Artificial Intelligence
CS4242 ONLINE
Menelio Alvarez

Programming Assignment #2.3

RoutFinder

RoutFinder is a simple application which uses the concept of A* search to find the shortest rout through a graph. RoutFinder contains 4 classes:

- **AstarSearch:** This class preforms the actual A* search given the staring location in the form of a Node object.
- **Node:** This class contains all relevant information about a node in the given graph that AstarSearch class will search.
- **Edge:** Contains all the information about edges in a graph that connect nodes
- **GUI:** This class was used to Display the Graph and graphically represent the A* search.

AstarSearch:

```
package RoutFinder;

import java.util.ArrayList;

public class AstarSearch {
	//edges and nodes to search

	//current node in search
	private Node current;
	//goal node
	private Node goal;
	/**<h1>Constructor</h1>
	 * Creates an AstarSearch object based on given paramaters
	 * <p>
	 * @param nodes                                : Node[] array of nodes to
search through
	 * @precondition                          : Starting node must be first node in
nodes and
	 * goal node must be last in nodes
	 * @postcondition                         : an instance of the AstarSearch class
has
	 * been instantiated with nodes
	 *  */
	public AstarSearch(Node[] nodes) {
```

```
		this.current = nodes[0];//starting node must first
		this.goal = nodes[nodes.length-1];//goal node must be last
	}

	/**<h1>Step</h1>
	 * Step through search
	 * <p>
	 * @postcondition                    : The A* search has been advance by one
steps*/
	public void step() {
		if(!current.getName().equals(goal.getName())) {
			//create list possible moves
			ArrayList<Node> moves = new ArrayList<Node>();
			for(int i=0; i< current.getEdges().length;i++) {

	if(!current.getEdges()[i].getFirstNode().getName().equals(current.getName()))
{
				moves.add(current.getEdges()[i].getFirstNode());
			}else {
				moves.add(current.getEdges()[i].getSecondNode());
			}
			}

			//find index lowest fn
			int fn=moves.get(0).getHn()+current.getEdges()[0].getGn();
			int index = 0;
			for(int i=0; i < moves.size();i++) {
				if(fn >
moves.get(i).getHn()+current.getEdges()[i].getGn()) {

	fn=moves.get(i).getHn()+current.getEdges()[i].getGn();
				index =i;
			}
			}
			current = moves.get(index);
		}
	}

	/**<h1>Get Current</h1>
	 * Returns current node
	 * <p>
	 * @return current                   : Node the current node
	 * @postcondition                    : The current node has been returned

	 * */
	public Node getCurrent() {
		return current;
	}
}
```

Node:

```java
package RoutFinder;
public class Node {
	//global Var
	//distance from goal
	private int hn;
	//Edges
	private Edge[] edges;
	//name
	private String name;

	/**<h1>Constructor</h1>
	 * Creats node based on given param
	 * <p>
	 * @param name                          : String name of this distination
	 * @param hn                    : int value of h(n)
	 * @postcondition               : A Node object has been instantiated
	 */
	public Node(String name, int hn) {
		this.name = name;
		this.hn = hn;

	}

	/**<h1>Set Edges</h1>
	 * Set edges array associated with this node
	 * <p>
	 * @param edges                         :Edge[] of this nodes edges
	 * @postconditions              :the edges of this node have been set */
	public void setEdges(Edge[] edges) {
		this.edges = edges;
	}

	/**<h1>Get Edges</h1>
	 * Returns array of edges
	 * <p>
	 * @return edges                : edges[] of edges
	 * @postcondition               : And array of edges has returned
	 * */
	public Edge[] getEdges() {
		return edges;
	}

	/**<h1>Get h(n)</h1>
	 * Gets the hn of this node
	 * <p>
	 * @return hn                   : int of this node's h(n)
	 * @postcondition               : h(n) has been returned
	 * */
	public int getHn() {
		return hn;
	}

	/**<h1>Get name</h1>
```

```
	 * Gets name of this node
	 * <p>
	 * @return name                        : String name of this node
	 * @postcondition                   : String name has been returned
	 * */
	public String getName() {
		return name;
	}
}
```

Edge:

```
package RoutFinder;

public class Edge {
	//cost of traversing edge
	private int gn;
	//first node associated with this edge
	private Node firstNode;
	//first node associated with this edge
	private Node secondNode;

	/**<h1>Constructor</h1>
	 * Constructs object based on given parameters
	 * <p>
	 * @param gn                    : int indicating cost of traversing this edge
	 * @param firstNode             : First Node this edge is attached to
	 * @param secondNode            : Second Node this edge is attached to
	 * @postioncondition            : An Edge object has been instantiated .
	 * */
	public Edge(int gn, Node firstNode, Node secondNode) {
		this.gn = gn;
		this.firstNode = firstNode;
		this.secondNode = secondNode;

	}

	/**<h1>Get firstNode</h1>
	 * Gets the firstNode of this edge
	 * <p>
	 * @return firstNode            : firstNode of this edge
	 * @postcondition               : A node has been returned
	 * */
	public Node getFirstNode() {
		return firstNode;
	}

	/**<h1>Get SecondNode</h1>
	 * Gets the secondNode of this edge
	 * <p>
	 * @return secondNode           : SecondNode of this edge
	 * @postcondition               : A node has been returned
	 * */
	public Node getSecondNode() {
		return secondNode;
	}

	/**<h1>Get g(n)</h1>
	 * Gets the gn of this edges
	 * <p>
	 * @return gn                   : int of this edges's g(n)
	 * @postcondition               : g(n) has been returned
	 * */
	public int getGn() {
		return gn;
```

```
        }
}
```

GUI:

```
// Course: CS4242
// Student name: Menelio Alvarez
// Student ID: 000874829
// Assignment #: 2.3
// Due Date: September 13, 2019
// Signature: _____________________
// Score: _____________________
package RoutFinder;

import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.ListView;
import javafx.scene.layout.AnchorPane;
import javafx.scene.layout.GridPane;
import javafx.scene.shape.Circle;
import javafx.scene.shape.Line;
import javafx.stage.Stage;

public class GUI extends Application {
	//gui elements
	Button step = new Button("step");
	ListView<String> stepList = new ListView<String>();


	//map elements
	Circle[] locs = new Circle[7];
	Line[] edges = new Line[11];

	@Override
	public void start(Stage stage) throws Exception {
		//outer pane
		GridPane outer = new GridPane();
		GridPane control = new GridPane();
		//AnchorPane for map
		AnchorPane map = new AnchorPane();
		//map.setMinSize(1500, 1000);
		//setup map
		map = setupMap(map);

		control.add(step, 0, 0);
		control.add(stepList, 0, 1);
		outer.add(control, 0, 0);
		outer.add(map, 1, 0);

		//initialize nodes
		Node[] nodes= {
				new Node("Start", 11),//0
				new Node("Loc2", 8),//1
				new Node("Loc3", 7),//2
				new Node("Loc4", 6),//3
				new Node("Loc5", 6),//4
```

```
                new Node("Loc6", 4),//5
                new Node("Goal", 0),//6
        };
//setup edges
Edge[] edges = {
                new Edge(3, nodes[0], nodes[1]),//1
                new Edge(2, nodes[0], nodes[2]),//2
                new Edge(2, nodes[1], nodes[3]),//3
                new Edge(1, nodes[2], nodes[3]),//4
                new Edge(7, nodes[1], nodes[2]),//5
                new Edge(4, nodes[2], nodes[4]),//6
                new Edge(5, nodes[3], nodes[4]),//7
                new Edge(3, nodes[3], nodes[5]),//8
                new Edge(2, nodes[5], nodes[4]),//9
                new Edge(3, nodes[5], nodes[6]),//10
                new Edge(6, nodes[6], nodes[4]),//11
};

//set edges in nodes
Edge[] temp1= {edges[0],edges[1]};
nodes[0].setEdges(temp1);

Edge[] temp2 = {edges[0], edges[1], edges[4]};
nodes[1].setEdges(temp2);

Edge[] temp3 = {edges[1], edges[4], edges[3], edges[5]};
nodes[2].setEdges(temp3);

Edge[] temp4 = {edges[2], edges[3], edges[4], edges[7]};
nodes[3].setEdges(temp4);

Edge[] temp5 = {edges[5], edges[6], edges[8],edges[10]};
nodes[4].setEdges(temp5);

Edge[] temp6 = {edges[7], edges[8], edges[9]};
nodes[5].setEdges(temp6);

Edge[] temp7= {edges[9],edges[10]};
nodes[6].setEdges(temp7);

//create A* search object
AstarSearch aSearch = new AstarSearch(nodes);

stepList.getItems().add("Start");
step.setOnAction(e->{
        aSearch.step();

        if(aSearch.getCurrent().getName().equals("Loc2")) {
                locs[1].setFill(javafx.scene.paint.Color.RED);
        }
        if(aSearch.getCurrent().getName().equals("Loc3")) {
                locs[2].setFill(javafx.scene.paint.Color.RED);
        }

        if(aSearch.getCurrent().getName().equals("Loc4")) {
```

```
                locs[3].setFill(javafx.scene.paint.Color.RED);
            }
            if(aSearch.getCurrent().getName().equals("Loc5")) {
                locs[4].setFill(javafx.scene.paint.Color.RED);
            }

            if(aSearch.getCurrent().getName().equals("Loc6")) {
                locs[5].setFill(javafx.scene.paint.Color.RED);
            }

            stepList.getItems().add(aSearch.getCurrent().getName());
        });



        //setup satage
        Scene scene = new Scene(outer);
        stage.setScene(scene);
        stage.show();

    }

    public AnchorPane setupMap(AnchorPane map) {
        map = new AnchorPane();

        map.setMinSize(1500, 1000);

        //setup map
        //edge1
        edges[0] = new Line(0,0,0,300);
        AnchorPane.setTopAnchor(edges[0], 57.50);
        AnchorPane.setLeftAnchor(edges[0], 57.50);
        edges[0].setStroke(javafx.scene.paint.Color.MEDIUMPURPLE);
        edges[0].setStrokeWidth(15);
        map.getChildren().add(edges[0]);

        Label el1 = new Label("edge 1 g(n)= "+3);
        el1.setTextFill(javafx.scene.paint.Color.PURPLE);
        el1.setScaleX(1.50);
        el1.setScaleY(1.75);
        AnchorPane.setTopAnchor(el1, 180.0);
        AnchorPane.setLeftAnchor(el1, 90.0);
        map.getChildren().add(el1);

        //edge2
        edges[1] = new Line(0,0,270,20);
        AnchorPane.setTopAnchor(edges[1], 55.0);
        AnchorPane.setLeftAnchor(edges[1], 57.0);
        edges[1].setStroke(javafx.scene.paint.Color.MEDIUMPURPLE);
        edges[1].setStrokeWidth(15);
        map.getChildren().add(edges[1]);

        Label el2 = new Label("edge 2 g(n)= "+2);
        el2.setTextFill(javafx.scene.paint.Color.PURPLE);
        el2.setScaleX(1.50);
```

```
el2.setScaleY(1.75);
AnchorPane.setTopAnchor(el2, 30.0);
AnchorPane.setLeftAnchor(el2, 190.0);
map.getChildren().add(el2);

//edge3
edges[2] = new Line(0,0,400,10);
AnchorPane.setTopAnchor(edges[2], 350.50);
AnchorPane.setLeftAnchor(edges[2], 57.50);
edges[2].setStroke(javafx.scene.paint.Color.MEDIUMPURPLE);
edges[2].setStrokeWidth(15);
map.getChildren().add(edges[2]);

Label el3 = new Label("edge 3g(n)= "+2);
el3.setTextFill(javafx.scene.paint.Color.PURPLE);
el3.setScaleX(1.50);
el3.setScaleY(1.75);
AnchorPane.setTopAnchor(el3, 330.0);
AnchorPane.setLeftAnchor(el3, 210.0);
map.getChildren().add(el3);

//edge4
edges[3] = new Line(0,0,90,300);
AnchorPane.setTopAnchor(edges[3], 50.50);
AnchorPane.setLeftAnchor(edges[3], 350.50);
edges[3].setStroke(javafx.scene.paint.Color.MEDIUMPURPLE);
edges[3].setStrokeWidth(15);
map.getChildren().add(edges[3]);

Label el4 = new Label("edge 4 g(n)= "+1);
el4.setTextFill(javafx.scene.paint.Color.PURPLE);
el4.setScaleX(1.50);
el4.setScaleY(1.75);
AnchorPane.setTopAnchor(el4, 190.0);
AnchorPane.setLeftAnchor(el4, 440.0);
map.getChildren().add(el4);

//edge5
edges[4] = new Line(300,0,0,270);
AnchorPane.setTopAnchor(edges[4], 105.50);
AnchorPane.setLeftAnchor(edges[4], 35.50);
edges[4].setStroke(javafx.scene.paint.Color.MEDIUMPURPLE);
edges[4].setStrokeWidth(15);
map.getChildren().add(edges[4]);

Label el5 = new Label("edge 5 g(n)= "+7);
el5.setTextFill(javafx.scene.paint.Color.PURPLE);
el5.setScaleX(1.50);
el5.setScaleY(1.75);
AnchorPane.setTopAnchor(el5, 250.0);
AnchorPane.setLeftAnchor(el5, 220.0);
map.getChildren().add(el5);

//edge6
edges[5] = new Line(800,10,0,0);
```

```
AnchorPane.setTopAnchor(edges[5], 80.50);
AnchorPane.setLeftAnchor(edges[5], 330.50);
edges[5].setStroke(javafx.scene.paint.Color.MEDIUMPURPLE);
edges[5].setStrokeWidth(15);
map.getChildren().add(edges[5]);

Label el6 = new Label("edge 6 g(n)= "+4);
el6.setTextFill(javafx.scene.paint.Color.PURPLE);
el6.setScaleX(1.50);
el6.setScaleY(1.75);
AnchorPane.setTopAnchor(el6, 50.0);
AnchorPane.setLeftAnchor(el6, 720.0);
map.getChildren().add(el6);

//edge7
edges[6] = new Line(700,0,0,250);
AnchorPane.setTopAnchor(edges[6], 100.50);
AnchorPane.setLeftAnchor(edges[6], 450.50);
edges[6].setStroke(javafx.scene.paint.Color.MEDIUMPURPLE);
edges[6].setStrokeWidth(15);
map.getChildren().add(edges[6]);

Label el7 = new Label("edge 7 g(n)= "+5);
el7.setTextFill(javafx.scene.paint.Color.PURPLE);
el7.setScaleX(1.50);
el7.setScaleY(1.75);
AnchorPane.setTopAnchor(el7, 210.0);
AnchorPane.setLeftAnchor(el7, 700.0);
map.getChildren().add(el7);

//edge8
edges[7] = new Line(0,0,360,130);
AnchorPane.setTopAnchor(edges[7], 370.50);
AnchorPane.setLeftAnchor(edges[7], 490.50);
edges[7].setStroke(javafx.scene.paint.Color.MEDIUMPURPLE);
edges[7].setStrokeWidth(15);
map.getChildren().add(edges[7]);

Label el8 = new Label("edge 8 g(n)= "+3);
el8.setTextFill(javafx.scene.paint.Color.PURPLE);
el8.setScaleX(1.50);
el8.setScaleY(1.75);
AnchorPane.setTopAnchor(el8, 400.0);
AnchorPane.setLeftAnchor(el8, 700.0);
map.getChildren().add(el8);

//edge9
edges[8] = new Line(320,0,0,410);
AnchorPane.setTopAnchor(edges[8], 100.50);
AnchorPane.setLeftAnchor(edges[8], 835.50);
edges[8].setStroke(javafx.scene.paint.Color.MEDIUMPURPLE);
edges[8].setStrokeWidth(15);
map.getChildren().add(edges[8]);

Label el9 = new Label("edge 9 g(n)= "+2);
```

```
el9.setTextFill(javafx.scene.paint.Color.PURPLE);
el9.setScaleX(1.50);
el9.setScaleY(1.75);
AnchorPane.setTopAnchor(el9, 350.0);
AnchorPane.setLeftAnchor(el9, 995.0);
map.getChildren().add(el9);

//edge10
edges[9] = new Line(0,0,310,400);
AnchorPane.setTopAnchor(edges[9], 505.50);
AnchorPane.setLeftAnchor(edges[9], 850.50);
edges[9].setStroke(javafx.scene.paint.Color.MEDIUMPURPLE);
edges[9].setStrokeWidth(15);
map.getChildren().add(edges[9]);

Label el10 = new Label("edge 10 g(n)= "+3);
el10.setTextFill(javafx.scene.paint.Color.PURPLE);
el10.setScaleX(1.50);
el10.setScaleY(1.75);
AnchorPane.setTopAnchor(el10, 670.0);
AnchorPane.setLeftAnchor(el10, 1020.0);
map.getChildren().add(el10);

//edge11
edges[10] = new Line(0,0,0,800);
AnchorPane.setTopAnchor(edges[10], 90.50);
AnchorPane.setLeftAnchor(edges[10], 1160.50);
edges[10].setStroke(javafx.scene.paint.Color.MEDIUMPURPLE);
edges[10].setStrokeWidth(15);
map.getChildren().add(edges[10]);

Label el11 = new Label("edge 11 g(n)= "+6);
el11.setTextFill(javafx.scene.paint.Color.PURPLE);
el11.setScaleX(1.50);
el11.setScaleY(1.75);
AnchorPane.setTopAnchor(el11, 500.0);
AnchorPane.setLeftAnchor(el11, 1040.0);
map.getChildren().add(el11);

//location1
locs[0] = new Circle();
locs[0].setRadius(40.0);
locs[0].setFill(javafx.scene.paint.Color.CORNFLOWERBLUE);
AnchorPane.setTopAnchor(locs[0], 25.0);
AnchorPane.setLeftAnchor(locs[0], 25.0);
map.getChildren().add(locs[0]);
Label l1 = new Label("Start \nh(n)"+11);
l1.setTextFill(javafx.scene.paint.Color.AZURE);
l1.setScaleX(1.50);
l1.setScaleY(1.75);
AnchorPane.setTopAnchor(l1, 45.0);
AnchorPane.setLeftAnchor(l1, 45.0);
map.getChildren().add(l1);

//location2
```

```
    locs[1] = new Circle();
    locs[1].setRadius(40.0);
    locs[1].setFill(javafx.scene.paint.Color.CORNFLOWERBLUE);
    AnchorPane.setTopAnchor(locs[1], 325.0);
    AnchorPane.setLeftAnchor(locs[1], 25.0);
    map.getChildren().add(locs[1]);
    Label l2 = new Label("Loc 2\nh(n)= "+8);
    l2.setTextFill(javafx.scene.paint.Color.AZURE);
    l2.setScaleX(1.50);
    l2.setScaleY(1.75);
    AnchorPane.setTopAnchor(l2, 345.0);
    AnchorPane.setLeftAnchor(l2, 45.0);
    map.getChildren().add(l2);

    //location3
    locs[2] = new Circle();
    locs[2].setRadius(40.0);
    locs[2].setFill(javafx.scene.paint.Color.CORNFLOWERBLUE);
    AnchorPane.setTopAnchor(locs[2], 50.0);
    AnchorPane.setLeftAnchor(locs[2], 325.0);
    map.getChildren().add(locs[2]);
    Label l3 = new Label("Loc 3\nh(n)= "+7);
    l3.setTextFill(javafx.scene.paint.Color.AZURE);
    l3.setScaleX(1.50);
    l3.setScaleY(1.75);
    AnchorPane.setTopAnchor(l3, 70.0);
    AnchorPane.setLeftAnchor(l3, 345.0);
    map.getChildren().add(l3);

    //location4
  locs[3] = new Circle();
    locs[3].setRadius(40.0);
    locs[3].setFill(javafx.scene.paint.Color.CORNFLOWERBLUE);
    AnchorPane.setTopAnchor(locs[3], 325.0);
    AnchorPane.setLeftAnchor(locs[3], 425.0);
    map.getChildren().add(locs[3]);
    Label l4 = new Label("Loc 4\nh(n)= "+6);
    l4.setTextFill(javafx.scene.paint.Color.AZURE);
    l4.setScaleX(1.50);
    l4.setScaleY(1.75);
    AnchorPane.setTopAnchor(l4, 345.0);
    AnchorPane.setLeftAnchor(l4, 445.0);
    map.getChildren().add(l4);

    //location5
    locs[4] = new Circle();
    locs[4].setRadius(40.0);
    locs[4].setFill(javafx.scene.paint.Color.CORNFLOWERBLUE);
    AnchorPane.setTopAnchor(locs[4], 70.0);
    AnchorPane.setLeftAnchor(locs[4], 1125.0);
    map.getChildren().add(locs[4]);
    Label l5 = new Label("Loc 5\nh(n)= "+6);
    l5.setTextFill(javafx.scene.paint.Color.AZURE);
    l5.setScaleX(1.50);
    l5.setScaleY(1.75);
```

```
        AnchorPane.setTopAnchor(l5, 90.0);
        AnchorPane.setLeftAnchor(l5, 1145.0);
        map.getChildren().add(l5);

        //location6
        locs[5] = new Circle();
        locs[5].setRadius(40.0);
        locs[5].setFill(javafx.scene.paint.Color.CORNFLOWERBLUE);
        AnchorPane.setTopAnchor(locs[5], 470.0);
        AnchorPane.setLeftAnchor(locs[5], 825.0);
        map.getChildren().add(locs[5]);
        Label l6 = new Label("Loc 6\nh(n)= "+4);
        l6.setTextFill(javafx.scene.paint.Color.AZURE);
        l6.setScaleX(1.50);
        l6.setScaleY(1.75);
        AnchorPane.setTopAnchor(l6, 490.0);
        AnchorPane.setLeftAnchor(l6, 845.0);
        map.getChildren().add(l6);

        //location7
        locs[6] = new Circle();
        locs[6].setRadius(40.0);
        locs[6].setFill(javafx.scene.paint.Color.CORNFLOWERBLUE);
        AnchorPane.setTopAnchor(locs[6], 870.0);
        AnchorPane.setLeftAnchor(locs[6], 1125.0);
        map.getChildren().add(locs[6]);
        Label l7 = new Label("Goal \nh(n)"+0);
        l7.setTextFill(javafx.scene.paint.Color.AZURE);
        l7.setScaleX(1.50);
        l7.setScaleY(1.75);
        AnchorPane.setTopAnchor(l7, 890.0);
        AnchorPane.setLeftAnchor(l7, 1145.0);
        map.getChildren().add(l7);



        return map;
    }
    //initializes nodes
    public Node[] setUpNodes() {
        Node[] nodes= {
                new Node("Start", 11),
                new Node("Loc2", 8),
                new Node("Loc3", 7),
                new Node("Loc4", 6),
                new Node("Loc5", 6),
                new Node("Loc6", 4),
                new Node("Goal", 0),
        };
        return nodes;
    }
    //setup edges

    public static void main(String[] args) {
        launch(args);
```

```
        }
    }
```