

Proyecto práctico: Willy en el desierto.

Sistemas Inteligentes
Universidad de Córdoba

Realizado por:
Raúl Serrano Torres
i22setor@uco.es

Introducción.

En este documento se explica la solución propuesta por el alumno Raúl Serrano Torres para resolver los mapas del proyecto "Willy en el desierto" de la asignatura de Sistemas inteligente de la UCO.

1. Plantilla y hechos iniciales.

Lo primero de todo es crear una plantilla que guardara la información sobre las casillas del tablero. Esta consta de los siguientes slots:

- Pos: Los valores de este slot serán utilizados para marcar las casillas como visitadas. Para poder saber dónde estamos, se toma su valor como si indicara unas coordenadas. Esto es posible porque el mapa es 9x9 y nunca se encontrarán valores de 3 cifras. Así, las decenas se considerarán como el valor de y en un eje de ordenadas y las unidades como el valor de x en un eje de abscisas. Por ejemplo, si la posición actual es 12, querrá decir que Willy está una fila más arriba y dos columnas más a la derecha que cuando empezó el recorrido. Su posición inicial será 0 ($y=0$, $x=0$). La posición inicial da igual, ya que esto solo se utiliza para poder marcar eficientemente las casillas como visitadas.
- Visitada: Indica si la casilla ha sido visitada (1) o no (0).
- Sonido: Indica si se ha percibido un sonido (1) o no (0).

1.1 Hechos iniciales.

Tras crear la plantilla y antes de crear las reglas, se afirman los hechos iniciales. Dado que Willy empieza en la casilla 0, marcaremos esta como visitada.

- (casilla (pos 0)(visitada 1)(sonido 0))

También se añadirá a la base de hechos:

- (celda 0): Este hecho se utiliza para tener información de la posición en la que se encuentra Willy. Con la plantilla *casilla* y este hecho se irá mapeando el recorrido para guardar información de las casillas visitadas.
- (mover): Hecho que será condición en las reglas de movimiento para que Willy se pueda mover.
- (Contador 1): Hecho que se irá incrementando con cada movimiento de Willy.
- (posición 0): Hecho que se usará para activar algunas reglas.

2. Reglas de movimiento.

Tras crear la plantilla y los hechos iniciales. Procedemos a realizar las reglas que definirán el movimiento de Willy. Cuatro de estas reglas se corresponden con los cuatro movimientos posibles: *MoverN*, *MoverS*, *MoverE* y *MoverW*.

Debido a que todas estas reglas tienen la misma prioridad, Willy se moverá de forma aleatoria a las casillas que no han sido visitadas.

Solo se podrán activar las reglas cuya dirección esté disponible. Cuando Willy este pegado a los límites de los mapas no se podrá mover más allá de ellos. Por ejemplo, si al empezar el recorrido Willy está en la esquina superior izquierda, los movimientos al norte y al oeste no estarán disponibles.

Para que cualquiera de estas reglas se active lo más importante es que la casilla/celda a la que se va a mover no esté visitada. Esto se consigue comparando la celda actual con la base de hechos que se ha ido actualizando. Si existe una casilla cuya posición tenga el mismo valor que la suma celda actual más 10, en caso de norte, -10 en caso de sur, -1 en caso de este y +1 en caso de oeste, la regla no se activará. También es requisito que el hecho *mover* este afirmado.

Como consecuencia de la activación de alguna de estas reglas, se moverá a Willy y se afirmará el hecho *Actualizar*, el cual junto con posición (norte, sur, este u oeste) activará la regla de actualización correspondiente. También se quitará el hecho *mover* de la base de hechos para impedir que se activen otras reglas de movimiento antes de actualizar.

El problema de estas reglas se produce cuando, para llegar a la meta, Willy tiene que pasar por una casilla en la que tiene peligros a ambos lados. Dado que estas reglas solo solucionan el peligro "huyendo" en la dirección contraria, nunca podrá superar los mapas por los que tenga que pasar por un pasillo entre peligros.

En caso de que todas las casillas que rodean a Willy estén visitadas, ninguna de las reglas anteriores se activará, por lo que Willy quedaría parado. Para solucionarlo, se añade la regla *rodeadoVisitadas*, que lo moverá a una posición aleatoria.

2.1 Reglas de actualización de movimiento.

Una vez creadas las reglas de movimiento, se establecen otras cuatro reglas (una para cada dirección) para actualizar la posición de Willy. Estas son *ActualizarN*, *ActualizarS*, *ActualizarE* y *ActualizarW*.

Lo que hacen estas reglas es guardar la celda en la que estaba Willy, cuya información viene dada por *celda*, en la base de hechos, añadiendo nuevos hechos usando la plantilla *casilla*, en donde además de guardar la celda anterior, la marcará como visitada. Así con el slot *pos* se irá guardando en la base de hechos las casillas visitadas y se irá construyendo un mapa del recorrido de Willy. También se actualiza el contador de movimientos y se afirma el hecho *mover*, que permitirá que se sigan activando reglas de movimiento.

3. Reglas para evitar peligros.

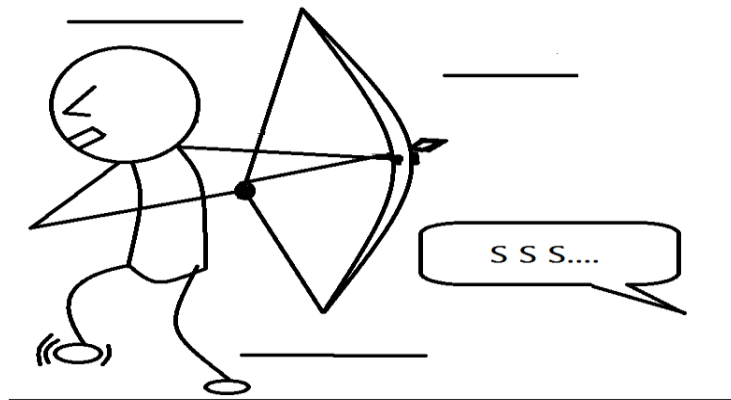
Para evitar que Willy no muera a causa de encontrar algún peligro, se han creado 16 reglas (*TemblorN*, *TemblorS*, *TemblorE*, *TemblorW*, *SerpienteN*, *SerpienteS*, *SerpienteE*, *SerpienteW*, *TemblorSerpienteN*, *TemblorSerpienteS*, *TemblorSerpienteE*, *TemblorSerpienteW*, *SerpienteTemblorN*, *SerpienteTemblorS*, *SerpienteTemblorE*, *SerpienteTemblorW*) que le permitirán evitar estos peligros.

3.1 Reglas para evitar arenas movedizas.

El mecanismo de estas reglas es muy sencillo. Si al moverse a una casilla Willy percibe un temblor o escucha un sonido, este se moverá a en la dirección contraria, es decir, volverá a la casilla anterior. La dirección viene marcada por *posición*. Para conseguir esto, lo que hace la regla es cambiar la posición/dirección a la contraria (por ejemplo: si se percibe al tener la posición al norte, esta cambia a sur), al mismo tiempo que afirma el hecho *Actualizar*, permitiendo así que se active la regla de actualización correspondiente al movimiento realizado, ya que, a efectos prácticos, la regla de actualización se activará como si se hubiera activado anteriormente alguna regla de MoverX.

3.2 Reglas para evitar a la serpiente.

En caso de que lo que se perciba sea el siseo de la serpiente (*Sound*), además de volver a la casilla anterior se disparará la flecha. Es decir, Cuando Willy escucha un siseo de la serpiente, huye aterrorizado y de los nervios dispara hacia atrás sin mirar.



Definición gráfica de regla Serpiente

El inconveniente de usar esta estrategia es que hay tan solo un 25% de probabilidad de acertar al blanco. Esto presenta un problema en algunos mapas en los que matar a la serpiente supone el poder llegar al *Oasis*.

3.3 Reglas para evitar peligro en caso de notar *Tremor+Sound*.

En un principio se pensó que con las reglas de *Temblor* y *Serpiente* se esquivarían todos los peligros, pues bastaría con percibirlos para dar la vuelta. Sin embargo en algunos mapas esto no era así. Para resolver esta situación, se han creado las reglas *TemblorSerpiente* y *SerpienteTemblor*, cuyo funcionamiento es muy similar, consiguiendo que Willy no muera nunca a causa de peligros. Esto incrementa enormemente la posibilidad de completar mapas, ya que aunque no muera la serpiente, siempre se va a poder evitar cruzarse con ella o con alguna arena movediza. Sin embargo, algunos siguen sin poder completarse debido al problema que suponen las reglas de movimiento a la hora de pasar por pasillos de peligros, como se comentó anteriormente.

4. Regla para evitar muerte por agotamiento.

La regla *Agotado* evita que Willy muera por exceso de movimientos. Esta regla se activará cuando Willy realice 999 movimientos, lo cual se puede conocer gracias a *contador*, el cual se va incrementando con las reglas *Actualizar* tras realizar un movimiento tanto al avanzar como al evitar peligros.