

# Practica 2

## Belkan's World

Javier Gómez Luzón  
2º B

•**void** PasarVectoraMapaCaracteres2(int fila, int columna, char m[100][100], char \*v, int brujula)

Esta función realiza lo mismo que PasarVectoraMapaCaracteres pero está orientada a rellenar la matriz mapa\_solucion\_. La única diferencia es que en la cabecera piden una matriz 100x100.

•**void** Agent::ActualizarInformacion(Environment \*env)

Esta función actualiza la información. Si mueres inicializa todos los mapas a “vacíos” y algunas variables booleanas a false. Esta función solo actualiza x\_ e y\_ y la orientacion\_ si en la acción anterior no nos hemos estropeado contra algo. Y si actualiza hay dos variables booleanas al final (para ver si hemos encontrado los dos PK), si están a true rellena las casillas con valor '?' del mapa\_solucion con terreno, por si al final no lo hemos descubierto, lo rellena con ese valor.

•**void** Agent::ActualizarPulgarcito()

Esta función va poniendo en la posición actual del mapa pulgarcito un numero que se va incrementando cada vez que pone una casilla.

•**bool** Agent::MochilaLlena()

Esta función recorre todos los valores de MOCHILLA\_ y nos dice si todos están ocupados, lo que significará que la mochila está llena.

•**int** Agent::UltimaAccion(int brujula)

Esta función recoge los valores del mapa pulgarcito que hay delante, a la izquierda y a la derecha de nuestro personaje (solo los coge si el personaje puede pasar por ahí) y guarda los números en 3 variables, (si el personaje no puede pasar por dicha casilla, la variable asociada a esa casilla vale -1). Compara los valores y se queda con el menor ( y que no sea -1). Si decide que es mejor ir hacia adelante devuelve un 0 y el personaje después irá hacia adelante. Si decide que es mejor ir por la derecha o por la izquierda (devolverá 1 ó 2) pondrá a true una variable, para que el personaje se gire y de un paso en esa dirección.

Si todos los valores son -1 ( que no puede ir hacia adelante, derecha o izquierda) se activará una variable que indicará que el personaje tiene que dar media vuelta.

•**void** Agent::PasarMapa()

Esta función, con los valores del pk1 y del pk2, las filas y columnas en las que estábamos cuando recogimos esos 2 pk, gira el mapa para orientarlo y poder copiarlo correctamente en mapa\_solucion\_ que será la matriz que será comparada con el mapa original.

Esta función se llama cada vez que encuentres 2 pk (cuando mueres los 2 pk se borran, pero el mapa solucion descubierto no).

·bool Agent::MochilaVacía()

Recorre el vector MOCHILLA\_ y si todos los valores son distintos de los valores de 0 a 9. Si todos son '-', la mochila estará vacía.

Agent::ActionType Agent::Think()

Guardamos en un entero lo que devuelve la función UltimaAccion.

- Si estamos en un PK guardamos su fila y columna y nuestra fila y columna y ponemos pk1 a true para que no volvamos a coger otro pk1.

- Si estamos en un pk y ya hemos cogido otro pk. Guardaremos nuestra fila y columna y la fila y columna del pk solo si son distintas a las del pk1.

- Se llamará a la función que gira la matriz y nos recoloca y activara un booleano para que no vuelva a hacer mas esto hasta que mueras y encuentres otros 2 pk.

Y se entraría en una cadena de if y else if:

- Si estamos cogiendo algo (variable cogiendoalgo a true). Guardamos el objeto que tenemos en la mano en la mochila. Y ponemos todas las variables de giro a false, por si estábamos haciendo un giro y nos hemos encontrado un objeto por casualidad.

- Si podemos coger un objeto con numero 9, 8, 4, 3 (que son los objetos que todos los personajes que dan puntos aceptan, creo). Lo cogemos y volvemos a poner todas las variables de giro a false y ponemos el mapa de objetos con la posición actual a '?', ya que el objeto no está ahí.

- Si estamos dándole algo a un personaje (dandocosas =true) sacamos un objeto de la mochila y ponemos dandocosas a false para que si tenemos más objetos entremos en el else if siguiente.

- Si delante tenemos a un personaje al que le podemos dar objetos y tenemos cosas que dar. Le damos lo que tenemos en la mano y si tenemos cosas en la mochila activamos dandocosas para que entre en el if anterior.

- Si estamos dando el primer paso de dar la vuelta, giramos a la izquierda y activamos el booleano del paso 2.

- Giramos otra vez a la izquierda y tendremos la media vuelta completa.

- Si estamos girando a un lado, avanza hacia adelante.

- Si la función de hacia donde ir nos indico de ir hacia adelante, iremos un paso hacia adelante.

- Igual para derecha e izquierda.

Y al final llamamos a actualizar pulgarcito.