**Федеральное агентство связи**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

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

Кафедра Математической кибернетики и информационных технологий

**Отчет по лабораторной работе**

по предмету «Кроссплатформенные технологии программирования»

на тему:

«Алгоритм А\*»

Выполнил: студент группы

Митрохин Ярослав Игоревич

Руководитель:

Мосева Марина Сергеевна

Москва 2020

*Цель работы:* реализовать некоторые методы алгоритма A\*.

*Ход работы:*

1. *Код методов:*

private HashMap<Location, Waypoint> openWayPoints = new HashMap<Location,Waypoint>(); //Добавление нестатичного поля в класс  
private HashMap<Location, Waypoint> closeWayPoints = new HashMap<Location,Waypoint>();

public Waypoint getMinOpenWaypoint()//Возвращает вершину с наименьшей общей стоимостью  
{  
 if(numOpenWaypoints() == 0)  
 return null;  
  
 Waypoint best = null;  
 float maxValue = Float.*MAX\_VALUE*;  
 for (Waypoint value:openWayPoints.values())  
 {  
 float totalCost = value.getTotalCost();  
 if (totalCost < maxValue)  
 {  
 best = value;  
 maxValue = totalCost;  
 }  
  
 }  
 return best;  
}

public boolean addOpenWaypoint(Waypoint newWP)//Добавить если вершина хуже другой  
{  
 Location location = newWP.getLocation();// Находим location новой точки  
 if (openWayPoints.containsKey(location))//Есть ли в карте заданный ключ  
 {  
 Waypoint currentWP = openWayPoints.get(location);  
 if (newWP.getPreviousCost() < currentWP.getPreviousCost())  
 {  
 openWayPoints.put(location, newWP);//Добавляем элемент в карту  
 return true;  
 }  
 return false;  
 }  
 openWayPoints.put(location, newWP);  
 return true;  
}

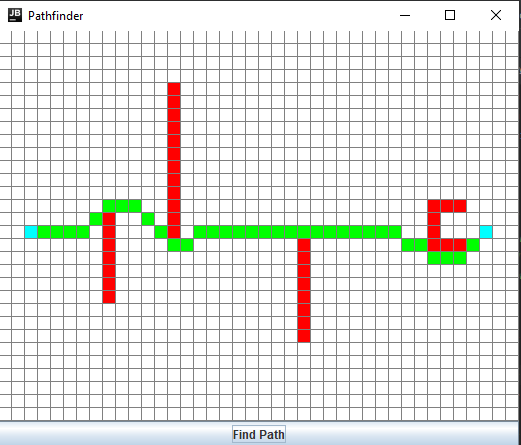
public int numOpenWaypoints()  
{  
 return openWayPoints.size();//возращает размер карты  
}

public void closeWaypoint(Location loc)//перемещение  
{  
 Waypoint waypoint = openWayPoints.remove(loc);  
 closeWayPoints.put(loc, waypoint);  
}

public boolean isLocationClosed(Location loc)  
{  
 return closeWayPoints.containsKey(loc);//Возвращает bool если location уже встречается в вершинах  
}

public boolean equals(Object object) {  
 if (this == object) return true;  
 if (object == null || getClass() != object.getClass()) return false;  
 Location location = (Location) object;  
 return xCoord == location.xCoord &&  
 yCoord == location.yCoord;  
}  
  
public int hashCode() {  
 return Objects.*hash*(xCoord, yCoord);  
}

1. *Результат программы:*

**

*Выводы:* реализовал некоторые методы алгоритма А\*. В конце получен корректный результат программы.