

INTELIGENCIA ARTIFICIAL (2016-2017)
DOBLE GRADO EN INGENIERÍA INFORMÁTICA Y MATEMÁTICAS
UNIVERSIDAD DE GRANADA

Práctica 1

Miguel Ángel Torres López

24 de abril de 2017

Índice

1	Objetivo	3
2	Almacenamiento de información	3
2.1	El tablero real	3
2.2	El tablero provisional	3
2.3	Los tableros real y auxiliar de gravedades	4
3	Comportamiento general	5
4	Casos especiales	6
4.1	Búsqueda de puntos de localización	6
4.2	Objetos en el camino	6
4.3	Entidades en el camino	7
5	Relleno preinformativo y postinformativo	7
5.1	Relleno preinformativo	7
5.2	Relleno postinformativo	7
6	El rey y los regalos.	8
6.1	El algoritmo de búsqueda A estrella.	8
6.2	Búsqueda de reyes	8
6.3	Búsqueda de regalos	8

1. Objetivo

El objetivo de este documento es explicar el algoritmo realizado para la primera práctica, los extraños mundos de Belkan. Para ello empezaremos introduciendo el modelo usado para almacenar la información recogida del mapa, continuaremos describiendo el comportamiento del agente en el caso general y por último veremos algunas restricciones y comportamientos específicos añadidos para mejorar el recorrido. Adicionalmente, he utilizado dos métodos para mejorar el resultado del recorrido, uno de ellos al principio y el otro al final de la ejecución.

Para la práctica 2, es necesario además explicar el algoritmo A estrella y el uso que se le da para buscar y entregar regalos a reyes.

2. Almacenamiento de información

2.1. El tablero real

Como se especifica en la práctica, el problema consiste en dibujar un mapa de la zona sobre una matriz de caracteres¹. El territorio se sabe que, como máximo, tiene 100 casillas de alto y otras 100 casillas de ancho. El jugador tiene un campo de visión triangular hacia la dirección en que está mirando de tres casillas de distancia. En cada turno la primera acción de nuestro agente será observar el campo de visión y, si no está reflejado en el mapa, pasar la información a la matriz que refleja el tablero.

Esta acción es importante hacerla antes de pensar cuál será el movimiento realizado en este paso, ya que de otra forma, el agente tomaría una decisión si saber realmente la situación en la que se encuentra.

2.2. El tablero provisional

Puesto que al inicio de cada vida no sabemos la posición en la que nos encontramos, no podemos trasladar la información de nuestro alrededor al tablero real. Sin embargo, desperdiciar la información hallada hasta encontrar un punto de localización es poco inteligente. Por este motivo, realizamos un tablero provisional para ir manejando la información provisional con unas coordenadas relativas.

Puesto que no sabemos la posición inicial de nuestro agente, el tamaño de este tablero provisional tiene que ser el doble que el real menos uno, para impedir que el agente se quede sin espacio para guardar eventual información. Se adjunta la Figura 2.1 para ilustrar esta selección. En esta figura se supone un mapa real de 3x3,

¹La identificación de cada caracter con su respectivo campo en el mapa real es igual que la especificada en el guión de la práctica

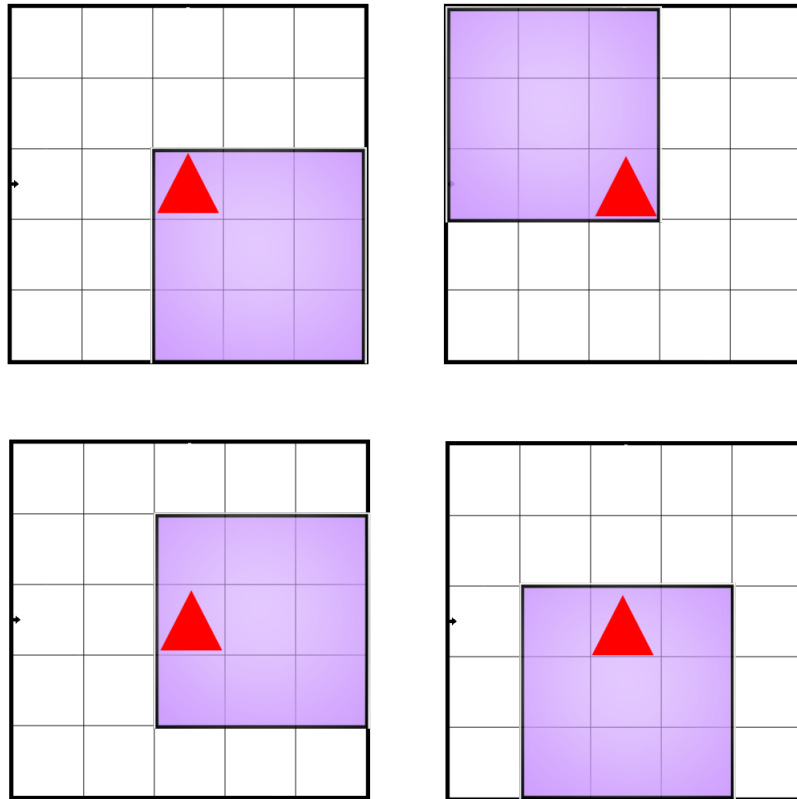


Figura 2.1: Ejemplo de mapas reales sobre mapas provisionales

El agente empezará utilizando el mapa provisional desde la casilla central. Con los tamaños especificados es imposible que se salga del mapa provisional. Cuando el agente pisa un punto de localización y adquiere las coordenadas reales, hace el traspaso de información. Si el mapa real ya estaba escrito en una vida anterior, se sobrescribe, si está bien hecho no actualizará nada y será más eficiente que comprobar si es o no el mismo. A partir de este momento el agente guarda y maneja la información del mapa real.

2.3. Los tableros real y auxiliar de gravedades

Del mismo modo que los tableros anteriores tenemos dos matrices de enteros, cuya utilidad será explicada más tarde. Su tamaño es el mismo que sus respectivos. No obstante, cuando el agente se localiza en el mapa real y se traspasa la información, esta no se sobrescribe, sino que se suma. Luego se explicará con más claridad, pero estos tableros son utilizados para contabilizar la cantidad de veces que hemos pasado por una casilla y, por lo tanto, cuál es el camino menos explorado.

3. Comportamiento general

El movimiento de nuestro agente es decidido por la matriz de gravedades. Al empezar una vida el tablero auxiliar de gravedades está inicializado a 0. A medida que el autómata se mueve actualiza estos valores. Al principio de cada movimiento, después de actualizar el campo de visión, se suma una cantidad a las casillas adyacentes y a la propia. Como casillas adyacentes se entienden que son solo las contiguas, es decir, las casillas que comparten un lado con la casilla en la que nos encontramos. La puntuación² añadida viene dada por:

- Casilla propia: **+2**
- Casilla adyacente con agua, precipicio, bosque, muro o puerta: **+20**
- Casilla adyacente con tierra, suelo o punto de localización: **+1**

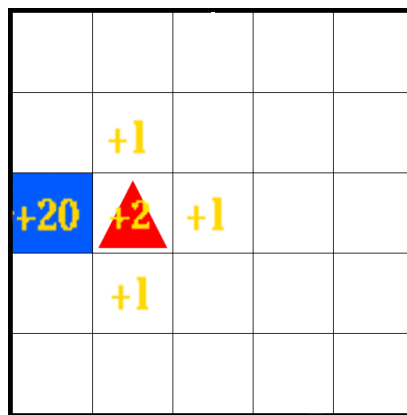


Figura 3.1: Ejemplo de puntuaciones

A continuación el agente elige un dirección a tomar: Norte, Sur, Oeste o Este. Se elige girar hacia la dirección elegida si no es la dirección actual o se elige andar hacia adelante si la dirección actual coincide con la elegida.

La dirección elegida es aquella que tenga menos puntuación en el mapa de gravedades (auxiliar o real, depende del estado de localización). En caso de empate, la dirección tomada es la más cercana a la dirección actual, en caso de un nuevo empate, la dirección tomada es cualquiera de las empatadas.

²Nótese que con la puntuación dada, el agente nunca va a intentar andar hacia un muro, puerta, bosque, agua o precipicio. En caso de manejar algún objeto, este comportamiento puede ser susceptible de alterarse.

4. Casos especiales

Los siguientes casos, en caso de que se cumpla la condición de activación, tienen más prioridad que el comportamiento general descrito en el apartado 3.

4.1. Búsqueda de puntos de localización

En caso de no haber localizado ya las coordenadas reales del agente, este tiene un pequeño truco para alcanzar de forma más directa una casilla con localizador. En cada paso comprueba el tipo de las casillas que hay a su derecha, izquierda y frente, si alguna de estas es un punto de localización, avanza o gira hasta él sin importar la puntuación de la casilla en el tablero de gravedades.

Adicionalmente, comprobamos las casillas que hay en las diagonales más cercanas a la nuestra. Por motivos lógicos, solo hace falta comprobar las casillas diagonales que están más cercanas a nuestra dirección³. En caso de que alguna de éstas sea un localizador, ponemos rumbo hacia ellas. Buscamos avanzar y luego girar hacia ella para avanzar otra vez. Existe la posibilidad de que la casilla frontal sea una casilla inaccesible⁴, en este caso, primero giramos, para luego girar hacia el punto de localización y avanzar de nuevo. Este mecanismo falla en mapas con esquinas complejas, obviemos esta casuística ya que en nuestros mapas no existe.

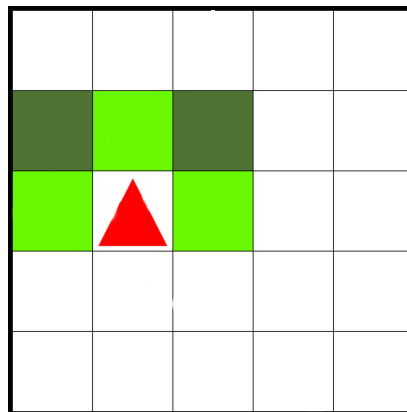


Figura 4.1: Ejemplo de comprobación inteligente de localizadores

4.2. Objetos en el camino

Tras varias pruebas y comprobaciones de resultados he adoptado el comportamiento que mejor se adaptaba a los mapas de los que disponíamos. Aunque a priori pueda parecer que entrar a las habitaciones, a los bosques y al agua es necesario para obtener una buena puntuación no es así. Se pueden sustituir el uso de algunos objetos por algoritmos de rellenado de mapa en la última vida.

³Las casillas del frente a la izquierda y frente a la derecha.

⁴Las casillas inaccesibles son muros y precipicios, depende del manejo de objetos también lo son agua, bosque y puertas.

Sin embargo, no coger los objetos es una mala opción, ya que estos están en ocasiones en posiciones críticas y hacen que cierta parte del mapa sea innaccesible. No se puede caminar por una casilla por la que hay un objeto, luego mi solución para quitarlos ha sido cogerlos y lanzarlos hacia atrás. Las llaves son útiles para entrar a habitaciones cerradas, no obstante, en una buena cantidad de veces, las habitaciones tienen el mismo tipo de suelo que los bordes. El agua suele estar en forma de ríos estrechos y la mayoría de las veces existen pasos para llegar al otro lado del río. Los bosques, por su parte, son bastante más largos y pueden ocultar algo en su interior. Los huesos no merece la pena usarlos para despistar a los lobos.

Al final la práctica adoptada es coger y deshechar los huesos y los bikinis, coger unas botas y deshechar el resto, coger y guardar en la mochila tantas llaves como encuentre. El mecanismo de coger y guardar objetos está realizado, pero por falta de tiempo y la efectividad del algoritmo simple han hecho que no utilice ni las llaves ni las zapatillas. Sería recomendable efectuar un comportamiento para máxima efectividad.

4.3. Entidades en el camino

En ocasiones un aldeano o un lobo se sitúa en una casilla a la que queremos acceder. Puesto que el movimiento de las entidades es errático, tarde o temprano se apartarán del camino. Lo único que hago es esperar a que se aparten y tener en cuenta que, aunque mi acción siga siendo avanzar, no lo estoy haciendo luego no actualizo la posición en el mapa del agente.

5. Relleno preinformativo y postinformativo

5.1. Relleno preinformativo

Antes de comenzar, ya sabemos que los bordes del mapa están llenos de precipicios, en concreto tres filas de precipicios. A pesar de esto, no podemos hacer la acción obvia, pues no sabemos la dimensión del tablero. Luego el relleno a realizar es colocar precipicios en los bordes superior e izquierdo, que siempre serán límites del mapa por empezar a contarse en la esquina superior izquierda.

5.2. Relleno postinformativo

El segundo relleno se realiza en la última vida, cuando el porcentaje de mapa real descubierto es bastante alto y podemos cometer menos fallos. El proceso consiste en rellenar las casillas cuyo tipo desconocemos con el tipo de la casilla adyacente. Tras varias comprobaciones, he determinado que el rellenado se realiza de forma acertada copiando el tipo de la casilla situada al norte. De esta forma, el interior de los bosques que no fuese visible se rellena con bosque y, el interior de las habitaciones se rellena con el tipo encontrado en las casillas del norte de la habitación.

6. El rey y los regalos.

6.1. El algoritmo de búsqueda A estrella.

Este algoritmo está basado en la búsqueda de soluciones dentro de un grafo dando prioridad a los nodos con mayor valor en la función heurística. La función heurística del método usado es la suma de la distancia entre la posición original y la posición del nodo y la distancia estimada con la distancia Manhattan entre la posición del nodo y el destino. El algoritmo con esta heurística tiene la solución óptima asegurada. Es bastante rápido, luego podremos ejecutarlo múltiples veces.

6.2. Búsqueda de reyes

Cuando el agente tiene un regalo en la mano, cada tres pasos intenta encontrar a un rey que se encuentre cerca de él para entregarle el regalo. Por tanto, el agente hace uso de la función de búsqueda cada tres pasos. El algoritmo está preparado para devolver caminos que pasen por el bosque en caso de que el agente ya disponga de unas zapatillas en la mano. En caso de que exista el camino, se actualiza el mapa de gravedades para formar un pasillo de casillas con valores negativos. Esto provocará que el agente siga dicho camino porque serán las casillas con puntuación más baja, es el único momento en el que se permiten valores negativos. Conforme el agente camina por esas casillas con gravedad negativa, las va restableciendo a un valor positivo, más o menos el valor que tuvieran antes. Al llegar al rey se entrega el regalo y deja de comprobarse si estamos pasando por un camino.

Además, tenemos que tener en cuenta un caso peligroso. El camino termina en un bosque y el rey está en tierra pero al lado del bosque. Nuestro agente intentará darle el regalo, pero al quitarse las botas, morirá por estar en un bosque sin botas. Esto provocará no solo que perdamos movimientos, sino que perderemos un regalo a las puertas de ser entregado. Así que, en lugar de dirigir el A estrella para hacer un camino entre la casilla del rey y el agente, lo haremos dirigir un camino entre el agente y la casilla adyacente al rey que no tenga un obstáculo. Esto, añadiendo al rey una casilla negativa, hará que el agente planifique mejor la entrega de regalos.

6.3. Búsqueda de regalos

Los sensores nos aportan un vector con las casillas donde se encuentra los regalos. Por esto, puede ser interesante utilizar el algoritmo de búsqueda para llegar hasta ellos también. No obstante, puesto que tenemos la ubicación real de los regalos, no podremos buscarlos hasta que sepamos en qué casilla real se encuentra el agente, es decir, hasta que hallamos pisado un punto de localización. Seguimos el mismo plan llevado a cabo en la búsqueda de reyes, usar el mapa de gravedades para abrir un camino entre el agente y el regalo.