

# Practica 1 Inteligencia Artificial: Agente Reactivo

Por: Arturo Cortés Sánchez

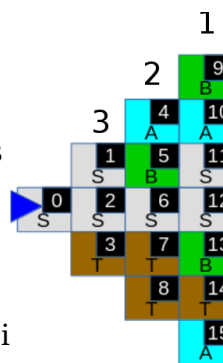
Para empezar he definido una serie de variables que me ayudan a llevar el control sobre el agente y tres matrices que guardan su camino y visión. También he definido una serie de funciones:

avanza(): Decide si avanzar o girar en función de la matriz de prioridades. Si no está orientado mirará la matriz de prioridades temporal (camino\_memoria), en caso contrario la matriz permanente (camino\_mapaRes). Para ello mira delante y a los lados del agente en la matriz correspondiente, e intenta ir hacia el valor mas bajo.

bloquear(): Esta función esta diseñada para llamarse antes de que se decida girar. En la matriz de prioridades correspondiente aumenta en uno el valor de la posición que esté delante del agente.

gira(): Similar a la función avanza() pero solo devuelve girar en una dirección u otra. Mira a los laterales del agente en la matriz de prioridades correspondiente y gira hacia el mas bajo. En caso de ser iguales decide aleatoriamente.

ver(): Esta función se encarga del dibujado del mapa, ya sea en mapaResultado o en una matriz auxiliar (memoria) si no está orientado. Para ello mira en que dirección apunta el agente y en función de ella lanza tres bucles (por las tres lineas de la representación del vector ejecución) además hace una ultima escritura en la matriz de la posición del jugador (sensores.terreno[0]). Antes de toda escritura en una matriz comprueba que está dentro de los limites.



sacar\_objeto(): una función sencilla usada para sacar objetos de la mochila. Mira si la ultima acción ha sido meter un objeto en la mochila, en ese caso saca un objeto, en caso contrario mete el que tiene en la mochila

La función think comienza con dos fragmentos de código que se lanzan en la primera iteración. El primero rellena los bordes del mapa con precipicios y el segundo inicializa la matriz auxiliar. Después hay un par de bucles anidados que se dedican a borrar las interrogaciones del mapa sustituyéndolas por el carácter anterior, estas líneas se activan en la ultima iteración.

A continuación viene el control de coordenadas, si la ultima acción fue avanzar y no está bloqueado, aumenta o decrementa la coordenada correspondiente en función de la orientación. Posteriormente se realiza el control de muertes, cuando el sensor reset se activa vacía las matrices temporales, pone las coordenadas a 100, la orientación apuntando al norte y los booleanos de control a false.

En el momento en el que el agente se situá sobre un GPS y está desorientado realiza una copia de las matrices temporales sobre las permanentes, asigna las coordenadas correctamente y activa el booleano que controla si está orientado. Después se aumenta la variable tiempo para llevar un control de la iteración actual y se llama a la función ver() para dibujar el mapa.

Si el agente tiene delante una casilla bosque, agua o puerta, verificará si tiene el objeto correspondiente( zapatillas, bikini o llave correspondientemente) si lo tiene en la mano, avanzará o lo usará en el caso de la llave, si lo tiene en la mochila lo sacará. Si no tiene el objeto en cuestión, llamará a la función bloquear y girará. En caso de tener un precipicio o un muro delante simplemente lo bloqueará y girará, si tiene un GPS avanzará y si tiene cualquier otro terreno llamará a la función avanza(). Posteriormente se realiza una comprobación de que no hay lobos o aldeanos cercanos al agente, si los hay los bloqueará y girará.

En caso de tener un objeto delante y no haber girado, el agente comprobará si ya tiene uno igual, si no lo tiene y tiene la mano vacía lo cogerá ,y si tiene algún objeto en la mano, primero lo guardará en la mochila y en la siguiente iteración lo cogerá. En caso de tener el objeto, llamará a la función `bloquear()` y girará. Si lo que tiene delante es un aldeano o un lobo, girará y lo bloqueará. Además, si el agente tiene un hueso en la mano, lo tirará, esto se hace para que siempre tenga un hueco libre en la mochila y poder buscar objetos fácilmente.

Si la variable `girar` es verdadera (la cual es usada a lo largo de la función `think` para indicar que el agente quiere girar) y no se ha girado con `avanza()` entonces se llama a la función `gira()`. Por ultimo se almacena la ultima acción en una variable para que esté disponible en la siguiente iteración.