Reporte de DibujarDiagrama

Se debe realizar un programa que muestre un laberinto la cual un personaje busque la distancia mas corta para llegar al otro en estre programa debe estar implementado el algoritmo dijkstra y se debe cargar los nodos de una base de datos hecha

**Algoritmos principales**

-rellena el grafo con nodos

public Grafo relleado() {

Grafo grafo = new Grafo();

for (int i = 0; i < 225; i++) {

grafo.addNodo("" + i, i);

}

logger.info("relleno 225 nodos");

return grafo;

}

-cargas la relaciones que existen entre nodos

public void relaciones(Grafo gra) {

Grafo grafo = gra;

try {

ArcoDAOMysql arc = new ArcoDAOMysql();

Iterator<ArcoDTO> i = arc.seleccionar("1").iterator();

while (i.hasNext()) {

ArcoDTO xx = i.next();

grafo.conectarDoble(xx.getOrigen(),xx.getDestino());

}

} catch (Exception l) {

}

}

-carga el estado de la imágenes q tendrá cada nodo

public void estadoImagen(Grafo gra){

Grafo grafo = gra;

FileReader tr;

try {

File t = new File("estado.txt");

tr = new FileReader(t);

BufferedReader br = new BufferedReader(tr);

String linea = "";

int p=1;

while (br.ready()) {

linea = br.readLine();

grafo.getNodos().get(""+p).setEstadoDibujo(Integer.parseInt(linea));

p++;

}

br.close();

tr.close();

} catch (Exception l) {

}

grafo.getNodos().get("224").setEstadoDibujo(3);

grafo.getNodos().get("0").setEstadoDibujo(2);

}

}

-muestra el recorrido q tiene que hacer

public Vector<Integer> getRecorrido(int de, int a) {//224 0

Vector<Integer> recorrido = new Vector<Integer>();

int fin = a;

while (true){

recorrido.add(fin);

fin = anterior[fin];

if(fin==de)break;

}

return recorrido;

}

-busca la distancia mas corta para llegar a un nodo

public Vector<Integer> distanciaMasCorta(int de, int a) {

reiniciar();

Vector<Nodo<Integer>> porVisitar=new Vector<Nodo<Integer>>();

porVisitar.add(new Nodo<Integer>(""+de,de));

distancia[de]=0;

for (int i = 0; i <porVisitar.size(); i++) {

Nodo<Integer> visitado=grafo.getNodos().get(porVisitar.get(i).getId());//1

Iterator<Nodo<Integer>> i2 = visitado.getConexiones().values().iterator();

visitado.setVisitado(true);

while (i2.hasNext()){

Nodo<Integer> n = i2.next();

if(n.isVisitado()==false){

porVisitar.add(n);

// n.setVisitado(true);

this.calculo(visitado,n);

}

}

}

return this.getRecorrido(de, a);

}

public void calculo(Nodo<Integer> de,Nodo<Integer> a){

anterior[a.getContenido()]=de.getContenido();

int nuevaDistancia=distancia[de.getContenido()]+1;

distancia[a.getContenido()]=distancia[a.getContenido()]>nuevaDistancia?nuevaDistancia:distancia[a.getContenido()];

}

**Comentario**

Haciendo este programa aprendí el manejo del algoritmo dijkstra , a poder graficar cada nodo en una respectiva posición usando mouseListener y a cargar una base de dato a mi progama