

Práctica 4. Exploración de grafos

Carmen del Mar Ruiz de Celis
carmen.ruizdecelis@alum.uca.es
Teléfono: xxxxxxxx
NIF:49565250C

19 de enero de 2020

1. Comente el funcionamiento del algoritmo y describa las estructuras necesarias para llevar a cabo su implementación.

El algoritmo A* utiliza una función de evaluación que consiste en la suma del coste real del camino para llegar al nodo actual desde el nodo inicial, y del valor heurístico del nodo a evaluar desde el actual hasta el final,

Utiliza principalmente dos estructuras de datos auxiliares: los abiertos, ordenados por el valor devuelto por la función de evaluación; y los cerrados, donde se tiene un seguimiento de los nodos visitados.

Para cumplir la condición de la estructura que almacena los nodos abiertos, haré uso de un montículo, tal y como hemos visto en clase. Para los cerrados, basta con un vector.

2. Incluya a continuación el código fuente relevante del algoritmo.

```
float costDefense(Defense* d1){
    return (d1->health*1/5 + d1->attacksPerSecond*1/5 + d1->range*1/5 + d1->damage*1/5
            + d1-> dispersion*1/5);
}

void DEF_LIB_EXPORTED calculateAdditionalCost(float **cellCost, int cellsWidth, int
    cellsHeight,
    float mapWidth, float mapHeight, List<Object *> obstacles, List<Defense *>
    defenses){

    float cellWidth = mapWidth / cellsWidth;
    float cellHeight = mapHeight / cellsHeight;
    List<Defense *>::iterator itDef;
    List<Object *>::iterator itObs;
    Vector3 cellPosition;
    float cost = 0;

    for (int i = 0; i < cellsHeight; i++){
        for (int j = 0; j < cellsWidth; j++){

            cellPosition = cellCenterToPosition(i, j, cellWidth, cellHeight);
            cost=0.0;
            for (itDef = defenses.begin(); itDef != defenses.end(); itDef++){
                float rr= 4* _distance((*itDef)->position, cellPosition) - (*itDef)->
                    radio;
                for (itObs = obstacles.begin(); itObs != obstacles.end(); itObs++){
                    if(_distance((*itObs)->position, cellPosition) - (*itObs)->radio
                        < rr )
                        rr += pow(_distance(cellPosition, (*itObs)->position),3);
                }

                cost +=(rr + costDefense(*itDef));
            }
            cellCost[i][j] = cost ;
        }
    }
}
```

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 esta práctica confirmo que he leído la normativa de la asignatura, incluido el punto que respecta al uso de material no original.