Práctica 1. Algoritmos devoradores

Juan Manuel Grondona Nuño
juanmanuel.grondonanu@alum.uca.es
Teléfono: 656485032
NIF: 49193526E

5 de noviembre de 2022

1. Describa a continuación la función diseñada para otorgar un determinado valor a cada una de las celdas del terreno de batalla para el caso del centro de extracción de minerales.

   En este caso la funcion se divide en dos partes.

   - Valoracion por lo centrado que este en el tablero.
   - Valoracion por lo centrado que este con respecto el numero de obstaculos alrededor.

   Para el primer tipo, doy una valoracion positiba a la celda central, y voy disminuyendo segun se aleje. La valoracion mas alta de de las celdas es el de minimo ente la logitud y la altutud del mapa dividada ente 3, de este modo de genera un "cuadrado" entorno a esta celda central y despues de 0, otroga valores negativos.

   Para la segunda parte de la valoracion, comprobamos alredeor de la celda, cuantos obstaculos hay cerca, cuantos mas, mayor valor maximo puede adoptar la casilla, posteriormente miro la diferencia ente el obstaculo as cercano y el obstaculo mas lejano, y en funsion de lo grande que sea la diferecia se llevara o la maxima puntuacion o por el contrario 0, de esta manera me encargo a rasgos generales de que se situe en el centro de estas puedras.

2. Diseñe una función de factibilidad explicita y descríbala a continuación.

   Escriba aquí su respuesta al ejercicio 2.

3. A partir de las funciones definidas en los ejercicios anteriores diseñe un algoritmo voraz que resuelva el problema para el caso del centro de extracción de minerales. Incluya a continuación el código fuente relevante.

```
// sustituya este codigo por su respuesta
void placeDefenses(...) {

    List<Defense*>::iterator currentDefense = defenses.begin();
    while(currentDefense != defenses.end() && maxAttemps > 0) {

        (*currentDefense)->position.x = ((int)(_RAND2(nCellsWidth))) * cellWidth + cellWidth
            * 0.5f;
        ...
        ++currentDefense;
    }
}
```

4. Comente las características que lo identifican como perteneciente al esquema de los algoritmos voraces.

   Escriba aquí su respuesta al ejercicio 4.

5. Describa a continuación la función diseñada para otorgar un determinado valor a cada una de las celdas del terreno de batalla para el caso del resto de defensas. Suponga que el valor otorgado a una celda no puede verse afectado por la colocación de una de estas defensas en el campo de batalla. Dicho de otra forma, no es posible modificar el valor otorgado a una celda una vez que se haya colocado una de estas defensas. Evidentemente, el valor de una celda sí que puede verse afectado por la ubicación del centro de extracción de minerales.

   Escriba aquí su respuesta al ejercicio 5.

6. A partir de las funciones definidas en los ejercicios anteriores diseñe un algoritmo voraz que resuelva el problema global. Este algoritmo puede estar formado por uno o dos algoritmos voraces independientes, ejecutados uno a continuación del otro. Incluya a continuación el código fuente relevante que no haya incluido ya como respuesta al ejercicio 3.

```
// sustituya este codigo por su respuesta
void placeDefenses(...) {

    List<Defense*>::iterator currentDefense = defenses.begin();
    while(currentDefense != defenses.end() && maxAttemps > 0) {

        (*currentDefense)->position.x = ((int)(_RAND2(nCellsWidth))) * cellWidth + cellWidth
            * 0.5f;
        ...
        ++currentDefense;
    }
}
```

Todo el material incluido en esta memoria y en los ficheros asociados es de mi autoría o ha sido facilitado por los profesores de la asignatura. Haciendo entrega de este documento confirmo que he leído la normativa de la asignatura, incluido el punto que respecta al uso de material no original.