

Práctica 1. Algoritmos devoradores

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

10 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 función se divide en dos partes.

- Valoración por lo centrado que esté en el tablero.
- Valoración por lo centrado que esté con respecto el número de obstáculos alrededor.

Para el primer tipo, doy una valoración positiva a la celda central, y voy disminuyendo según se aleje. La valoración más alta de las celdas es el de mínimo ente la longitud y la altitud del mapa dividida ente 3, de este modo se genera un "cuadrado" en torno a esta celda central y después de 0, otorga valores negativos.

Para la segunda parte de la valoración, comprobamos alrededor de la celda, los obstáculos que hay, después esta se valora según la media de estas rocas, cuanto mayor sea esta, mejor puntuación, ya que con un mayor media existe una mayor distancia entre estas, y de esta forma se puede posicionar bien la torreta principal.

Además, según el tamaño del mapa, el valor máximo que pueden aportas estos se ven modificados, de esta manera podemos encontrar la posición más segura que puede estar más alejada.

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

Para la función factibilidad simplemente haremos tres observaciones.

- La defensa no se colocara fuera del mapa.
- La defensa no se colocará donde coincida con algún obstáculo.
- La defensa no se colocará donde coincida con otra defensa.

Si una de estas condiciones se cumpliera, la función devolverá false, y por el contrario devolverá true.

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 código por su respuesta
void placeDefenses(...) {

    List<Defense*>::iterator currentDefense = defenses.begin();
    while(currentDefense != defenses.end() && maxAttempts > 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 código por su respuesta
void placeDefenses(...) {

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