
Desarrollo Basado en Agentes

Curso 2016-2017

Práctica 3: ' Gugel ' Vehicles.

Equipo 4

**Carmen Bueno Ben Boubker
Concepción Carcedo Carnero
Ernesto Gómez Leyva
Pedro Luis Hurtado González
Isidora Reina Molina
Martin Zumarraga Uribe**

ÍNDICE.

1. Introducción.
2. Descripción de la sociedad de agentes.
3. Diagramas.
 - 3.1. Diagrama de clases.
 - 3.2. Diagrama de secuencia.
 - 3.3. Diagrama de actividad.
4. Heurística.
5. Trazas.
 - 5.1. Mapa 1.
 - 5.2. Mapa 2.
 - 5.3. Mapa 3.
 - 5.4. Mapa 4.
 - 5.5. Mapa 5.
 - 5.6. Mapa 6.
 - 5.7. Mapa 7.
 - 5.8. Mapa 8.
 - 5.9. Mapa 9.
 - 5.10. Mapa 10.

1. Introducción.

Se propone la búsqueda en un mapa de un objetivo con un máximo de cuatro agentes y moverse a dicho objetivo una vez encontrado, para su desarrollo se utilizarán agentes con distintos roles, que les serán asignados en el momento de comenzar la ejecución y modificarán su comportamiento y capacidades.

Para su desarrollo se ha optado por que el primer agente haga una búsqueda y ya el resto de vehículos se muevan hacia él. También se ha decidido dividir su desarrollo particular en distintos roles (Dron, Coche y Camión). En los cuales el camión y el coche se moverán de manera distinta al dron, para así poder aprovechar sus capacidades.

2. Descripción de la sociedad de agentes.

Al tener múltiples agentes, es necesario sincronizarlos, para realizar la tarea propuesta, para conseguir esto, se propone el uso de una variación del algoritmo token-ring.

En dicho algoritmo, los agentes no podrán realizar ninguna acción mientras no tenga el token. El token será un mensaje que se irán pasando los agentes entre ellos.

Además el token se aprovecha para pasar un mensaje con la posición del objetivo, es decir, nuestra sociedad de agentes funcionaria de la siguiente manera:

- El agente 0, que en nuestro caso intentaremos que sea un dron, realizará un barrido del mapa en busca del objetivo.
- Pasará el token al siguiente agente y junto a este, un mensaje que contiene la posición en la que se encuentra el objetivo.
- El agente 1 “dará un paso” y pasara el token al siguiente agente (Agente 2), y la posición del objetivo, esto se repetirá hasta que los agentes lleguen al objetivo.

3. Diagramas.

3.1. Diagrama de clases. [\(Diagrama de Clases\)](#)

Nuestro diagrama de clases se divide de la siguiente manera:

- **Sensor:** Esta clase es la encargada de recibir la información del servidor con los datos de los sensores.
- **Agente:** Clase que representa al agente dentro de la sociedad y se encarga de las comunicaciones y el control del token.
- **Rol:** Clase abstracta que representa el rol de un agente (Dron, Coche, Camión) y define su comportamiento. Además de contener la parte común de la heurística.
- Las clases **Dron**, **Coche** y **Camión** representan los distintos roles que puede tomar un agente.
- **Heurística:** Esta clase posee diversos métodos comunes para su uso en las diferentes heurísticas.

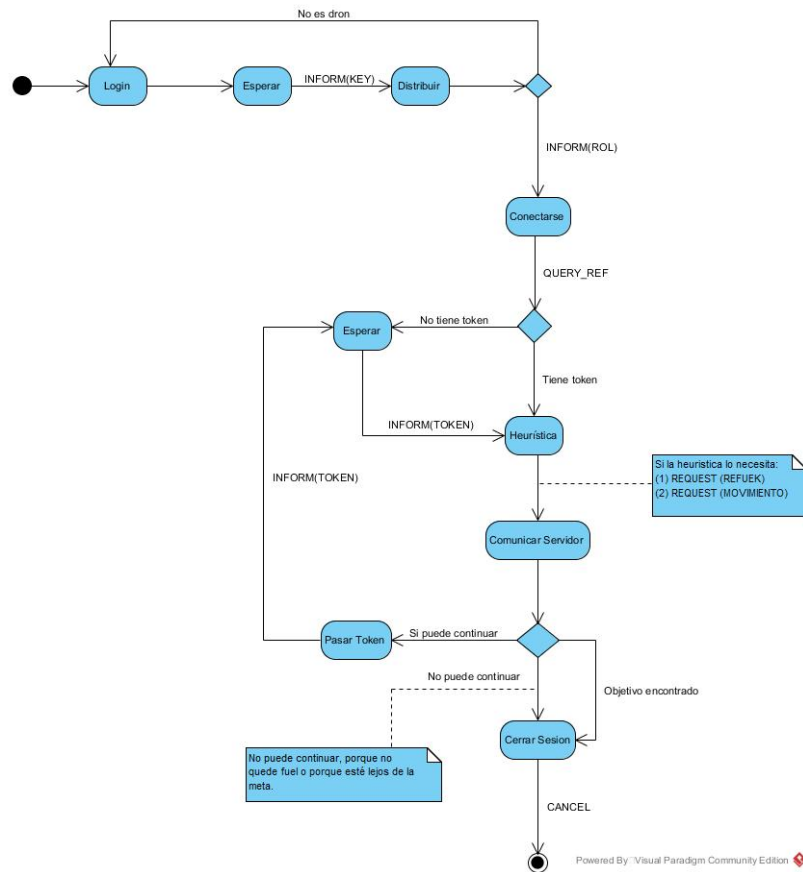
- [illegible]

Para terminar el agente0 es el encargado de mandar al servidor el mensaje de cierre de sesión y finaliza la ejecución.



Para comenzar, realizamos la búsqueda de un agente Dron, tras ello, todos los agentes reciben un INFORM con la key, el cual la reenvía al siguiente agente hasta que todos la han recibido (el último agente no la reenvía).

4



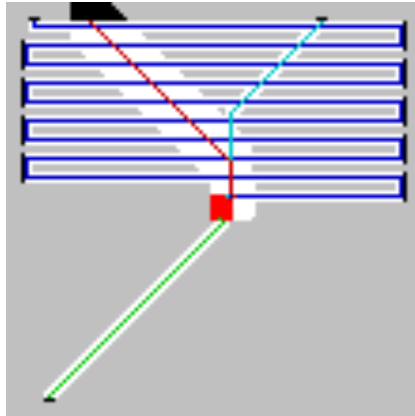
4. Heurística.

Hemos diseñado una estrategia basada en dos partes. Primero, necesitamos contar con un agente de tipo dron para hacer una búsqueda recorriendo todo el mapa. Aprovechando que el agente saldrá en uno de los bordes del mapa, localizará su posición (arriba o abajo) para comenzar a rastrear todo el mapa en dirección contraria a su origen, hasta encontrar el objetivo.

Una vez localizado el objetivo, pasamos a la segunda parte, que consistirá en que el resto de vehículos lleguen al objetivo. Los vehículos se van moviendo por turnos, cuando les llegue el token, y solo harán un movimiento por turno. El coche y el camión tienen dos heurísticas implementadas, que utilizarán según el número de muros o casillas libres que encuentren a su alrededor. Por un lado tenemos un algoritmo similar al algoritmo de la mano derecha, que consiste en rodear obstáculos como si llevara la mano derecha pegada a la pared y por otro lado un algoritmo greedy basado en pesos como el utilizado en la práctica anterior, solo que para hacer el cálculo nosotros creamos nuestro propio gps dándole los valores a las casillas según su cercanía al objetivo. En caso de que tengamos otro dron entre los vehículos, éste utilizará solo el algoritmo greedy basado en pesos, ya que el dron no tiene que evitar los muros.

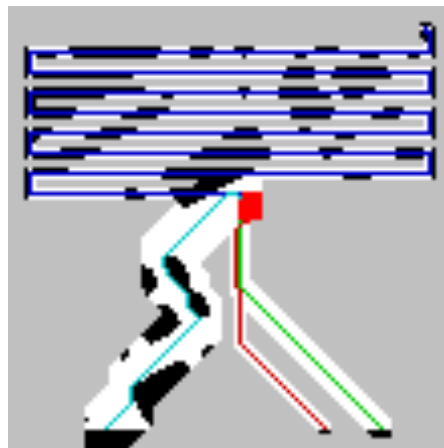
5. Trazas.

5.1. Mapa 1.



Conversation-ID: CONV-ofy1bfcr

5.2. Mapa 2.



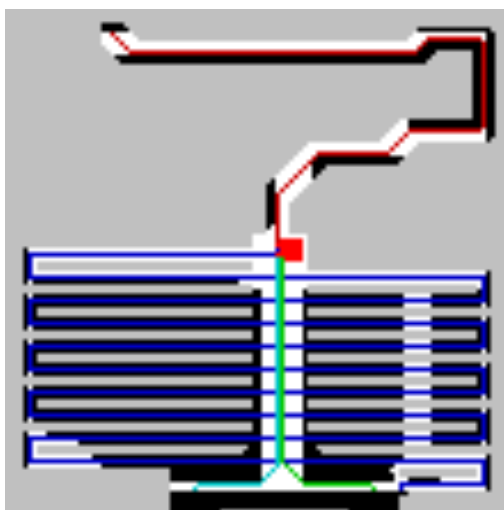
Conversation-ID: CONV-o2m8geks

5.3. Mapa 3.



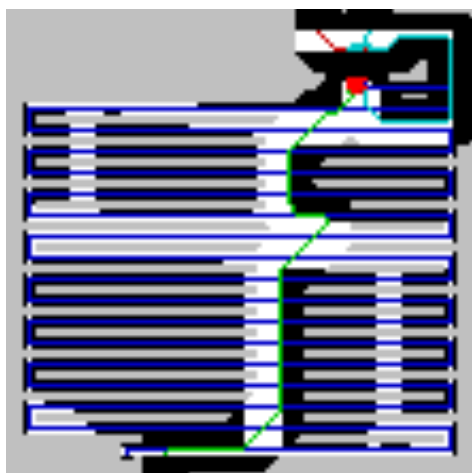
Conversation-ID: CONV -b9huaanv

5.4. Mapa 4.



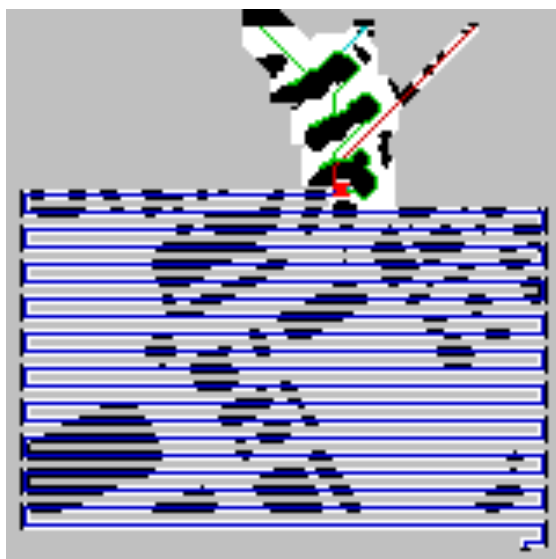
Conversation-ID: CONV-o9t87kjc

5.5. Mapa 5.



Conversation-ID: CONV-xtnqwodh

5.6. Mapa 6.



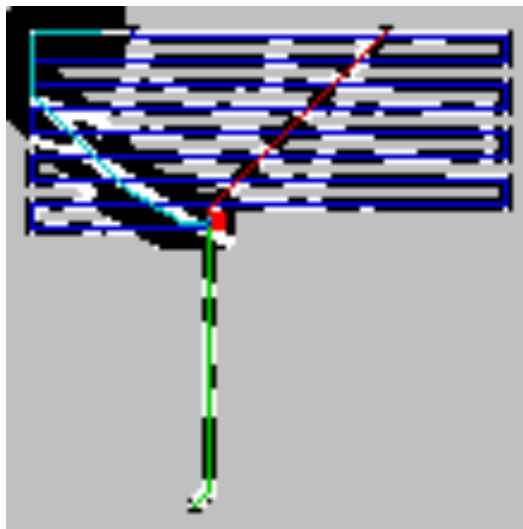
Conversation-ID: CONV-buexnoze

5.7. Mapa 7.



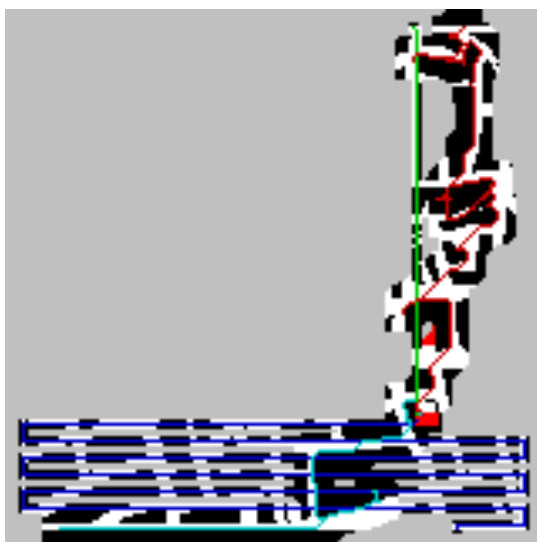
Conversation-ID: CONV-u4f6jq0j

5.8. Mapa 8.



Conversation-ID: CONV-gbrfyztv

5.9. Mapa 9.



Conversation-ID: CONV-qfds4g8m

5.10. Mapa 10.

Conversation-ID: