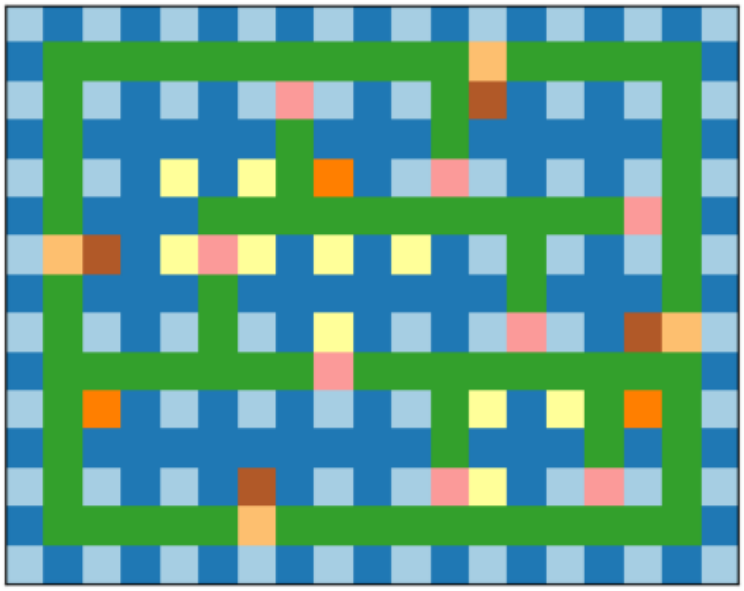
Recolección de datos y visualización del estado

El código incluye un recolector de datos (DataCollector) que captura el estado de la casa en cada paso de la simulación. La función get_grid recopila las posiciones del fuego, los agentes y las paredes para crear una representación visual en forma de una matriz. Esta matriz se actualiza con cada paso del modelo y permite generar visualizaciones o análisis del estado actual del edificio.

Resultados

Se logró realizar un modelado en python que recrea la solución que debería ser mostrada en unity. El problema que tuvimos fue que los datos que se mandan de python a el código en Unity no sé convertían correctamente a Json y no sabía como interpretarlos C#, por lo que Unity simplemente imprimía el grid de la tabla de python y no la simulación correctamente.

Modelado en Unity;



Nomenclatura:

Celdas Azul Fuerte: Es la cuadrícula del modelado (en este apartado el agente no interactúa con el tablero.

Celdas Azul Claro: El agente se puede mover libremente por esta sección

Celdas Verdes: Son las paredes de la simulación, si la pared es dañada se convierte en un color más claro.

Celdas Amarillas: Es interpretado como el fuego, cada turno que pasa se coloca de un color diferente, cuenta con todas sus interacciones, explosiones, humos que se convierten en fuego, etc.

Celdas Moradas: Son los humos que se van agregando cada ronda

Celdas cafés: Son los agentes(bomberos) que aparecen aleatoriamente en las entradas.

Celdas Rosas: Son las puertas que al ser abiertas por un bombero cambian de color para especificar que la puerta esta abierta.

Boceto en Unity: