

Práctica 1:

Agentes reactivos

Para empezar voy a explicar las variables que he usado para definir el comportamiento del agente:

- `fil`: indica la fila en el mapa
- `col`: indica la columna en el mapa
- `brujula`: indica nuestra orientación en el mapa (norte, sur, este, oeste)
- `ultimaAccion`: indica la última acción que hemos realizado
- `bien_situado`: indica si ya hemos pisado un PK y sabemos seguro donde estamos situados
- `girar_derecha`: indica si ha girado a la derecha
- `proxima_accion`: indica la próxima acción que debe hacer el agente
- `mi_mapa_auxiliar`: mapa que utilizo cuando ya estoy bien situado para utilizar la técnica de pulgarcito
- `mi_mapa_auxiliar_no_bien_situado`: mapa que utilizo cuando no sé seguro donde estoy para utilizar la técnica de pulgarcito
- `tiene_botas`: indica si el personaje tiene unas botas, ya sea en la mano o en la mochila
- `tiene_llave`: indica si el personaje tiene una llave, ya sea en la mano o en la mochila
- `tiene_bikini`: indica si el personaje tiene un bikini, ya sea en la mano o en la mochila
- `tiene_hueso`: indica si el personaje tiene un hueso, ya sea en la mano o en la mochila
- `rellenar_mapa`: esta variable la utilizo para rellenar el mapa justo al iniciar el programa
- `tamano_mapa`: indica el tamaño del mapa
- `va_a_por_PK`: indica que el agente ha localizado un PK y va a ir a por él
- `va_a_por_objeto`: indica que el agente ha localizado un objeto que aún no tiene y va a ir a por él

Explicando la técnica usada

Lo primero que hago es inicializar las tres últimas filas visibles al norte, sur, este y oeste a 'P' (Precipicio) ya que es lo único que sabemos seguro de este mundo. Además, también inicializo el resto de *mapaResultado* a 'T' (tierra) para acertar en las zonas que tengan tierra.

Este agente reactivo lo primero que hace es, en caso de que no haya pasado por ningún PK, ver si tiene alguno a la vista. En caso de que lo tenga, va a ir a por él. En caso de que haya pasado por un PK, va a ir en busca de objetos. Si no tiene objetos cerca, va a actuar dependiendo de lo que tenga en frente y va a intentar ir hacia zonas que aún no haya explorado (técnica de pulgarcito)

Además, el agente nunca va a tener objetos repetidos en la mochila, siempre va a tener uno en la mano y tres en la mochila como máximo. De esta forma, el cuarto hueco de la mochila lo utilizará para guardar el objeto de la mano y sacar otro de la mochila. Así nunca habrá problema de espacio y no tendrá que tirar objetos.

Explicando el código

Voy a señalar las líneas sobre las que estoy explicando el código para así poder hacer un poco más fácil el leer la documentación y ver/entender el código a la vez.

- Líneas 8-9: si el sensor de reseteo se ha activado, indica que el agente ha muerto y hay que reiniciar sus variables. Las variables se reinician al valor que tenían asignado en el constructor.

- Líneas 20-48: como ya indiqué en las variables del comportamiento del agente, *rellenar_mapa* sólo está a *true* una vez, justo al iniciar el programa. Contiene bucles *for* que van a inicializar *mapaResultado* con todo a 'T' (tierra) y las tres últimas filas visibles al norte, sur, este y oeste a 'P' (precipicio). Al final se cambia *rellenar_mapa* a *false* para no entrar más en este *if*.
- Líneas 51-82: este *switch* utiliza *ultimaAccion* como argumento:
 - Si la última acción fue avanzar hacia delante, según la orientación que marque nuestra brújula vamos a incrementar o decrementar filas o columnas.
 - Si la última acción fue girar a la derecha, cambiaremos la orientación que marca la brújula.
 - Si la última acción fue girar a la izquierda, también cambiaremos la orientación de la brújula.
- Líneas 85-89: si la casilla en la que estamos situados es un PK y aún no estamos bien situados, actualizamos nuestras filas y columnas para que coincidan con las reales y cambiamos *bien_situado* a *true*.
- Líneas 92-113: si estamos bien situados, vamos a guardar en *mapaResultado* las casillas que el sensor terreno nos permite ver. Obviamente, las filas y columnas que tendremos que actualizar van a depender de si estamos al norte, sur, este u oeste. Para ello, utilizo un *switch* que llama a una función que va a cambiar *mapaResultado*. Además, también actualizo *mi_mapa_auxiliar* para guardar la casilla en la que me encuentro y la que tengo justo delante. En caso de que tenga un muro a los lados de la casilla que tengo delante, también voy a guardarlo, ya que si es un muro lo doy por explorado porque no me interesa pasar por ahí.
- Líneas 116-117: si no estoy bien situado, guardo en *mi_mapa_auxiliar_no_bien_situado* la casilla en la que estoy situado.
- Líneas 122-345: una serie de sentencias *if/else* explicadas por bloques:
 - Líneas 122-125: si el agente tiene activado *va_a_por_un_objeto*, la acción que hará será la que haya en *próxima_acción* y volverá a poner *va_a_por_un_objeto* a *false*.
 - Líneas 128-131: igual que en las líneas anteriores, pero con la variable *va_a_por_PK*
 - Líneas 134-138: si detecta que hay un PK en la zona izquierda del vector terreno y no está bien situado, va a poner la variable *va_a_por_PK* a *true*, va a ir hacia delante y la próxima acción que va a realizar va a ser girar a la izquierda. Así, cuando gire a la izquierda va a ver de nuevo el PK y va a ir hacia él.
 - Líneas 141-145: igual que las anteriores líneas, pero esta vez cuando el PK está en la zona derecha del vector terreno.
 - Líneas 148-338: si hay algún objeto en la casilla de delante, o a los lados de esta casilla, va a actuar dependiendo de varias cosas:
 - Líneas 151-220: si tengo un objeto inmóvil en la casilla de delante:
 - Líneas 154-155: si la casilla en la que estamos situados es un bosque o agua, vamos a girar a la izquierda para esquivar el objeto
 - Líneas 159-162: si no tenemos ningún objeto aún, vamos a cogerlo y vamos a incrementar el número de objetos que tenemos.
 - Líneas 165-219: si ya tenemos objetos, vamos a actuar dependiendo del objeto que tengamos delante. Voy a explicar el primer caso del *switch*, porque los demás casos son iguales.
En el primer caso, si lo que tenemos delante es un hueso pero ya tenemos un hueso (en la mochila o en la mano) vamos a girar hacia la derecha, pues no nos interesa cogerlo. Si no tenemos un hueso, la acción que vamos a realizar es meter lo que tengamos en la mano en la mochila y programar la siguiente acción para que sea coger el objeto. Vamos a incrementar el número de objetos que tenemos, vamos a poner *tiene_hueso* a *true* y *va_a_por_objeto* también. Para el resto de objetos se sigue el mismo procedimiento.

De esta forma aseguramos que no cogemos objetos repetidos y podemos actuar bien a todas las situaciones que se nos impongan.

- Líneas 223-227: si hay un objeto en el segundo elemento del vector superficie, vamos a ir hacia delante y vamos a poner *va_a_por_objeto* a *true* y la próxima acción que haremos será girar a la izquierda para tener el objeto de frente.
- Líneas 230-234: mismo caso que el anterior pero para un objeto en el cuarto elemento del vector superficie.
- Líneas 236-237: si justo delante hay un aldeano, giramos a la izquierda para esquivarlo
- Líneas 239-247: si justo delante tenemos un lobo:
 - Líneas 240-243: si tenemos en la mano un hueso, se lo damos y decrementamos el número de objetos que tenemos.
 - Líneas 245-246: si no tenemos un hueso en la mano, giramos a la izquierda para esquivarlo.
- Líneas 249-257: si justo delante tenemos una puerta, actuamos de la misma forma que con el lobo pero esta vez si tenemos en la mano una llave.
- Líneas 259-337: si no pasa nada de lo anterior, vamos a actuar dependiendo de si estamos bien situados o no:
 - Líneas 260-322: si estamos bien situados, vamos a utilizar *mi_mapa_auxiliar* para utilizar la técnica de pulgarcito e ir hacia alguna casilla que aún desconozcamos. Si conocemos la casilla que tenemos delante, la de la izquierda y la de la derecha, la acción será ir hacia delante.
 - Líneas 324-336: si no estamos bien situados, vamos a girar a derecha o izquierda dependiendo de un número aleatorio.
- Líneas 340-370: si no tenemos un objeto en la casilla de delante o en las de sus laterales, vamos a actuar dependiendo del terreno que tengamos en la casilla de enfrente. Para ello, en cada caso llamamos a una función. Estas funciones las voy a explicar en el apartado de funciones más adelante.
- Línea 372: asignamos la acción que hemos escogido.
- Líneas 396-438: si no estamos bien situados, vamos a actualizar nuestro *mi_mapa_auxiliar_no_bien_situado* para guardar lo que tenemos en la casilla de delante. La razón por la que no lo hicimos anteriormente junto con *mi_mapa_auxiliar* es porque así en el momento de decidir la acción, la casilla de delante va a estar aún desconocida y la acción será ir hacia adelante. Si no estuviera desconocida, giraría, y prefiero que no sea así. Al igual que al actualizar *mi_mapa_auxiliar*, si tenemos un muro en las casillas *terreno[1]* o *terreno[3]*, vamos a guardarlo también ya que si no la guardáramos, podría ser que el agente girara porque esa casilla está desconocida pero no resultaría eficiente porque va a colisionar con el muro.

Explicando las funciones

- ***ActualizarMapaNorte()*, *ActualizarMapaEste()*, *ActualizarMapaSur()*, *ActualizarMapaOeste()*:** como su nombre indica, estas funciones actualizan *mapaResultado* para guardar todo lo que vemos en el vector terreno y también actualizan *mi_mapa_auxiliar* como he indicado en la anterior sección. Estas funciones sólo se llaman cuando estamos bien situados.
- ***HayPKCercaDerecha()*:** devuelve *true* cuando hay un PK en las casillas de la zona derecha.
- ***HayPKCercaIzquierda()*:** devuelve *true* cuando hay un PK en las casillas de la zona izquierda.
- ***HayPKCercaRecto()*:** devuelve *true* cuando hay un PK en las casillas de delante.
- ***HayObjetoCercaDerecha()*:** devuelve *true* cuando hay un objeto en las casillas de la zona derecha y ese objeto aún no lo tenemos.

- **HayObjetoCercaIzquierda()**: lo mismo que la anterior función pero para la zona izquierda.
- **HayObjetoCercaRecto()**: lo mismo que las dos anteriores funciones pero para las casillas de delante.
- **Reinicio()**: esta función pone las variables de comportamiento del agente al valor que tenían en el constructor. Todas excepto *mi_mapa_auxiliar*, que es el mapa que utilizamos para saber qué casillas hemos pisado cuando estamos bien situados.

Funciones utilizadas para el switch de las líneas 340-370:

- **Arbol()**:
 - Líneas 550-734: si estamos bien situados, vamos a decidir la acción dependiendo de...
 - Líneas 551-613: si tenemos activadas las zapatillas, vamos a utilizar *mi_mapa_auxiliar* para intentar ir a una zona que aún desconozcamos (técnica de pulgarcito). Utilizamos un *switch(brujula)* porque las filas y columnas de *mi_mapa_auxiliar* varían dependiendo del sentido en el que estemos. El orden que seguimos es el siguiente:
 - Si la casilla de delante está desconocida, avanzamos.
 - Si la casilla de la izquierda está desconocida, giramos a la izquierda.
 - Si la casilla de la derecha está desconocida, giramos a la derecha.
 - Si las tres casillas ya las conocemos, avanzamos hacia delante.
 - Líneas 615-621: si no tenemos las zapatillas activadas y la casilla sobre la que estamos situados no es agua, guardamos el objeto que tengamos en la mano y vamos sacando objetos de la mochila hasta encontrar las zapatillas. Si la casilla sobre la que estuviéramos fuera agua, el objeto que tendríamos activo sería el bikini y al guardarlo en la mochila nos ahogáramos.
 - Líneas 623-733: si no tenemos zapatillas en la mochila ni activadas, vamos a utilizar el mismo *switch* que cuando teníamos activadas las zapatillas para utilizar la técnica de pulgarcito, con la diferencia de que esta vez si desconocemos la casilla de delante vamos a girar a la izquierda porque no podemos entrar al bosque. Por otro lado, si conocemos las tres casillas, la acción será decidida aleatoriamente, pero siempre se girará hacia un lado.
 - Líneas 736-920: si no estamos situados, vamos a actuar de la misma forma que si estuviéramos bien situados pero obviamente utilizando *mi_mapa_auxiliar_no_bien_situado* para seguir la técnica de pulgarcito.
- **Agua()**: esta función actúa de la misma forma que la anterior, pero obviamente el objeto que vamos a utilizar es el bikini. Esta vez:
 - Si estamos bien situados y tenemos el bikini activado, intentar ir hacia una casilla desconocida.
 - Si estamos bien situados, tenemos el bikini en la mochila y no estamos situados en un bosque, guardar y sacar objetos hasta sacar el bikini
 - Si estamos bien situados y no tenemos un bikini, intentar ir hacia una casilla desconocida.
 - Si no estamos bien situados, actuamos de la misma forma pero con *mi_mapa_auxiliar_no_bien_situado*.
- **Precipicio()**:
 - Líneas 1334-1476: si estamos bien situados, vamos a ir hacia una casilla desconocida que obviamente no sea la de delante, pues nos caeríamos. Si conocemos las tres casillas, vamos a fijarnos en las casillas de *terreno[5]* y *terreno[7]* para ver si son desconocidas. Si estas son conocidas, vamos a girar hacia derecha o izquierda, dependiendo de un número aleatorio.
 - Líneas 1478-1593: si no estamos bien situados, seguimos la misma técnica que anteriormente pero con *mi_mapa_auxiliar_no_bien_situado*. Además, esta vez si las tres casillas son conocidas, giraremos hacia izquierda o derecha aleatoriamente, puesto que desconocemos el mapa y no sabemos hacia donde nos conviene ir.

- **Muros():** esta función actúa de la misma forma que la anterior.
- **SueloPedregosoArenoso():**
 - Líneas 1602-1670: si no estamos bien situados...
 - Líneas 1603- 1605: si tenemos un PK en alguna de las casillas que obtenemos con el sensor del terreno, ir hacia delante.
 - Líneas 1607-1669: si no tenemos un PK cerca, seguir la técnica de pulgarcito para ir hacia casillas desconocidas, siguiendo el mismo orden que en las funciones anteriores (primero mira delante, luego a la izquierda y luego a la derecha). Si las tres casillas son conocidas, ir hacia delante.
 - Líneas 1673-1743: si estamos bien situados...
 - Si localizamos algún objeto cerca (en el vector superficie), avanzar hacia delante
 - Si no hay un objeto cerca, intentamos ir hacia una casilla desconocida, siguiendo el mismo esquema que en las otras funciones.
- **PuntoReferencia():** esta función simplemente devuelve ir hacia delante porque si tenemos un PK delante, nos conviene pisarlo.
- **Puertas():**
 - Líneas 2017-2018: si tenemos una llave en la mano, seguimos hacia delante. Esto significaría que la puerta ya está abierta, pues si no estuviera abierta y la tuviéramos delante, la hubiéramos abierto con las primeras condiciones que teníamos (no hubiéramos entrado en esta función).
 - Líneas 2020-2026: si tenemos una llave en la mochila, guardar y sacar objetos hasta tener una llave. Esto permite que cuando entremos por una puerta tengamos una llave en mano por si se cierra, ya que a veces al entrar se puede cerrar.
 - Líneas 2028-2040: si no tenemos una llave, decidir aleatoriamente si girar hacia la derecha o hacia la izquierda.