**Федеральное агентство связи**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра Информатики

**Отчет по лабораторной работе №3**

по дисциплине «КТП»

**«Алгоритм А\*(A star)»**

Выполнил: студент группы БВТ1802

Охватов Д.Д.

Руководитель:

Полянцева Ксения Андреевна

Москва 2020

**Цель:**

Дописать код алгоритма поиска A\*. Концепция алгоритма довольно проста, начиная с исходного местоположения, алгоритм постепенно строит путь от исходной точки до места назначения, используя наикратчайший путь, чтобы сделать следующий шаг. Это гарантирует, что полный путь будет также оптимальным.

**Код программы:**

import java.util.\*;  
  
public class AStarState  
{  
 private Map2D map;  
 private Map<Location, Waypoint> Opened = new java.util.HashMap<Location, Waypoint>();  
 private Map<Location, Waypoint> Closed = new java.util.HashMap<Location, Waypoint>();  
  
 public AStarState(Map2D map)  
 {  
 if (map == null)  
 throw new NullPointerException("map cannot be null");  
 this.map = map;  
 }  
  
 public Map2D getMap()  
 {  
 return map;  
 }  
  
 public Waypoint getMinOpenWaypoint()  
 {  
 if (Opened.size() == 0) return null;  
 ArrayList<Waypoint> waypoints = new ArrayList<Waypoint>(Opened.values());  
 float mincost = waypoints.get(0).getTotalCost();  
 Waypoint min = waypoints.get(0);  
 for (int i = 1; i < waypoints.size(); i++) {  
 if (waypoints.get(i).getTotalCost() < mincost) {  
 min = waypoints.get(i);  
 mincost = min.getTotalCost();  
 }  
 }  
 return min;  
 }  
  
 public boolean addOpenWaypoint(Waypoint newWP)  
 {  
 if (Opened.get(newWP.getLocation()) == null ) {  
 Opened.put(newWP.getLocation(), newWP);  
 return true;  
 }  
 else  
 {  
 if (Opened.get(newWP.getLocation()).getPreviousCost() >  
 newWP.getPreviousCost()) {  
 Opened.put(newWP.getLocation(), newWP);  
 return true;  
 }  
 }  
 return false;  
 }  
   
 public int numOpenWaypoints()  
 {  
 return Opened.size();  
 }  
  
 public void closeWaypoint(Location loc)  
 {  
 Closed.put(loc, Opened.remove(loc));  
 }  
  
 public boolean isLocationClosed(Location loc)  
 {  
 if (Closed.containsKey(loc)) return true;  
 return false;  
 }  
}

import java.util.Objects;  
  
public class Location  
{  
 public int xCoord;  
 public int yCoord;  
  
 public Location(int x, int y)  
 {  
 xCoord = x;  
 yCoord = y;  
 }  
  
 public Location()  
 {  
 this(0, 0);  
 }  
  
 public boolean equals(Object o) {  
 if (this == o) return true;  
 if (o == null || getClass() != o.getClass()) return false;  
 Location location = (Location) o;  
 return xCoord == location.xCoord &&  
 yCoord == location.yCoord;  
 }  
  
 public int hashCode() {  
 return Objects.*hash*(xCoord, yCoord);  
 }  
}

**Примеры работы программы:**

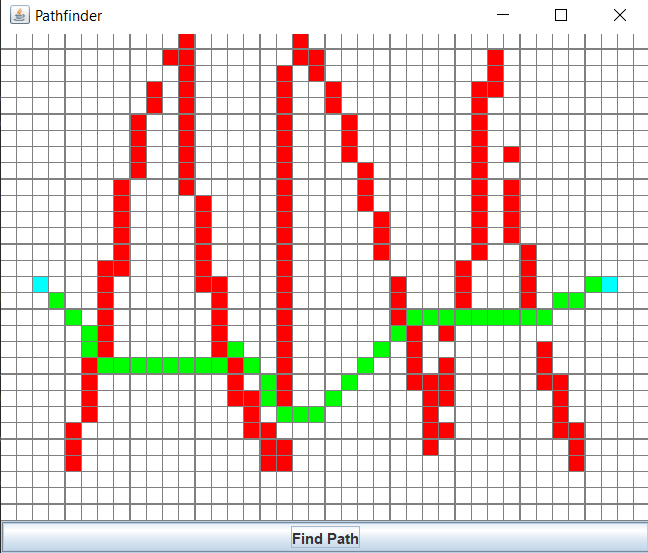


Рисунок 1 – Работа программы (1)

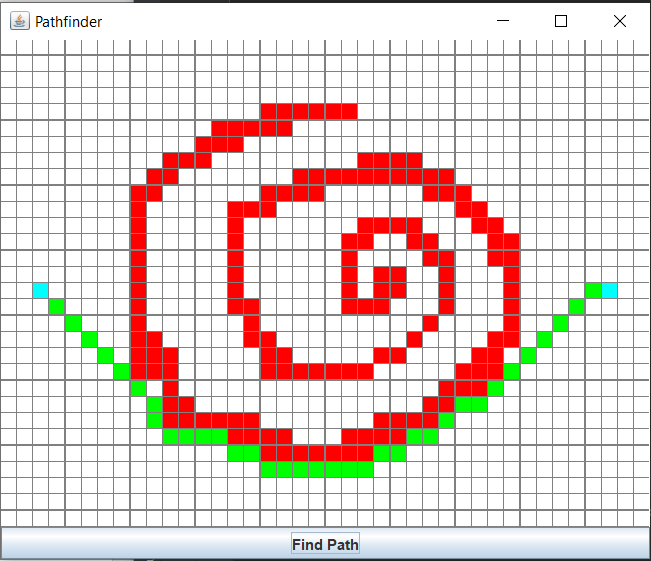


Рисунок 2 – Работа программы (2)