

Proyecto en CLIPS: Willy en el espacio

Curso 2023/2024

1 Enunciado

Se debe crear un sistema inteligente que dé solución al siguiente problema:

Disponemos de un mapa dividido en $M \times N$ casillas por el que Willy (nuestro personaje) deberá moverse por el espacio en busca de regresar al planeta Tierra. El problema está en que hay casillas del mapa donde podrá encontrarse con agujeros negros, donde al acceder morirá (fin de la partida), además de un (único) alienígena maligno cuyo encuentro es mortal (fin de la partida). Willy no sabe a priori dónde están cada uno de los peligros, ni el resto de ítems del mapa, por lo que deberá explorarlo, casi a ciegas. Como ayuda, cuando Willy está situado junto a una casilla que contiene agujeros negros percibirá una fuerza de atracción, y si está situado junto al malvado alienígena percibirá un característico ruido.

Willy solo puede desplazarse, en cada momento, a las casillas situadas a la izquierda, derecha, arriba o abajo, pero nunca en diagonal, ni más de una casilla por movimiento. Como ayuda, dispone de un láser que podrá disparar en una de las cuatro direcciones, con el objetivo de matar al alienígena. El disparo del láser recorrerá la fila o columna seleccionada, en la dirección indicada, hasta salir del mapa o alcanzar al alienígena. En cualquiera de las dos situaciones, Willy ya no podrá volver a disparar otra vez. En caso de haber matado al alienígena, entonces ya no se escuchará su sonido, y será seguro pasar por la casilla en la que se encontraba.

Se debe programar un sistema basado en reglas, haciendo uso de CLIPS, que decida en cada momento el movimiento que debe realizar Willy para alcanzar su objetivo (encontrar el planeta Tierra).

En la Figura 1 se muestra un ejemplo de mapa con la situación de los ítems en dicho mapa. Nótese que dichos ítems no son visibles para Willy hasta que son visitados.

	1	2	3	4
1			E	
2		A		H
3	W			
4			H	

Figure 1: Ejemplo de mapa. Símbolos: W=Willy, A=alienígena, E= planeta Tierra, H=agujero negro

2 Material proporcionado

1. Tener cualquier java runtime instalado en el sistema operativo. Se puede verificar que la instalación existe escribiendo en el terminal: `java -version` o `java -version`.
2. Descargar de Moodle el zip con el programa y descomprimirlo en la carpeta que se prefiera.
3. Copiar la librería del directorio que se corresponda con sistema operativo (win32, win64, linux32, ...) en el directorio donde se encuentra el fichero WillyDemo.jar.
4. Ejecutar el programa mediante la orden:

```
java -Djava.library.path=. -jar WillyDemo.jar
```

- Aclaración sistema UCO Thinstation (Linux de las aulas): las configuraciones de Java actualmente en uso en este sistema operativo dan varios problemas de compatibilidad con el programa. La siguiente opción de ejecución es la que menos errores da:

```
/usr/local/java-7-openjdk-amd64/bin/java -Djava.library.path=. -jar WillyDemo.jar
```

Aparecerá una pantalla como la que se muestra:

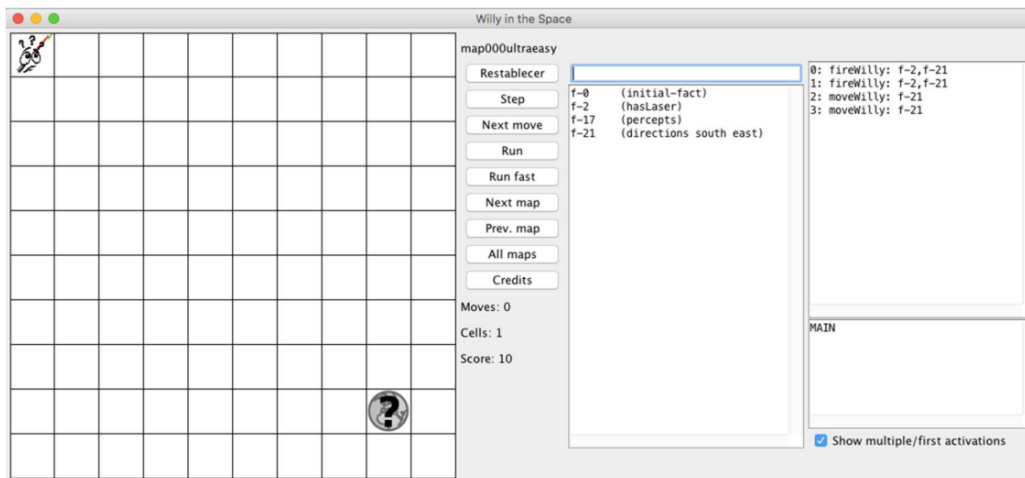


Figure 2: Ejemplo de mapa. Símbolos: W=Willy, A=alienígena, E= planeta Tierra, H=agujero negro

Se proporciona además un código base *willy.clp* sobre el que se deben añadir las correspondientes reglas que implementen la estrategia de movimientos de Willy para conseguir resolver el problema.

3 Funcionamiento

Para implementar el programa que solucione el problema, se deben programar las reglas necesarias en un fichero *willy.clp* que debe estar en el directorio de ejecución del programa anterior.

En la base de hechos habrá un conjunto de hechos que informan al agente de las percepciones que tiene en la casilla en la que se encuentra, si tiene aún la pistola láser, y las posibles direcciones de movimiento desde su casilla actual. El sistema actualiza automáticamente estos hechos, por lo que se desaconseja totalmente realizar modificaciones sobre ellos. Los hechos que nos dan esta información serían:

- **(directions north south east west):** nos permite conocer las direcciones posibles para las que Willy puede realizar alguna acción (disparar o moverse). El sistema las actualiza automáticamente, sin tener por qué mantener el orden indicado entre los valores, de forma que el valor north no estará disponible si Willy está en la primera fila, etc.
- **(hasLaser):** nos indica que la pistola láser está disponible.

- **(percepts Pull)**: nos permite conocer que Willy está junto a una casilla con un agujero negro.
- **(percepts Noise)**: nos permite conocer que Willy está junto a una casilla con el alienígena.
- **(percepts Noise Pull)** o **(percepts Pull Noise)**: indicará que Willy está simultáneamente junto a una casilla con un agujero negro y junto al alienígena.

Funciones disponibles para que Willy realice acciones:

- **(fireLaser <dirección>)**: nos permite disparar el láser en una determinada dirección, que debe ser uno de los valores en el hecho (directions).
- **(moveWilly <dirección>)**: desplaza a Willy en una dirección, que debe ser uno de los valores en el hecho (directions).

Será responsabilidad del alumno crear hechos que permitan guardar información sobre las casillas exploradas y las correspondientes “percepciones”, para conseguir que el programa consiga ganar el mayor número de veces.

Otra condición por la que Willy fallecerá será el haber realizado 1000 o más movimientos, dado que la reserva de oxígeno en la nave espacial es limitada, y Willy debe llegar al planeta Tierra lo antes posible.

Se pondrá a disposición del alumno un conjunto de mapas de prueba, no necesariamente iguales a los utilizados por el profesorado para evaluar la práctica, que pueden guiar en el diseño del agente.

Por último, observe que el programa ofrece algunas estadísticas acerca de los objetivos alcanzados por el agente, como es si ha ganado o ha muerto (porcentajes en el caso de ejecutarse sobre todos los mapas), número de movimientos, casillas nuevas que ha recorrido y un score. Este último se calcula de la siguiente forma:

- **Si Willy ha encontrado el planeta Tierra**: El score será de 1000 menos el número de movimientos realizados más el número de casillas diferentes por las que ha pasado. El objetivo es premiar a los agentes que encuentren pronto el planeta Tierra sin pasar frecuentemente por casillas ya visitadas.

- **Si Willy no ha encontrado el planeta Tierra:** Se sumarán 10 puntos por cada casilla nueva recorrida y restará un punto por cada movimiento. El objetivo es valorar la capacidad de los agentes que, aunque no encuentren el planeta Tierra, sean capaces de explorar la mayor parte del mapa, sin pasar frecuentemente por casillas ya visitadas. En este caso, si además Willy ha muerto, se restarán 1000 puntos.

4 ¿Qué se valora?

Se valorará positivamente:

- La inclusión de comentarios descriptivos en el código.
- El uso de elementos vistos en clase (módulos, plantillas, funciones, EC avanzados...).
- La implementación de estrategias avanzadas para la resolución del problema (ej. algoritmos de búsqueda en árboles).
- El funcionamiento correcto de su propuesta sobre el conjunto de mapas de evaluación que se utilizarán.

Se valorará negativamente:

- Cualquier código total o parcialmente copiado será considerado PLAGIO y automáticamente será valorado con cero. No teniendo opción a recuperar la puntuación correspondiente a esta parte durante el actual curso. En caso necesario, se informará a la autoridad competente.
- El uso abusivo de funciones procedurales (if, switch, while,...).

5 Planificación de trabajo

El trabajo en el proyecto práctico de CLIPS ocupará tres semanas. Durante este tiempo se abordarán los siguientes objetivos:

Fase 1: Se parte del código disponible en Moodle, denominado `willy-completar.clp`, para implementar las defensas más básicas de Willy. Estas incluyen evitar que se exceda el número de pasos y retroceder a la celda previa si se detecta peligro.

Fase 2: Sobre el programa diseñado, se incluirá una estructura de celdas visitadas. A partir de aquí se puede diseñar algún tipo de estrategia para que Willy se mueva de una forma más inteligente. La estrategia más básica esperable es priorizar la visita de celdas desconocidas frente a otras ya visitadas. Sin embargo, se valorará positivamente técnicas de búsqueda más avanzadas para construir los “camino” que recorrerá Willy.

Fase 3 (Avanzado): Sobre el programa diseñado, se incluirá la estrategia para disparar y matar al alienígena. Para ello se debe tener en cuenta que el alienígena solo aparece una vez como máximo en el mapa y que su posición puede determinarse cuando se han detectado al menos dos celdas con ruido en el mapa. También se debe tener en cuenta que Willy tiene un único disparo, y mientras éste no se haya efectuado existirá el hecho (hasLaser). Willy puede disparar a norte, sur, este u oeste, y el disparo recorrerá toda la línea del mapa desde donde Willy se encuentre en adelante.

Es importante señalar que el código programado durante estas fases no se tiene que entregar ni se utilizará para la evaluación continua de la asignatura. Sin embargo, es fundamental para entender el funcionamiento de este puzzle y poder superar satisfactoriamente una prueba relacionada que se planteará para la última semana del trabajo.