



Instituto Tecnológico de Estudios Superiores Monterrey

CAMPUS QUERÉTARO

Modelación de sistemas multiagentes con gráficas computacionales

Denisse Lizbeth Maldonado Flores

Pedro Oscar Pérez Murueta

Alejandro Fernández Vilchis

TC2008B Grupo 102

**Reto
Proyecto WALL-E**

PRESENTAN

Jose Armando Rosas Balderas

A01704132

Ramona Nájera Fuentes

A01423596

Fecha:
06/09/2023

Idea de Solución

Etapla 0: Modelado del ambiente

En esta etapa definimos el tipo de mapa que íbamos a utilizar así como el manejo de tiempo y el tipo de agentes.

Para los agentes, optamos por crear 4 tipos de agentes: robot, basura, obstáculo y papelera.

Para el modelo decidimos usar un ambiente de tipo “Multigrid” para poder albergar a más de un agente por celda y utilizamos un manejo de tiempo de tipo “BaseSchedule” para evitar que dos robots o más robots tomen una decisión al mismo tiempo y haya conflictos.

Etapla 1: Exploración

Para la etapa de exploración decidimos dividir el mapa entre la cantidad de robots y que individualmente cada robot explorará esa parte del área. La división del mapa hace que la exploración sea más eficiente al evitar pasar por la misma celda muchas veces, además de reducir el mapa para cada robot. Para la exploración programamos una búsqueda en profundidad y terminamos hasta haber recorrido todas las celdas de su área.

Es importante mencionar que el número de robots varía dependiendo del ancho del mapa porque, si sus dimensiones son muy pequeñas, los obstáculos suelen interferir con la exploración de un área.

Etapla 2: Recolección

A lo largo de esta fase, los agentes trabajaron de forma cooperativa para deshacerse de la basura existente. Para lograrlo, dentro de cada step se buscaba la basura más cercana a cada uno de ellos (usando una bfs), posteriormente se calculaba el camino más corto que llevara al objetivo (con dijkstra) y se avanzaba un paso hacia este trayecto.

Dado que la tarea era compartida, cada iteración se recalcula la ruta a seguir para evitar conflictos en caso de haber tenido el mismo objetivo y, si no contaban con más almacenamiento o ya no había basura por recolectar pero aún tenían en su contenedor, los agentes marcaban a la papelera como su objetivo hasta soltar la basura.

Finalmente, al terminar todas sus tareas los robots se movían a posiciones vecinas de manera aleatoria para no detener la tarea de otros agentes en caso de llegar a interferir en su camino.

Implementación

Etapas 1

Para poder implementar la división de mapas, cada robot recibe como parámetro los límites inferiores y superiores correspondientes. De esta forma, el robot sólo era capaz de moverse a celdas dentro de sus límites.

Para implementar la búsqueda en profundidad empleamos dos pilas, una para poder almacenar las celdas que debemos visitar y otra para poder regresar a la celda previa si ya no podemos movernos.

Paralelo a esto, cada robot tiene una cantidad de celdas a explorar y todos comparten el total de celdas del mapa. Entonces esta etapa no concluye mientras aún haya celdas a explorar en el área del robot y en el área total.

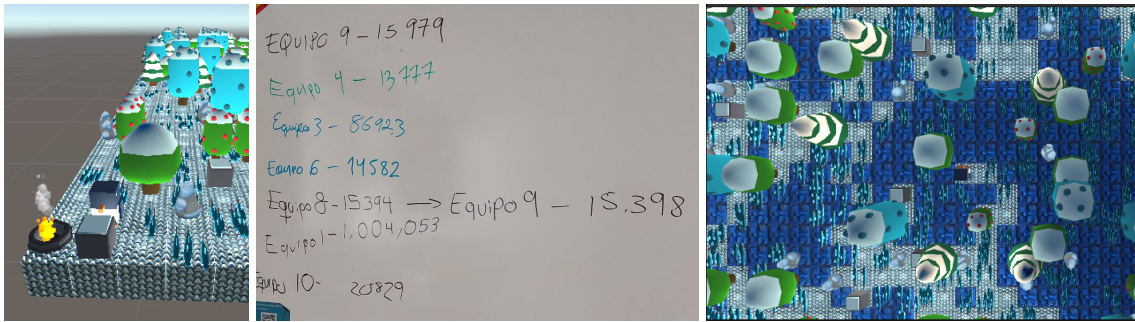
Etapas 2

En cuanto a la búsqueda de la basura más cercana, se recibe la posición de inicio y con ayuda de la bfs, se retorna una tupla con la posición de la más cercana o incluso vacía, en caso de no haber encontrado alguna.

Para el dijkstra, gracias a una priority queue donde tenemos ordenadas las posibilidades por número de pasos necesarios, vamos generando posibilidades desde el punto de inicio hasta coincidir con el objetivo especificado y almacenando las celdas previas a cada movimiento para, al terminar, retornar los pasos que te llevan al objetivo.

En caso de no poder llegar al objetivo, se valida que el objetivo se encuentre en el diccionario y de no ser así se retorna una lista vacía.

La prueba final



The screenshot shows a Windows desktop with a taskbar at the bottom containing icons for File Explorer, Edge, and several instances of the terminal application. The active window is a terminal titled "New Tab" and "Project View".

In the terminal, the user has navigated to the directory `C:\Users\kali\Documents\CTF\pwn\pwn3` and executed the command `python3 main.py`. The output shows the loading of various modules: `loading MultiClimbingSystem...`, `loading Agent subpackage...`, and `loading Graph subpackage...`. The terminal then displays the message `In termind of explorer ---- 827 steps` followed by a large block of hexadecimal data representing a memory dump or a list of addresses and values.

The file explorer window, titled "C:\Users\kali\Documents\CTF\pwn\pwn3", shows a single file named `main.py` with a size of 1 KB.

The taskbar at the bottom shows the following icons from left to right: File Explorer, Edge, and three instances of the terminal application.

The screenshot shows a Windows 10 desktop environment. A terminal window is open, displaying the output of a Qt application. The application appears to be a simple window with a large grid of 'X' characters. The terminal window title is "New Tab". The taskbar at the bottom shows the Start button, File Explorer, and the Qt application. The system tray shows the date and time as "11/11/2023 11:11".