Práctica 1. Algoritmos devoradores

Miguel Angel Chaves Perez
miguelangel.chavesperez@alum.uca.es
Teléfono: 675287145
NIF: 49071750p

November 17, 2016

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.

   Escriba aquí su respuesta al ejercicio 1.

   La función diseñada para determinar el valor de cada una de las celdas para la extracción de minerales es:

   Primero determinar la celda del centro del mapa, y luego, recorro todos los obstáculo y determino cual de ellos están más cercanos al centro del mapa, para determinar que objeto están más cercanos, he optado por la ecuación euclídeas que sumaran la distancia de ese objeto con la celda del centro del mapa, con ello se intenta conseguir el obstáculo mas centrado del mapa.

   Una vez obtenido el obstáculo candidato, empiezo a recorrer todas las celdas, guardando las distancias euclídeas de la posición de esa celda respecto al objeto candidato. Con esto se intenta conseguir que las celdas mas cercana a la roca candidata tenga un valor inferior a las celdas mas alejadas.

   Si he optado por esta estrategia es porque el centro del mapa es el punto más alejado de todos los bordes del mapa y por lo tanto mas alejados desde donde salen los ucos y consecuentemente esos segundos en llegar al centro del mapa sera un tiempo adicionales ganados.

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

   Escriba aquí su respuesta al ejercicio 2.

   La función de factibilidad comprueba si dada una celda candidata se puede colocar, teniendo en cuenta el radio de las defensas y el radio de los objeto.

   Para el caso de la colocación de la defensa extractora, calculamos el radio del extractor y lo diferenciamos con los radios de los objetos teniendo en cuenta no superar la distancias euclídeas de la celda candidata con los diversos objetos.

   Para el caso de la colocación de las defensa , calculamos el radio del las defensas y lo diferenciamos con los radios de los objetos teniendo en cuenta no superar la distancias euclídeas de la celda candidata con los diversos objetos.

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.

```
green!40!blackvoid blackDEF_LIB_EXPORTED blackplaceDefenses(green!40!blackbool** blackfreeCells,
    green!40!blackint blacknCellsWidth, green!40!blackint blacknCellsHeight, green!40!blackfloat
    blackmapWidth, green!40!blackfloat blackmapHeight
               , blackstd::blacklist<blackObject*> blackobstacles, blackstd::blacklist<blackDefense
                   *> blackdefenses)
{

  green!40!blackfloat blackcellWidth = blackmapWidth / blacknCellsWidth; gray!40!gray//
      gray!40!grayCalculamosgray!40!gray gray!40!grayelgray!40!gray gray!40!grayanchogray!40!gray
      gray!40!graydegray!40!gray gray!40!grayunagray!40!gray gray!40!graycelda
```

Figure 1: Estrategia devoradora para la mina

```
  green!40!blackfloat blackcellHeight = blackmapHeight / blacknCellsHeight; gray!40!gray//
      gray!40!grayCalculamosgray!40!gray gray!40!grayelgray!40!gray gray!40!grayaltogray!40!gray
      gray!40!graydegray!40!gray gray!40!grayunagray!40!gray gray!40!graycelda
  green!40!blackfloat** blackvalorCelda = green!40!blacknew green!40!blackfloat*[blacknCellsHeight];
      gray!40!gray//gray!40!gray gray!40!grayPuntuaci ngray!40!gray gray!40!graypuntosgray!40!gray
      gray!40!graylasgray!40!gray gray!40!grayceldas

gray!40!gray  gray!40!gray//gray!40!grayInicializamosgray!40!gray gray!40!graytodasgray!40!gray
      gray!40!graylasgray!40!gray gray!40!grayceldasgray!40!gray gray!40!graydelgray!40!gray
      gray!40!graymapagray!40!gray gray!40!graycongray!40!gray gray!40!grayelgray!40!gray gray!40!grayvalor
      gray!40!gray gray!40!graynulo
  green!40!blackfor(green!40!blackint blacki=0; blacki<blacknCellsHeight; blacki++)
  {
    blackvalorCelda[blacki] = green!40!blacknew green!40!blackfloat[blacknCellsHeight];
    green!40!blackfor(green!40!blackint blackj=0; blackj<blacknCellsHeight; blackj++)
    blackvalorCelda[blacki][blackj] = 0;
  }

  blackList<blackDefense*>::blackiterator blackcurrentDefense = blackdefenses.blackbegin();

gray!40!gray  gray!40!gray//gray!40!grayValoresgray!40!gray gray!40!graydegray!40!gray gray!40!graylas
      gray!40!gray gray!40!grayceldagray!40!gray gray!40!graypuntosgray!40!gray gray!40!graylagray!40!gray
      gray!40!graycolocaciongray!40!gray gray!40!graydelgray!40!gray gray!40!graycentrogray!40!gray
      gray!40!grayextrator
  blackvalorCeldaExtratora(blackvalorCelda,blackcellHeight,blackobstacles,blacknCellsWidth);

  green!40!blackbool blackasignada = green!40!blackfalse;

gray!40!gray  gray!40!gray//gray!40!grayAlgorizmogray!40!gray gray!40!grayvorazgray!40!gray gray!40!graypara
      gray!40!gray gray!40!graylagray!40!gray gray!40!graycolocaciongray!40!gray gray!40!graydegray!40!gray
      gray!40!graylagray!40!gray gray!40!grayceldagray!40!gray gray!40!grayextractora
  green!40!blackwhile(!blackasignada)gray!40!gray//gray!40!graymientrasgray!40!gray gray!40!grayque
      gray!40!gray gray!40!graynogray!40!gray gray!40!graysegray!40!gray gray!40!grayhayagray!40!gray
      gray!40!grayasignadogray!40!gray gray!40!grayelgray!40!gray gray!40!grayvalorgray!40!gray
      gray!40!grayagray!40!gray gray!40!graylagray!40!gray gray!40!grayceldagray!40!gray
      gray!40!grayextratora
  {
    green!40!blackint blackcoorX,blackcoorY;

    blackseleccion(blackvalorCelda, &blackcoorX, &blackcoorY,blacknCellsWidth);gray!40!gray//
        gray!40!graySeleccionamosgray!40!gray gray!40!graycandidatos

gray!40!gray    gray!40!gray//gray!40!graypuntosgray!40!gray gray!40!graymedios
    green!40!blackfloat blackpuntoMedioY = blackcoorY*blackcellHeight + blackcellHeight/2;
    green!40!blackfloat blackpuntoMedioX = blackcoorX*blackcellWidth + blackcellWidth/2;

gray!40!gray    gray!40!gray//gray!40!grayComprobaciongray!40!gray gray!40!graysigray!40!gray gray!40!grayes
        gray!40!gray gray!40!grayposiblegray!40!gray gray!40!grayesagray!40!gray gray!40!graysolucion
    green!40!blackif(blackfactibilidad(blackpuntoMedioX,blackpuntoMedioY,blackobstacles,
        blackdefenses,(*blackcurrentDefense)->blackradio,blackmapWidth))
    {
      (*blackcurrentDefense)->blackposition.blackx = blackpuntoMedioX;
      (*blackcurrentDefense)->blackposition.blacky = blackpuntoMedioY;
      (*blackcurrentDefense)->blackposition.blackz = 0;
      blackasignada = green!40!blacktrue;
    }
  }
}
```

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

   Escriba aquí su respuesta al ejercicio 4.

   Conjunto de candidatos: Cada celdas del mapa Conjunto de candidatos seleccionados: Las celdas con la mayor puntación para la colocación de las defensas Conjunto de solución: Se comprueba que las defensas y la extractora han sido colocadas y no hay mas candidatos, aunque podría ser no óptima conjunto de selección: Se comprueba la primera celda que encuentra de menor valor en el terreno, pudiendo o no, ser la solución óptima. Función de factibilidad: Se comprueba que la defensa o extractor se puede colocar en la celda candidata, si no, se sigue buscando mas candidatos Función objetivo: Maximizar el tiempo transcurridos sin que los ucos invadan la defensa extractora

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.

   Determino el valor de cada una de las celdas con su distancia euclídeas a la posición de la defensa extractora. Por lo tanto una puntación menor sera un buen candidato. Con ello se consigue que la defensa extractora este rodeadas de las demás defensas, y así poder aguantar los ataques de los ucos

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.

```
green!40!blackvoid blackDEF_LIB_EXPORTED blackplaceDefenses(green!40!blackbool** blackfreeCells,
    green!40!blackint blacknCellsWidth, green!40!blackint blacknCellsHeight, green!40!blackfloat
    blackmapWidth, green!40!blackfloat blackmapHeight
               , blackstd::blacklist<blackObject*> blackobstacles, blackstd::blacklist<blackDefense
                   *> blackdefenses)
{

  green!40!blackfloat blackcellWidth = blackmapWidth / blacknCellsWidth; gray!40!gray//
       gray!40!grayCalculamosgray!40!gray gray!40!grayelgray!40!gray gray!40!grayanchogray!40!gray
       gray!40!graydegray!40!gray gray!40!grayunagray!40!gray gray!40!graycelda
  green!40!blackfloat blackcellHeight = blackmapHeight / blacknCellsHeight; gray!40!gray//
       gray!40!grayCalculamosgray!40!gray gray!40!grayelgray!40!gray gray!40!grayaltogray!40!gray
       gray!40!graydegray!40!gray gray!40!grayunagray!40!gray gray!40!graycelda
  green!40!blackfloat** blackvalorCelda = green!40!blacknew green!40!blackfloat*[blacknCellsHeight];
       gray!40!gray//gray!40!gray gray!40!grayPuntuaci ngray!40!gray gray!40!graypuragray!40!gray
       gray!40!graylasgray!40!gray gray!40!grayceldas

gray!40!gray  gray!40!gray//gray!40!grayInicializamosgray!40!gray gray!40!graytodasgray!40!gray
       gray!40!graylasgray!40!gray gray!40!grayceldasgray!40!gray gray!40!graydelgray!40!gray
       gray!40!graymapagray!40!gray gray!40!graycongray!40!gray gray!40!grayelgray!40!gray gray!40!grayvalor
       gray!40!gray gray!40!graynulo
  green!40!blackfor(green!40!blackint blacki=0; blacki<blacknCellsHeight; blacki++)
  {
    blackvalorCelda[blacki] = green!40!blacknew green!40!blackfloat[blacknCellsHeight];
    green!40!blackfor(green!40!blackint blackj=0; blackj<blacknCellsHeight; blackj++)
    blackvalorCelda[blacki][blackj] = 0;
  }

gray!40!gray  gray!40!gray//gray!40!grayAlgoritmogray!40!gray gray!40!grayvorazgray!40!gray gray!40!graydel
       gray!40!gray gray!40!grayejerciciogray!40!gray gray!40!gray3

  blackcurrentDefense++;

gray!40!gray  gray!40!gray//gray!40!grayValoresgray!40!gray gray!40!graydegray!40!gray gray!40!grayceldas
       gray!40!gray gray!40!grayparagray!40!gray gray!40!graydefensas
  blackvalorCeldaDefensas(blackvalorCelda,blackcellHeight,blackdefenses,blacknCellsWidth);

gray!40!gray  gray!40!gray//gray!40!grayAlgorizmogray!40!gray gray!40!grayvorazgray!40!gray gray!40!graypara
       gray!40!gray gray!40!graylagray!40!gray gray!40!graycolocaciongray!40!gray gray!40!graydegray!40!gray
       gray!40!graylasgray!40!gray gray!40!graydefensas
  green!40!blackwhile(blackcurrentDefense != blackdefenses.blackend())gray!40!gray//
       gray!40!graymientrasgray!40!gray gray!40!grayquegray!40!gray gray!40!grayhayagray!40!gray
       gray!40!graydefensas
  {
    green!40!blackint blackcoorX,blackcoorY;

    blackseleccion(blackvalorCelda, &blackcoorX, &blackcoorY,blacknCellsWidth);gray!40!gray//
        gray!40!grayseleccionamosgray!40!gray gray!40!graycandidatos

gray!40!gray    gray!40!gray//gray!40!grayPuntogray!40!gray gray!40!graymedio
    green!40!blackint blackpuntoMedioY = blackcoorY*blackcellHeight + blackcellHeight/2;
    green!40!blackint blackpuntoMedioX = blackcoorX*blackcellWidth + blackcellWidth/2;
```

```
gray!40!gray     gray!40!gray//gray!40!grayComprobaciongray!40!gray gray!40!graysigray!40!gray gray!40!grayes
        gray!40!gray gray!40!grayposiblegray!40!gray gray!40!grayesagray!40!gray gray!40!graysolucion
    green!40!blackif(blackfactibilidad(blackpuntoMedioX,blackpuntoMedioY,blackobstacles,
        blackdefenses,(*blackcurrentDefense)->blackradio,blackmapWidth))
    {
      (*blackcurrentDefense)->blackposition.blackx = blackpuntoMedioX;
      (*blackcurrentDefense)->blackposition.blacky = blackpuntoMedioY;
      (*blackcurrentDefense)->blackposition.blackz = 0;
      blackcurrentDefense++;
    }
  }

  green!40!blackdelete [] blackvalorCelda;

}

green!40!blackvoid blackseleccion(green!40!blackfloat** blackvalorCelda, green!40!blackint *blackcoorX
    , green!40!blackint *blackcoorY,green!40!blackint blacknCelda)
{
  green!40!blackfloat blackcandidato = blackvalorCelda[0][0];

gray!40!gray  gray!40!gray//gray!40!grayInicializamosgray!40!gray gray!40!grayalgray!40!gray
      gray!40!grayinfinito
  blackvalorCelda[*blackcoorX][*blackcoorY] = blackstd::blacknumeric_limits<green!40!blackfloat>::
      blackmax();
  *blackcoorX = 0;
  *blackcoorY = 0;
  green!40!blackfor(green!40!blackint blacki = 1; blacki < blacknCelda; blacki++)
    green!40!blackfor(green!40!blackint blackj = 1; blackj < blacknCelda; blackj++)
    {
      green!40!blackif(blackvalorCelda[blacki][blackj] < blackcandidato)
      {
        *blackcoorX = blacki;
        *blackcoorY = blackj;
        blackcandidato = blackvalorCelda[blacki][blackj];
      }
    }
}
```

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.