Fundamentos de Inteligencia Artificial

Práctica 3
Lógica difusa

Autor: Víctor Guzmán Pedrajas
DNI: 48717414-X
Turno: Miércoles, 8:30 - 10

Índice

Especificaciones del sistema difuso

Evolución del controlador de lógica difusa

A continuación se encuentra un resumen de la evolución del controlador de lógica difusa, tanto de las reglas como de los conjuntos.

El controlador que se incluía en el enunciado definía únicamente un conjunto para el sensor s0 y dos reglas que hacían que el robot girase a la izquierda al encontrarse un muro.

A partir de ahí, se definieron unos conjuntos similares al que ya venía para los sensores delanteros del robot (s0, s1 y s8), además de un conjunto para la variable sig. Los conjuntos de los sensores definían 3 términos: cerca, medio y lejos; y el conjunto sig definía izquierda, derecha y centro.

Las primeras reglas que se incluyeron en el controlador hacían que el robot girara hacia donde sig le decía. Esto hizo que se encontrara con esquinas de frente, de forma que se añadió una regla para girar si había pared delante pero espacio a un lado:

```
IF s0 IS cerca AND s1 IS far THEN rot IS muyizq;
```

Viendo que la medida de “cerca” de los conjuntos era demasiado lejana, se añadió el valor “muycerca” y “muylejos” a cada uno de los sensores. Además, para simplificar el desarrollo, se definieron todos los conjuntos de sensores, y todos con las mismas definiciones.

El siguiente paso fue definir una serie de reglas que intentasen evitar que el robot chocase con las paredes al encontrárselas de frente e intentar girar, para lo que se añadió el término “atrás” en la variable de velocidad, definida de forma que entre -1 y 0 valiese 1, y en el resto del dominio, 0. Esto conseguía evitar algunos giros demasiado estrechos, utilizando reglas como las siguientes:

```
IF s0 IS cerca THEN vel IS atras;
IF s0 IS verynear AND s4 IS NOT verynear AND s3 IS NOT verynear THEN vel IS atras,rot IS der;
```

Como con las reglas anteriores, el robot tardaba demasiado en girar al encontrarse paredes de frente, se eliminó la regla `IF s0 IS cerca THEN vel IS atras;` permitiendo que se acercara más a la pared, pero girando más correctamente. Esto dio problemas en algunos mapas, en los que chocaba con la pared.

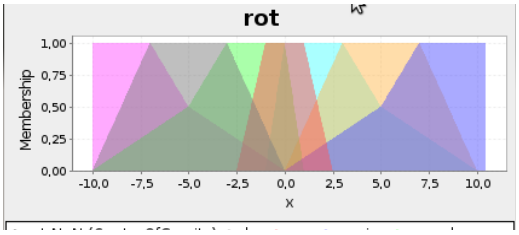
En este punto del desarrollo, se cambió la notación de las reglas, aumentando el código de la regla de 10 en 10 para poder añadir reglas intermedias sin tener que renumerar todas las reglas ya hechas, ni tener que incluir reglas con el código desordenado.
También se simplificaron algunas reglas para evitar tener reglas con muchas condiciones.

Después se cambió el conjunto del término “cerca”, de forma que dado un valor de 0,5m se lo considerara más cerca que intermedio, haciendo que el robot se pegara menos a las paredes.

Tras esto, se añadió el término “rozando” a los sensores, para dar un valor de emergencia en el que el robot se aparte de la pared si está excesivamente cerca.

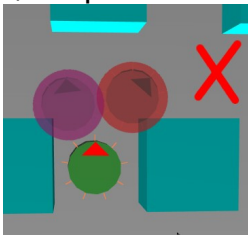
En este punto, el robot funciona medianamente bien, pero hay casos en los que las reglas crean un equilibrio, haciendo que el robot no se mueva al encontrar una esquina por el interior del giro, pese a que no se choca. Para solucionar esto, se intentó reducir la velocidad de movimiento en esos casos para que, si ocurren, el robot redujera drásticamente la velocidad, para poder evaluar las esquinas más correctamente. Sin embargo, esto hizo que se chocara con otros tipos de paredes.

Para intentar que no chocara con ninguna pared, se hizo que al estar muy pegado a una pared lateral, girara ligeramente hacia el lado contrario.
Por otro lado, en este punto se estandarizó la notación de los términos de los conjuntos, para mantenerlos todos en español (“cerca” en lugar de “near”, etc). Además, se definió el conjunto rot casi en su versión definitiva, que se explicará con más detalle más adelante:



A esta altura, se creó una rama divergente, para comprobar el uso de reglas sencillas con los conjuntos actuales. Así, se eliminaron todas las reglas, y se añadieron las reglas básicas para seguir el camino y evitar paredes. Sin embargo, hacía los giros excesivamente cerrados, por lo que se decidió cambiar el rango del conjunto de rot, que a partir de este punto va desde -5 hasta 5 rad/s.

En los choques con las esquinas, el robot tenía dos casos: o bien se había acercado demasiado a la esquina, con lo que rozaba la punta de la esquina al girar; o bien se alejaba demasiado intentando evitarlo y chocaba con la pared del otro lado (caso rojo y morado, respectivamente):

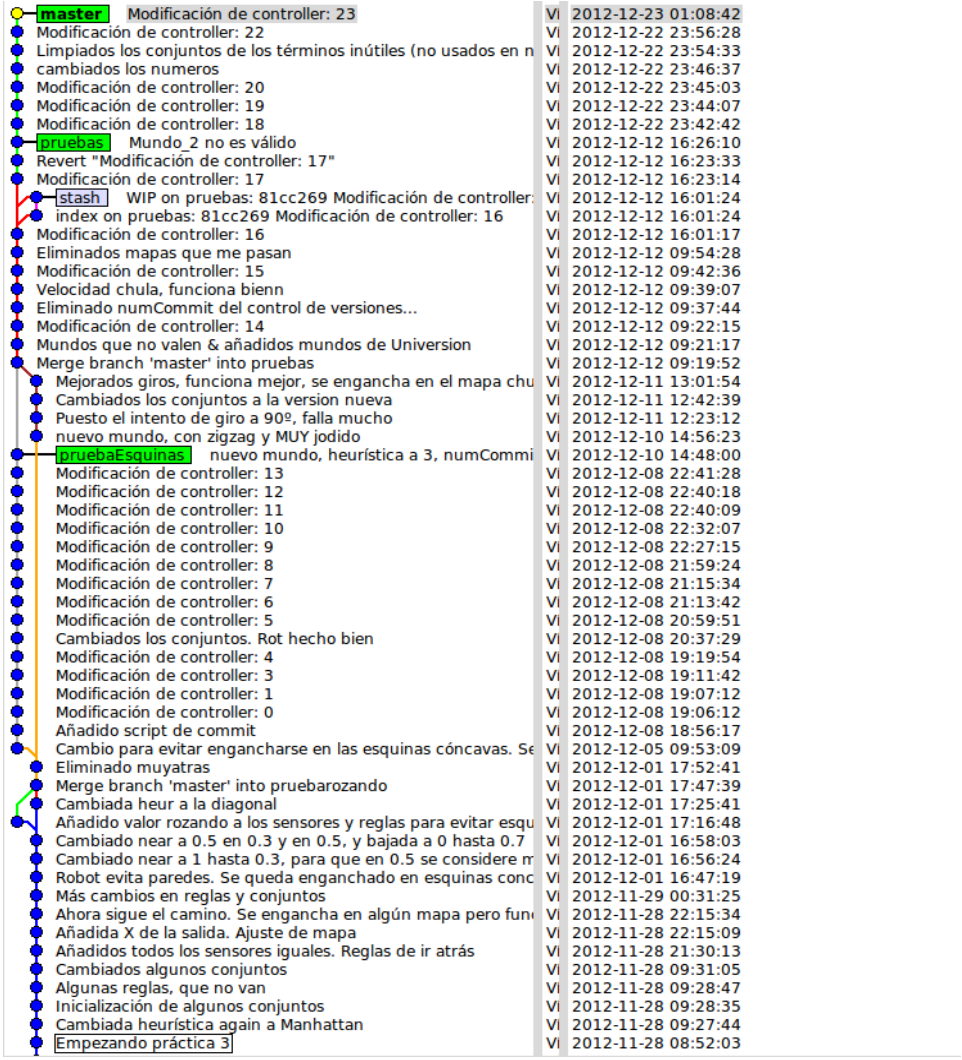


Para evitar estos casos, se intentó que el robot solo girara cuando sig le dijera que debía girar en ángulo recto. Así, se definió un término en sig para decir

cuándo se trataba de una esquina, un término en rot que girara 90º, y se cambiaron las reglas que hacían que el robot girara según sig, para que solo actuaran en caso de esquina, por ejemplo: `IF sig IS der AND sig IS esquina THEN rot IS esquinader;`

Una vez conseguido que el robot girase en ángulo recto, se sobrescribieron las reglas de la rama divergente anterior, ya que esta versión funcionaba mejor. Además, al girar en ángulo recto y a muy baja velocidad se consiguió evitar la mayoría de choques y enganches del robot, por lo que a partir de ese momento solo hubo que escribir algunas reglas de ajuste para casos extraordinarios.

Por último, se ha limpiado los conjuntos no usados en ninguna regla, y dentro de cada conjunto, las variables no usadas.



Esquema de la evolución de la práctica en orden cronológico inverso

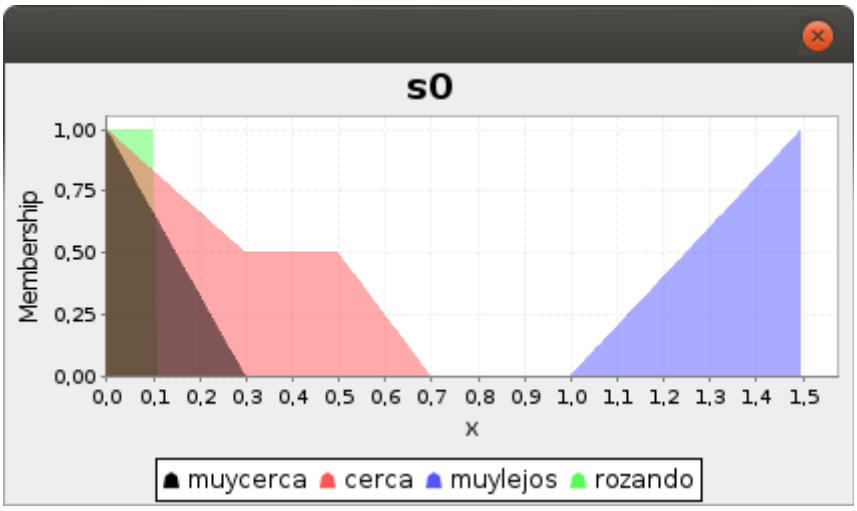
Estado final del controlador de lógica difusa

La última versión del controlador de lógica difusa es la siguiente:

Conjuntos difusos

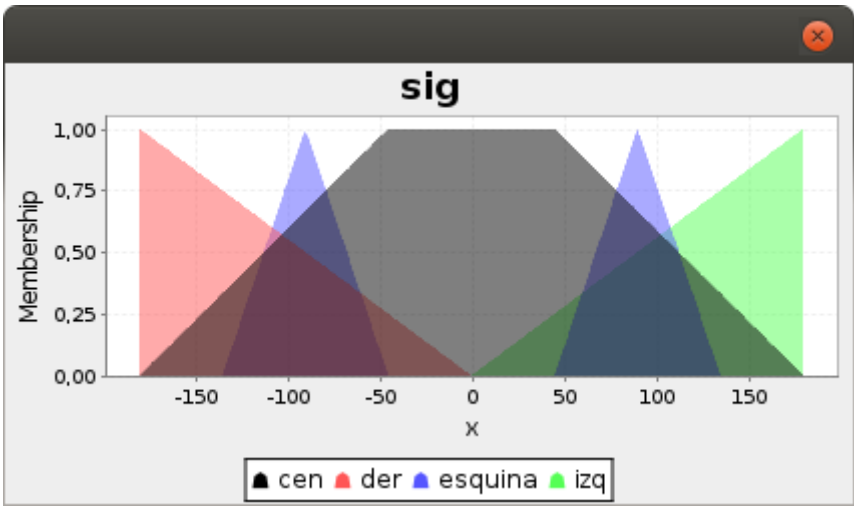
Sensores

Los conjuntos de sensores definen la distancia que detecta cada sensor hasta el primer obstáculo más próximo. Solo se han definido conjuntos para los sensores delanteros (0, 1, 2, 7 y 8), que son los que se usan en las reglas. Todos ellos tiene la misma definición de términos:



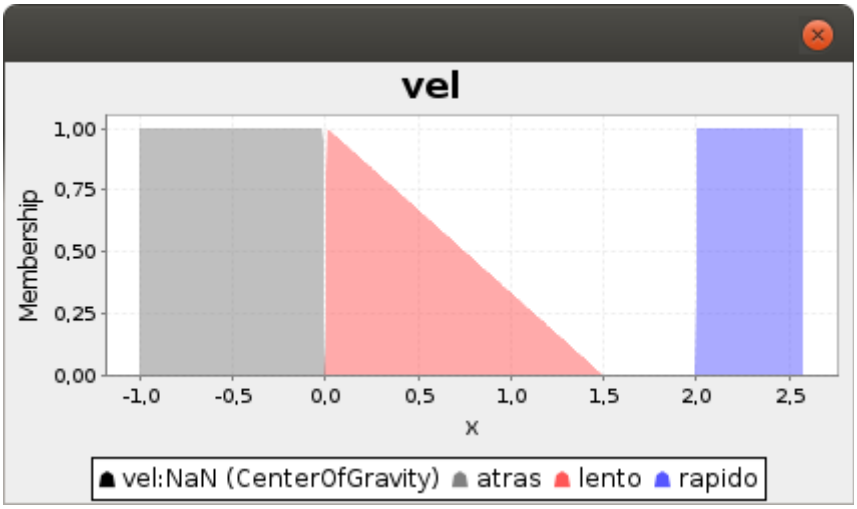
Dirección

El conjunto sig define, en grados, la dirección que debe seguir el robot para alcanzar la primera X en la próxima columna del mapa del camino calculado. El término “esquina” se define en los alrededores de los 90º.



Velocidad

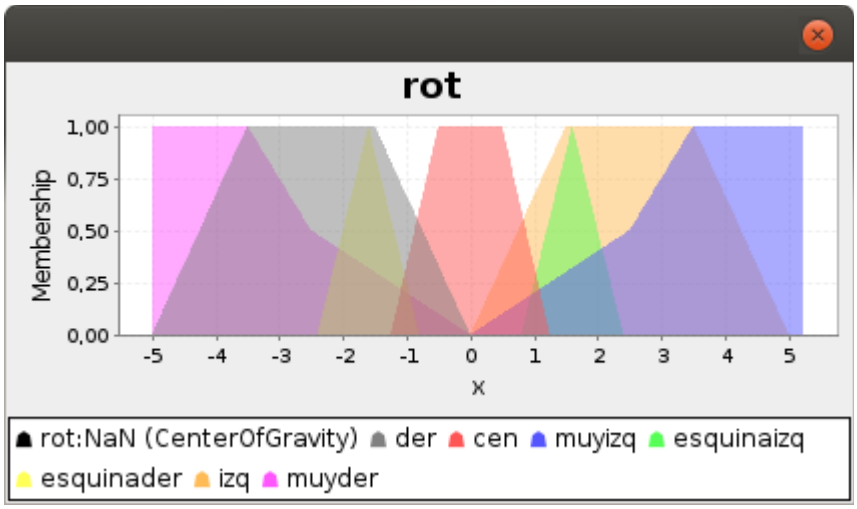
El conjunto de velocidad define las posibles velocidades a las que puede ir el robot. Su rango va desde -1 m/s hasta 2,5 m/s, el valor negativo hace que el robot camine hacia atrás, evitando obstáculos.



Rotación

El conjunto rot define la rotación del robot en radianes por segundo. Su rango es desde -5 hasta 5, de forma que, como mucho, el robot intente dar algo menos de una vuelta completa (2*pi = 360º) – este caso se dará en situaciones de emergencia, cuando el robot esté a punto de chocar.

Además, se definen 2 términos que indican al robot que debe girar 90º (pi/2) para recorrer esquinas con seguridad.



Reglas

Las reglas utilizadas en el controlador de lógica difusa son:

Reglas de control de la velocidad según el entorno:

```
RULE 10: IF s0 IS muylejos AND s1 IS muylejos AND s2 IS muylejos AND s7 IS muylejos AND s8 IS muylejos THEN vel IS rapido;

RULE 11: IF s0 IS NOT muylejos OR s1 IS NOT muylejos OR s2 IS NOT muylejos OR s7 IS NOT muylejos OR s8 IS NOT muylejos THEN vel IS lento;
```

Regla de retroceso en caso de peligro

```
RULE 20: IF s0 IS cerca THEN vel IS atras;
```

Reglas para intentar mantener el robot alejado de las paredes laterales

```
RULE 30: IF s1 IS muycerca THEN rot IS muyder;

RULE 31: IF s2 IS muycerca THEN rot IS der;

RULE 40: IF s8 IS muycerca THEN rot IS muyizq;

RULE 41: IF s7 IS muycerca THEN rot IS izq;
```

Reglas de seguimiento del camino

```
RULE 50: IF sig IS cen THEN rot IS cen;

RULE 60: IF sig IS der AND sig IS esquina THEN rot IS esquinader;

RULE 61: IF sig IS der AND sig IS NOT esquina THEN rot IS der;

RULE 62: IF sig IS izq AND sig IS NOT esquina THEN rot IS izq;

RULE 63: IF sig IS izq AND sig IS esquina THEN rot IS esquinaizq;
```

Reglas para evitar un choque inminente

```
RULE 70: If s8 IS muycerca AND s7 IS muycerca AND s0 IS muylejos AND s1 IS muylejos THEN vel IS atras,rot IS muyizq;

RULE 71: If s1 IS muycerca AND s2 IS muycerca AND s0 IS muylejos AND s8 IS muylejos THEN vel IS atras,rot IS muyder;
```

Reglas de emergencia

```
RULE 80: IF s8 IS rozando THEN rot IS muyizq, vel IS atras;

RULE 81: IF s7 IS rozando THEN rot IS muyizq;

RULE 90: IF s1 IS rozando THEN rot IS muyder, vel IS atras;

RULE 91: IF s2 IS rozando THEN rot IS muyder;
```

Experimentación

Para comprobar la corrección del controlador de lógica difusa se han utilizado varios mapas de prueba. Dada la limitación de que el robot no puede ir hacia la izquierda (debido a que la variable sig da el ángulo a la siguiente columna a la derecha), algunos mapas de la práctica anterior no son válidos; en esos casos se han eliminado y creado nuevos mapas:

Mapa original

Se trata del mapa proporcionado en el enunciado de la práctica:

```
20
10
10
* * * * * * * * * * * * * * * * * * * *
* . * . . . . . . . . . . . . . . . . *
* . * . . . . . . . . . . . . . . . . *
* . * . . . . . . . . . . - . . . . . *
* . * . . . . . . . . . . . . . . . . *
* . * . . . . . . . . . . . . . . . . *
* - * . . . . . . . . . . . . . . . . *
* . * . . . . . . . . . . . . . . . . *
* . * . . . . . . . . . . . . . . . . *
* . * * * * * . . . . . . . . . . . . *
* . . . . . * . . . . . . . . . . . . *
* . . . . . * . . . . . . . . . . . . *
* . . . . . * . . . . . . . . . . . . *
* . . . . . * . . . . . . . . . . . . *
* . . . . . * * * * * * . . . . . . . *
* . . . . . . . . . . . . . . . . . . *
* . . - . . . . . . . . . . . . . . . *
* . . . . . . . . . . . . . . . . . . *
* . . . . . . . . . . . . . . . . . . *
* * * * * * * * * * * * * * * * * * * *
```

Mapa 1

```
20
1
18
* * * * * * * * * * * * * * * * * * * *
* . . . . . . . . . . . . . . . . . . *
* . . . . . . . . . . . . . . . . . . *
* . . . . . . . . . . . . . . . . . . *
* . . . . . . . . . . . . . . . . . . *
* . . . . . . . . . . . . . . . . . . *
* . . . . . . . . . . . . . . . . . . *
* . . . . . . . . . . . . . . . . . . *
* . . . . . . . . . . . . . . . . . . *
* . . . . . . . . . . . . . . . . . . *
* . . . . . . . . . . . . . . . . . . *
* . . . . . . . . . . . . . . . . . . *
* . . . . . . . . . . . . . . . . . . *
* . . . . . . . . . . . . . . . . . . *
* . . . . . . . . . . . . . . . . . . *
* . . . . . . . . . . . . . . . . . . *
* . . . . . . . . . . . . . . . . . . *
* * * * * * * * * * * * * * * * * * * *
```

Mapa 2

```
10
1
1
* * * * * * * * * *
* . . . . . . * . *
* . * * * * * * . *
* . . . * . * * . *
* . * . * . . * . *
* . * . * . * * . *
* . * . . . . * . *
* . * * * * * * . *
* . . . . . . . . *
* * * * * * * * * *
```

Mapa 3

```
20
1
18
* * * * * * * * * * * * * * * * * * * *
* . . . * . . . * . . . * . . . . . . *
* . * . * . * . * . * . * . * * * * . *
* . * . * . * . * . * . * . . . . * . *
* . * . * . * . * . * . * . . . . * . *
* . * . * . * . * . * . * . * * * * . *
* . * . * . * . * . * . * . . . . * . *
* . * . * . * . * . * . * * * * . * . *
* . * . * . * . * . * . * . . . . * . *
* . * . * . * . * . * . * . * * * * . *
* . * . * . * . * . * . * . . . . * . *
* . * . * . * . * . * . * * * * . * . *
* . * . * . * . * . * . * . . . . * . *
* . * . * . * . * . * . * . * * * * . *
* . * . * . * . * . * . * . . . . * . *
* . * . * . * . * . * . * * * * . * . *
* . * . . . * . . . * . . . . . . * . *
* . * * * * * * * * * * * * * * * * . *
* . . . . . . . . . . . . . . . . . . *
* * * * * * * * * * * * * * * * * * * *
```

Mapa 4

```
15
3
7
* * * * * * * * * * * * * * *
* . . . . . . * . . . . . . *
* . . . . . . * . . . . . . *
* . - . . . . * . . . . . . *
* . . . . * * * * * . . . . *
* . . . . * . . . * . . . . *
* . . . . * . . . * . . - . *
* . . . . * . . . * . . . . *
* . . . . * * * * * . . . . *
* . . . . . . * . . . . . . *
* . . . . . . . . . * . . . *
* . . - . . . . . . * . . . *
* . . . . . * . . . . . - . *
* . . . . . * . . . . . . . *
* * * * * * * * * * * * * * *
```

Mapa 5

```
20
1
18
* * * * * * * * * * * * * * * * * * * *
* . * . . . . . . . . . . . . . . . * . *
* . * * * . . . . . . . . . . . . * * * . *
* . . . * . . . . . . . . . . . . * . . . *
* . . . * * * . . . . . . . . . * * * . . . *
* . . . . . * . . . . . . . . . * . . . . . *
* . . . . . * * * . . * * * . . . . . . *
* . . . . . . . . * . . * . . . . . . . . *
* . . . . . . . . * . . * . . . . . . . . *
* – . . . – . . . * * . . . – . . . . . *
* . – . – – – . . * * . . . . . – . . . *
* – . . . – . . . * * . – . – . . . – * *
* . . . . . . . . . . . . . . . – . . * *
* . . . . . . . . * * . – . – . . . – * *
* . . . . . . . . * * . . . . . – . . * *
* . . . . . . . . * * . – . – . . . – * *
* . . . . . . . . . . . . . . . – . . * *
* . . . . . . . . * * . . . – . . . . * *
* . . . . . . . . * * . . . . . . . . * *
* * * * * * * * * * * * * * * * * * * *
```

Mapa 6

```
20
1
18
* * * * * * * * * * * * * * * * * * * *
* . . – . . . – . . . – . * . . . . . * *
* . . . . – . . . – . . . * . . . . . * *
* * * * * * * * * * * * . * . . . . . * *
* . . . . . . . . . . . . * . . . . . * *
* . . . . . . . . . . . . * . . . . . * *
* . . . . . . . . . . . . * . * * * * * *
* . . . . . . . . . . . . * . * . – . * *
* . . . . . . . . . . . . * . * . . . * *
* . . . . . . . . . . . . * . * . – . * *
* . . . . . . . . . . . . * . * . * . * *
* . . . . . . . . . . . . * * . . * . * *
* . . . . . . . . . . . . . . . . * . * *
* . . . . . . . . . . . . * * * * * . * *
* . . . . . . . . . . . . * . . . . . * *
* . . . . . . . . . . . . * . . . . . * *
* . . . . . . . . . . . . * . . . . . * *
* . . . . . . . . . . . . * . . . . . * *
* . . . . . . . . . . . . * . . . . . * *
* * * * * * * * * * * * * * * * * * * *
```

Mundo 7

```
20
16
3
* * * * * * * * * * * * * * * * * * * *
* . . . * . . . . . . * . . . — . . . *
* . . . * . . . . . * . . * . . . . . *
* . . . * . . . . . * . . * . — . . . *
* . . . * . . . . . * . . * . . . . . *
* . . . * . . . . * . . . . — . — . . *
* . . . * . . . * . . . . * . — . . . *
* . . . * . . * . . . . . * . . . . . *
* . . . * * * . . . . . . * . . . . . *
* . . . . . . . . . . . * . . . . . . *
* . . . . . . . . . . * . . . . . . . *
* . — . * * * * * * . . . . . . . . . *
* . . . * . . . . . . . . . . . . . . *
* . — . * . . . . — . . . . . — . . . *
* . . . * . . . . . . . . . . . . . . *
* . — . * . . . . — . . — . . . . . . *
* . . . * . . . . . . . . . . . . . . *
* . . . * . . . . . . . . . . . — . . *
* . . . * . . . . . . . . . . . . . . *
* * * * * * * * * * * * * * * * * * * *
```

Mundo 8

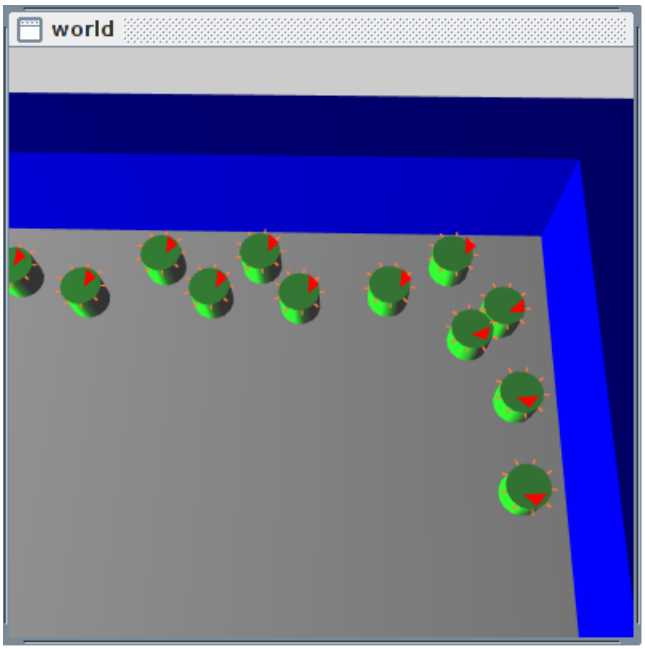
```
21
10
10
* * * * * * * * * * * * * * * * * * * * *
* . . . . . . . * . . . . . . . * . . . *
* . . . . . . . * . . . . . . . * . . . *
* . . . . . . . * . . . . . . . * . . . *
* . . . * . . . * . . . * . . . . . . . *
* . . . * . . . * . . . * . . . . . . . *
* . . . * . . . . . . . * . . . . . . . *
* . . . * . . . . . . . * . . . . . . . *
* . . . * . . . . . . . * . . . . . . . *
* . . . * . . . . . . . * . . . . . . . *
* . . . . . . . . . . . * . . . * . . . *
* . . . . . . . . . . . . . . . * . . . *
* . . . . . . . * . . . . . . . * . . . *
* . . . * . . . * . . . . . . . * . . . *
* . . . * . . . * . . . . . . . * . . . *
* . . . * . . . * . . . . . . . * . . . *
* . . . * . . . * . . . * . . . * . . . *
* . . . * . . . . . . . * . . . * . . . *
* . . . * . . . . . . . * . . . . . . . *
* . . . * . . . . . . . * . . . . . . . *
* . . . * . . . . . . . * . . . . . . . *
* * * * * * * * * * * * * * * * * * * * *
```

Otros experimentos

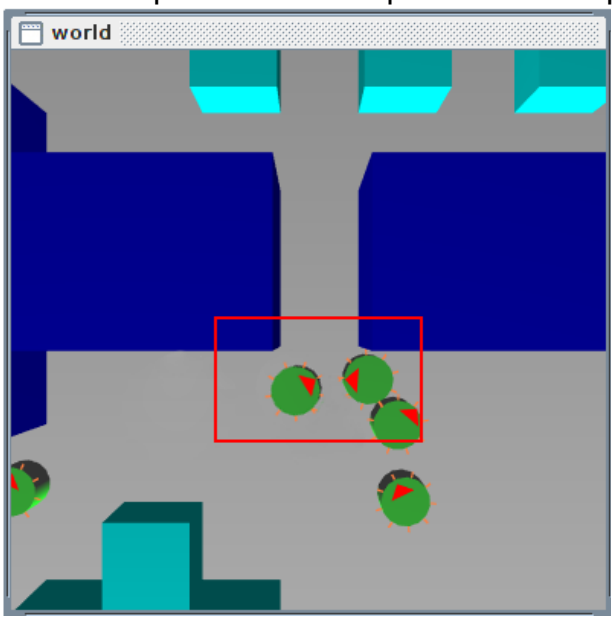
Para comprobar casos muy extremos, se ha añadido muchos robots a un mismo mapa, de forma que todos intenten llegar por su cuenta a la salida, produciendo muchas posibilidades de choque entre ellos. Esto se ha conseguido añadiendo muchas veces la línea

```
add(new MiRobot(new Vector3d(-(mitadMundo-practica1.origen-1-1), 0, mitadMundo-1), "SI robot2", practica1));
```

en el fichero `MiEntorno.java` cambiando la posición origen y el string identificador del robot.



Este experimento ha sido bastante exitoso, excepto en algunos casos en los que dos robots intentaban pasar a la vez por el mismo pasillo



En este caso, ambos calculan la distancia al otro y, decidiendo que pueden pasar, avanzan, reduciendo la distancia el doble de rápido, con lo que chocan.