

Memoria del trabajo: "Willy en el espacio"

Autores: Ángel Heredia Pérez y Sergio Gómez Morales

A continuación, se explicarán las estrategias seguidas para resolver el problema que se plantea en el juego "Willy en el espacio".

Hemos creado una serie de plantillas para almacenar la información del juego:

-**Plantilla willy:** almacena información sobre la casilla en la que willy se encuentra actualmente en dos campos que se corresponden con las coordenadas **x** e **y**. Cabe decir que el sistema de coordenadas que hemos utilizado establece el origen en la casilla en la que willy empieza, los movimientos a la derecha serán el eje positivo de la **x** y los movimientos hacia arriba serán el eje positivo de la **y**.

-**Plantilla casilla:** almacena información de las casillas que conoce Willy, contiene los campos:

x: la coordenada **x** de la casilla.

y: la coordenada **y** de la casilla.

visited: indica si Willy ha pasado por esa casilla.

safe: indica si la casilla es segura o puede contener un peligro (no tiene por qué).

alien: indica si la casilla contiene un alien.

hole: indica si la casilla contiene un agujero negro.

pull: indica si se ha detectado Pull en esa casilla.

noise: indica si se ha detectado Noise en esa casilla.

danger: indica que la casilla contiene un peligro, pero no sabemos cuál es.

-**Plantilla STOP:** contiene el slot **state** que indica si el programa debe parar o no.

El trabajo lo hemos dividido en distintos módulos dependiendo de la función que Willy realice, forma que el programa va iterando entre los módulos de forma ordenada (percepción, inferencia, movimiento), además de otro módulo myMAIN que realiza las iteraciones. Los módulos son:

-**Módulo myMAIN:** controla la iteración de los módulos y el fin de la ejecución.

1) **passToInitialRule**: inicia la ejecución del programa.

2) El conjunto **passToPercepcion/Inferencia/Movimiento** pasa la ejecución a uno de los tres módulos.

-**Módulo de percepción**: Willy obtiene información de la casilla en la que está y las que le rodean mediante las reglas que describimos a continuación:

1) **firstState**: esta regla se ejecuta una sola vez al principio del programa, inicializa el sistema de coordenadas y establece la posición inicial de Willy.

2) **actualState**: cada vez que se realiza un movimiento, actualiza la casilla en la que se encuentra Willy como visitada.

3) **surroundingsX/Y**: crea las casillas circundantes a donde se encuentra actualmente.

4) **detectNoise/Pull**: si detecta alguno de estos peligros, actualiza el campo pull o noise de la casilla en la que está con ese peligro, dependiendo de lo que detecte.

5) **lookAroundX/Y**: determinan si las casillas de alrededor son seguras o no.

6) **lookForAlienX/Y**: determina que en las casillas de alrededor no hay alien basándose en que no haya detectado Noise en la casilla en la que está actualmente.

7) **lookForHoleX/Y**: determina que en las casillas de alrededor no hay alien basándose en que no haya detectado Pull en la casilla en la que está actualmente.

8) **safeState**: utiliza la inferencia realizada con las dos reglas anteriores para determinar que en las casillas de alrededor no hay peligros y Willy puede moverse hacia ellas de forma segura.

9) El conjunto de reglas **direccion** y **conversionNorth/South/West/East** transforma las direcciones a las que se puede mover Willy a su homólogo en coordenadas, por ejemplo: **direccion south** = y -1.

10) **cleanDirX/Y**: elimina hechos de direcciones que no se van a utilizar más para evitar que Willy se intente salir del mapa.

- **Módulo de inferencia**: utiliza la información recabada en el módulo de percepción y realiza inferencias:

1) **infer-alien**: conjunto de reglas que infieren la posición del alien en función de las casillas que Willy ha visitado y tiene registradas en su base de hechos. Puede detectar un alien si ha estado en dos casillas opuestas con respecto a la posición del alien, y también cuando ha estado en tres casillas colindantes alrededor del alien.

2) **detect-alien**: conjunto de reglas que infieren al alien cuando Willy ha estado en las casillas de alrededor de la casilla donde detecta Noise, teniendo en cuenta si lo ha detectado en mitad del mapa, en una de las paredes, o en una de las esquinas.

3) **discard-alien**: conjunto de reglas que, una vez muerto el alien, elimina los ruidos de las casillas que conoce Willy e infiere que en la casilla donde se encontraba el alien, ya no hay nada.

4) **shoot**: conjunto de reglas que se activan una vez detectado el alien, y dependiendo de la posición en la que se encuentre el alien y Willy, dispara en la dirección en la que esté el alien.

- **Módulo de movimiento**: realiza movimientos sobre Willy utilizando la información que infiere en el módulo anterior:

1) **init**: inicializa un vector de movimientos que contendrá los últimos movimientos realizados por Willy, y se comportará como una pila. Resultará útil si Willy necesita volver de donde vino, o bien si Willy está realizando un recorrido impuesto por código. Esta regla también inicializa a 0 el contador de movimientos, que servirá para controlar que Willy no haga 1000 movimientos y muera.

2) **T-algorithm**: conjunto de reglas que, una vez que Willy detecta Noise, hacen que busque al alien por los alrededores, de forma que la mayoría de las veces que Willy detecta Noise, en los siguientes pasos detecta la existencia del alien y lo mata. Solo no ejecuta esta acción si durante el proceso, Willy detecta Pull en alguna de las casillas por las que tiene que pasar para inferir la posición del alien, de forma que sigue su recorrido habitual. La estrategia seguida es formar una T, siendo la base la casilla en la que detectó Noise.

3) **calm-down**: esta regla se activa cuando, durante el algoritmo T, Willy detecta Pull en alguna de las casillas por la que tiene que pasar, de forma que esta regla hace que Willy abandone el proceso de matar al alien para evitar que pueda caer en un agujero negro.

4) **moveNorth/South/East/West**: mueven a Willy hacia casillas seguras y no visitadas con una prioridad alta.

5) **moveBackNorth/South/East/West**: mueven a Willy por donde ha venido si las casillas por las que puede seguir no son seguras.

6) **stop**: establece una bandera de parada del programa dadas ciertas condiciones en las que Willy no puede seguir, bien porque no quedan casillas por visitar, o porque ha llegado al límite de movimientos.

- **Todos los módulos:**

1) **exitModule**: regla con muy baja prioridad que hace que el programa salga del módulo donde está.