Лабораторная работа №3

по дисциплине

“Объектно-ориентированное программирование”

Тема:

«Алгоритм A\* («A star»)»

Выполнила студентка

группы БФИ1901

Киселева Анна

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:**

1. **Файл AStarApp.class:**

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class AStarApp {

private int width;

private int height;

private Location startLoc;

private Location finishLoc;

private JMapCell[][] mapCells;

private class MapCellHandler implements MouseListener

{

private boolean modifying;

private boolean makePassable;

public void mousePressed(MouseEvent e)

{

modifying = true;

JMapCell cell = (JMapCell) e.getSource();

makePassable = !cell.isPassable();

cell.setPassable(makePassable);

}

public void mouseReleased(MouseEvent e)

{

modifying = false;

}

public void mouseEntered(MouseEvent e)

{

if (modifying)

{

JMapCell cell = (JMapCell) e.getSource();

cell.setPassable(makePassable);

}

}

public void mouseExited(MouseEvent e)

{

// This one we ignore.

}

public void mouseClicked(MouseEvent e)

{

// And this one too.

}

}

public AStarApp(int w, int h) {

if (w <= 0)

throw new IllegalArgumentException("w must be > 0; got " + w);

if (h <= 0)

throw new IllegalArgumentException("h must be > 0; got " + h);

width = w;

height = h;

startLoc = new Location(2, h / 2);

finishLoc = new Location(w - 3, h / 2);

}

private void initGUI()

{

JFrame frame = new JFrame("Pathfinder");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

Container contentPane = frame.getContentPane();

contentPane.setLayout(new BorderLayout());

// Use GridBagLayout because it actually respects the preferred size

// specified by the components it lays out.

GridBagLayout gbLayout = new GridBagLayout();

GridBagConstraints gbConstraints = new GridBagConstraints();

gbConstraints.fill = GridBagConstraints.BOTH;

gbConstraints.weightx = 1;

gbConstraints.weighty = 1;

gbConstraints.insets.set(0, 0, 1, 1);

JPanel mapPanel = new JPanel(gbLayout);

mapPanel.setBackground(Color.GRAY);

mapCells = new JMapCell[width][height];

MapCellHandler cellHandler = new MapCellHandler();

for (int y = 0; y < height; y++)

{

for (int x = 0; x < width; x++)

{

mapCells[x][y] = new JMapCell();

gbConstraints.gridx = x;

gbConstraints.gridy = y;

gbLayout.setConstraints(mapCells[x][y], gbConstraints);

mapPanel.add(mapCells[x][y]);

mapCells[x][y].addMouseListener(cellHandler);

}

}

contentPane.add(mapPanel, BorderLayout.CENTER);

JButton findPathButton = new JButton("Find Path");

findPathButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) { findAndShowPath(); }

});

contentPane.add(findPathButton, BorderLayout.SOUTH);

frame.pack();

frame.setVisible(true);

mapCells[startLoc.xCoord][startLoc.yCoord].setEndpoint(true);

mapCells[finishLoc.xCoord][finishLoc.yCoord].setEndpoint(true);

}

private void start()

{

SwingUtilities.invokeLater(new Runnable() {

public void run() { initGUI(); }

});

}

private void findAndShowPath()

{

// Create a Map2D object containing the current state of the user input.

Map2D map = new Map2D(width, height);

map.setStart(startLoc);

map.setFinish(finishLoc);

for (int y = 0; y < height; y++)

{

for (int x = 0; x < width; x++)

{

mapCells[x][y].setPath(false);

if (mapCells[x][y].isPassable())

map.setCellValue(x, y, 0);

else

map.setCellValue(x, y, Integer.MAX\_VALUE);

}

}

// Try to compute a path. If one can be computed, mark all cells in the

// path.

Waypoint wp = AStarPathfinder.computePath(map);

while (wp != null)

{

Location loc = wp.getLocation();

mapCells[loc.xCoord][loc.yCoord].setPath(true);

wp = wp.getPrevious();

}

}

public static void main(String[] args) {

AStarApp app = new AStarApp(40, 30);

app.start();

}

}

1. **Файл Location.class:**

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 obj) {

if (obj instanceof Location)

return ((Location) obj).xCoord == this.xCoord && ((Location) obj).yCoord == this.yCoord;

else

return false;

}

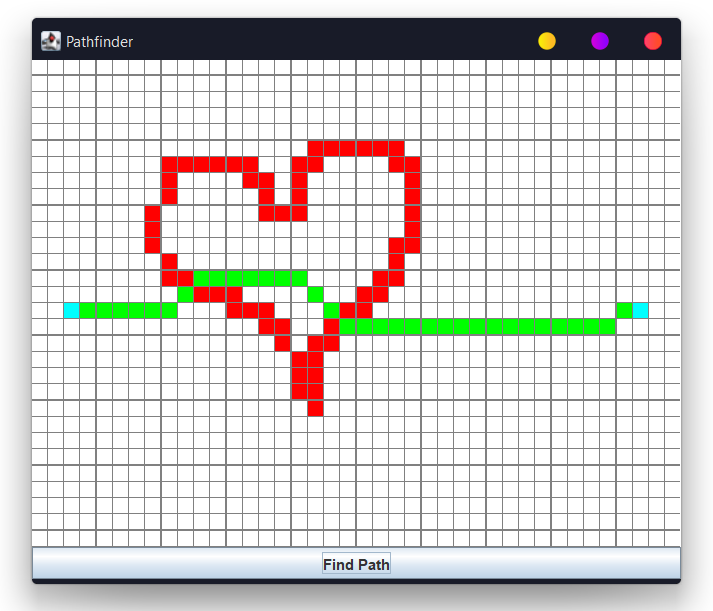
public int hashCode () {

return Objects.hash(xCoord,yCoord);

}

}

**Примеры работы программы:**

****