UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA DE LEGANÉS

Inteligencia Artificial en la Industria del Entretenimiento

Tutorial 2

Grado en Ingeniería Informática

Curso 2017/2018

Autores

Sara Yuste Fernández Alonso	**100330038**
Stefan Borgstein	**100379586**

Contenido

1 Introducción

El presente documento contiene la memoria del desarrollo de una serie de métodos con la finalidad de familiarizarse con las estructura de datos utilizadas para representar información extraída de un mapa del juego *Starcraft*.

1.1 Alcance

En primer lugar, se presenta una breve introducción a los contenidos a tratar a lo largo del documento.
Seguidamente, se procede a responder las cuestiones 4,5 y 6 del enunciado de la sección 3 del tutorial.
En tercer lugar, se describen detalladamente los algoritmos implementados para la realización de los puntos 7,8 y 9 del tutorial.
Por último, se exponen una serie de conclusiones y opiniones personales del grupo de prácticas tras la realización del tutorial.

2 Cuestiones

4.
readMap: lee el mapa del jueg y lo almacena en la clase BWTA, para que el programador pueda manipular el mapa internamente.
analyze: una vez leído el mapa, analiza los elementos que se encuentran en él. Esto permite poder situar las unidades neutrales, el centro de comando, etc.

5.
Position opera a nivel de píxel. Una casilla (tile) tiene tamaño 32 píxeles. Un chockepoint es la intersección entre dos regions, y se caracteriza porque por él solo puede pasar una unidad en un instante de tiempo.

6.
Devuelve la altura del terreno en una determinada posición (en formato TilePosition). Los valores posibles son 0,1,2,3,4,5. Esto es importante ya que solo se puede construir en terreno de una cierta altura.

3 Código implementado

3.1 generateMapMatrix()

Este método debe generar una matriz alfa numérica a partir del mapa del juego. Es llamado en el método `onStart()`, ejecutándose al principio de la partida. Este método hace uso de la variable global `mapMatrix[][]`, un array bidimensional de String.
El funcionamiento de este método es como sigue:
En primer lugar, se recorre el mapa del juego, analizando cada casilla. Dada una casilla determinada, pueden darse dos casos:

- No se puede construir en dicha casilla, comprobado con el método `isBuildable(TilePosition, true)`. Esto puede deberse a:
  - Ya hay un edificio en la casilla: esto se comprueba con el parámetro "true" del método `isBuildable`. En este caso, el valor de la casilla en la matriz será 0.
  - La altura o el terreno lo impiden: comprobado con el método `isBuildable`. En este caso el valor en la matriz también será 0.
  - Hay una mina o una fuente de vespeno: se recorre la lista de unidades neutrales y si la casilla que se está evaluando coincide con la posición de una mina o una fuente de vespeno, la posición en la matriz será una "M", o "V", respectivamente.
- Se puede construir en la casilla. En ese caso, se recorre el mapa avanzando en el sentido positivo del eje x y del eje y, hasta un máximo de 4 posiciones, para saber en qué punto ya no puede construirse desde esa casilla. Este punto se controla con una variable que se utiliza de contador, aumentando el valor de esa variable. Así, al finalizar el bucle que recorre el mapa, se tiene un valor numérico del tamaño máximo de los edificios que se pueden construir desde esa posición.

Una vez calculada la matriz, se escribe en un fichero.

3.2 matrixBuild()

Este método debe encontrar una casilla donde comenzar a construir un edificio dado dicho edificio. Para ello, en primer lugar obtiene el tamaño del edificio mediante el método tileSize(), y almacena la coordenada x de ese tamaño en una variable local (ya que se asumen que lso edificios son cuadrados, tomando la coordenada x como referencia).
Una vez hecho eso, recorre el mapa en círculos tomando como referencia las coordenadas del centro de comando, hasta que encuentre una casilla cuyo valor numérico se corresponda con el tamaño del edificio (almacendo en la variable local descrita anteriormente).
El método devuelve la casilla encontrada.

3.3 updateMatrixDoublePos()

Este método debe actualizar la matriz generada en generateMapMatrix() tras la construcción de un edificio dadas dos posiciones: la posición de la casilla de inicio del edificio (esquina superior izquierda) y la de fin (esquina inferior derecha).
En primer lugar, dado que al iterar sobre la matriz que decfine el mapa se hace a nivel de píxel, es necesario transformar las posiciones obtenidas en coordenadas x e y a nivel de píxel: con los métodos `getX()` y `getY()`, y posteriormente dividiendo entre 4.

Una vez hecho esto, se iterará sobre la matriz del mapa empezando en la posición de inicio calculada, cambiando el valor de los valores de la matriz a 0, tantas veces como la diferencia entre las coordenadas de inicio y fin.

3.4 updateMatrix() (propuesta por el grupo de prácticas)

Este método realiza la misma operación que el método anterior, solo que recibe por parámetro el edificio que se ha construido y la posición de inicio . El número de veces a iterar se obtendrá, en lugar de como la diferencia de posiciones de inicio y fin, como las coordenadas del edificio divididas entre 32.

4 Conclusiones

En este tutorial, al igual que en el anterior, el equipo de prácticas ha encontrado problemas a la hora de trabajar con la plataforma y recursos requeridos. Concretamente, los recursos empleados por Java y Eclipse llegaron a consumir el 20% de la memoria RAM de uno de los equipos empleados, lo que hizo muy difícil el trabajo con dicho equipo. Sin embargo, los problemas encontrados en la propia implementación no han sido un gran impedimento.

Una vez realizado el tutorial 1, el equipo se sentía mucho más cómodo con la plataforma *Starcraft*, por lo que este tutorial ha podido realizarse con más agilidad y facilidad que el anterior.

Durante la realización del tutorial 2, el equipo ha podido familiarizarse con la plataforma y los distintos métodos, consolidando las ideas del tutorial 1. El grupo de prácticas considera que este tutorial es de gran utilidad de cara a la realización de las prácticas futuras en la plataforma *Starcraft*.