Práctica 1. Algoritmos devoradores

Adrian Reyes Alba
adrian.reyesalba@alum.uca.es
Teléfono: 647243014
NIF: 49074565V

19 de noviembre de 2017

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.

   Pues para la defensa extractora, me he creado un cellValueExtractora que pone la extractora en el centro del mapa, dandole valores mayores a las celdas centrales del mapa.

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

   Lo que hago en la función de factibilidad es comprobar 3 casos. Primer caso: que se salga del mapa. Segundo caso: que choque con una defensa ya puesta. Y tercer caso: Que no choque con un obstaculo

```
bool factible(float mapWidth,float mapHeight, std::list<Object*> obstacles, std::list<Defense
    *> defenses,
                Defense* defensa,Cell currentCell, List<Defense*> defensesPutted) {
        List<Defense*>::iterator currentDefense = defensesPutted.begin();
        List<Object*>::iterator currentObstacles = obstacles.begin();
        //Comprobamos que no se salga del mapa
        if((currentCell.position.x - defensa->radio) < 0 || (currentCell.position.x + defensa
            ->radio  >= mapWidth)||(currentCell.position.y - defensa->radio) < 0 || (
            currentCell.position.y + defensa->radio ) >= mapHeight)
                            return false;

        while(currentDefense != defensesPutted.end()){

                if(defensa->id != (*currentDefense)->id){

                        if(_distance((*currentDefense)->position, currentCell.position) <= (*
                            currentDefense)->radio + defensa->radio) //Comprobamos con las
                            defensas

                                return false;

                currentDefense++;
                }
        }

 while(currentObstacles != obstacles.end()){
                if(_distance((*currentObstacles)->position, currentCell.position) <= (*
                    currentObstacles)->radio + defensa->radio) //Comprobamos con los
                    obstaculos
                        return false;
                currentObstacles++;
        }

        return 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.

```
/*PARA LA EXTRACTORA*/
```

```
        cellValueExtractora(cell, nCellsWidth,fil,col);
        while(currentCell != listCells.end() && flag == false) {
                selectCell(cell,nCellsWidth,(*currentCell));
                if(factible(mapWidth,mapHeight,obstacles,defenses,(*currentExtraction),(*
                    currentCell),defensesPutted)) {

                    flag = true;
                    (*currentExtraction)->position.x = (*currentCell).position.x;
                    (*currentExtraction)->position.y = (*currentCell).position.y;
                    (*currentExtraction)->position.z = 0;
                        fil = (*currentCell).id / nCellsWidth;
                        col = (*currentCell).id % nCellsWidth;
                    defensesPutted.push_front(*currentExtraction);


                }

                ++currentCell;
        }
```

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

   1.Conjunto de candidatos: conjunto de celdas. 2.Conjunto de candidatos seleccionados: celdas seleccionadas 3.Función solución: Comprueba que esta colocada la base y las defensas. 4.Función de selección: Seleccionamos las casillas con mayor valor, es decir las que están más en el centro del mapa. Función de factibilidad: si no sale del mapa, no choca con obstaculos y no choca con ninguna defensas. Función objetivo: que la base este rodeada de defensas. Objetivo: Que tarde lo menos posible en destruirse la 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.

   Es igual para la extractora pero lo que hago a parte de darle valores mayores a las celdas centrales, a las celdas adyacentes les pongo un valor mayor, para que despues la funcion seleccion coja esas celdas.

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.

```
/*PARA LAS DEFENSAS*/
        cellValueDefenses(cell, nCellsWidth,fil,col);
List<Defense*>::iterator currentDefense = ++defenses.begin();
        currentCell = listCells.begin();
    while(currentDefense != defenses.end() && currentCell != listCells.end()) {
        selectCell(cell,nCellsWidth,(*currentCell));
                if(factible(mapWidth,mapHeight,obstacles,defenses,(*currentDefense),(*
                    currentCell),defensesPutted)) {
                    (*currentDefense)->position.x = (*currentCell).position.x;
                    (*currentDefense)->position.y = (*currentCell).position.y;
                    (*currentDefense)->position.z = 0;
                    defensesPutted.push_back(*currentDefense);
                        ++currentDefense;

                }

                ++currentCell;

    }
```

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.