**Федеральное агентство связи**

Ордена Трудового Красного Знамени

федеральное государственное бюджетное образовательное учреждение высшего образования   
«Московский технический университет связи и информатики»

Кафедра «МКиИТ»

Лабораторная работа №3

по дисциплине «Кроссплатформенные технологии программирования»

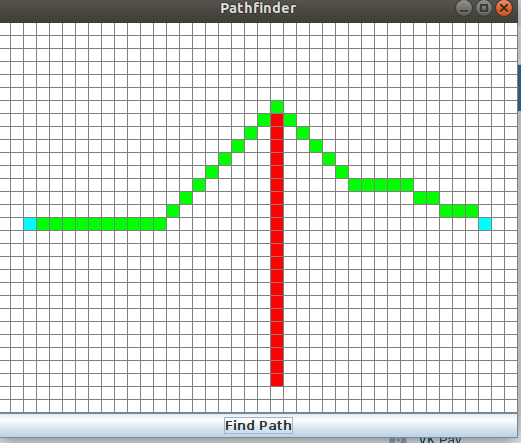
Выполнил: студент

группы БСТ1801

Скоморохов Виктор

Вариант 21

Москва 2020

**1. Задание на разработку программы**

Ознакомиться с алгоритмом A-star

**2. Ход выполнения работы**

Ниже предоставлен код классов Location и AStarState

2.1)Location.java

//

// Source code recreated from a .class file by IntelliJ IDEA

// (powered by Fernflower decompiler)

//

public class Location {

public int xCoord;

public int yCoord;

public Location(int var1, int var2) {

this.xCoord = var1;

this.yCoord = var2;

}

public Location() {

this(0, 0);

}

public int hashCode() {

byte var2 = 1;

int var3 = 31 \* var2 + this.xCoord;

var3 = 31 \* var3 + this.yCoord;

return var3;

}

public boolean equals(Object var1) {

if (this == var1) {

return true;

} else if (var1 == null) {

return false;

} else if (this.getClass() != var1.getClass()) {

return false;

} else {

Location var2 = (Location)var1;

if (this.xCoord != var2.xCoord) {

return false;

} else {

return this.yCoord == var2.yCoord;

}

}

}

}

2.2)AStarState.java

import java.util.HashMap;

import java.util.Map;

public class AStarState {

private Map2D map;

private HashMap<Location,Waypoint> cW = new HashMap<>();

private HashMap<Location,Waypoint> oW = new HashMap<>();

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 (this.numOpenWaypoints() == 0) return null;

Waypoint minCostWay = null;

for (Waypoint element : oW.values())

{

if (minCostWay == null) minCostWay = element;

if (element.getTotalCost()<minCostWay.getTotalCost()) minCostWay = element;

}

return minCostWay;

}

public boolean addOpenWaypoint(Waypoint newWP)

{

Location loc = newWP.getLocation();

if (oW.containsKey(loc))

{

if (oW.get(loc).getPreviousCost()>newWP.getPreviousCost())

{

oW.put(loc, newWP);

return true;

}

else return false;

}

else

{

oW.put(loc,newWP);

return true;

}

}

public int numOpenWaypoints()

{

return this.oW.size();

}

public void closeWaypoint(Location loc)

{

cW.put(loc, oW.get(loc));

oW.remove(loc);

}

public boolean isLocationClosed(Location loc)

{

if (cW.containsKey(loc)) return true;

else return false;

}

}

Результат выполнения задания :

