

MAAS - Street Network Visualization

Group: Commitment_Issues

Sushant Vijay Chavan
Ahmed Faisal Abdelrahman
Abanoub Abdelmalak

January 19, 2019

Contents

1	Introduction	3
2	The Graph Visualization Agent	3
3	Incoming Message Specifications	3
4	Running the Visualization	5

1 Introduction

This document contains a brief overview of the street network graph visualization agent developed by this group. The agent facilitates displaying the street network map and the activity of truck agent in it in a JavaFX GUI. The agent was developed collectively by all team members, and its development involved several stages of additions and adjustments to the code.

2 The Graph Visualization Agent

The graph visualization agent facilitates observing the delivery process of the orders in a graphical way. This agent is capable of displaying the entire street network (consisting of its nodes and edges) and all the trucks present at any given time as shown in figure 1. JavaFX was used to generate the nice display and we used and modified the implementation of **Roland** to generate the graph representation.

The nodes of the graph are displayed as either squares (for important nodes such as bakery or customer locations) or small circles. The nodes are then connected by the edges. The squares are further color coded in order to distinguish the type of node. Green represents customers, red delivery companies and cyan bakery locations. The trucks are represented using triangles on the graph and it is possible to update their location based on their actual position in the graph at any time step. Similar to the nodes, the trucks are also color coded to identify their states. Gray represents idle state, yellow moving towards bakery for collecting boxes and purple delivering boxes to a customer. Additionally, the trucks also display the expected time of arrival along their current route.

This agent communicates with two agents in the delivery stage. The street network agent sends the street network information (consisting of the nodes and the edges) at startup to the graph visualization agent. Similarly, the trucks inform the graph visualization agent about their start positions. Whenever the position of a truck is updated, a new message is sent to the graph visualization agent informing it about the new position, expected time of arrival and state of the truck.

3 Incoming Message Specifications

The graph visualization agent uses the following message formats to communicate with the other two agents.

Message Descriptions:

(Incoming) Street Network Message:

Description: The Graph Visualization Agent receives the street network information from the Street Network Agent in a message of the following format:

Performative: INFORM

Sender: AID of the StreetNetworkAgent

Receiver: AID of the GraphVisualizationAgent

Conversation ID: "graph-visualization"

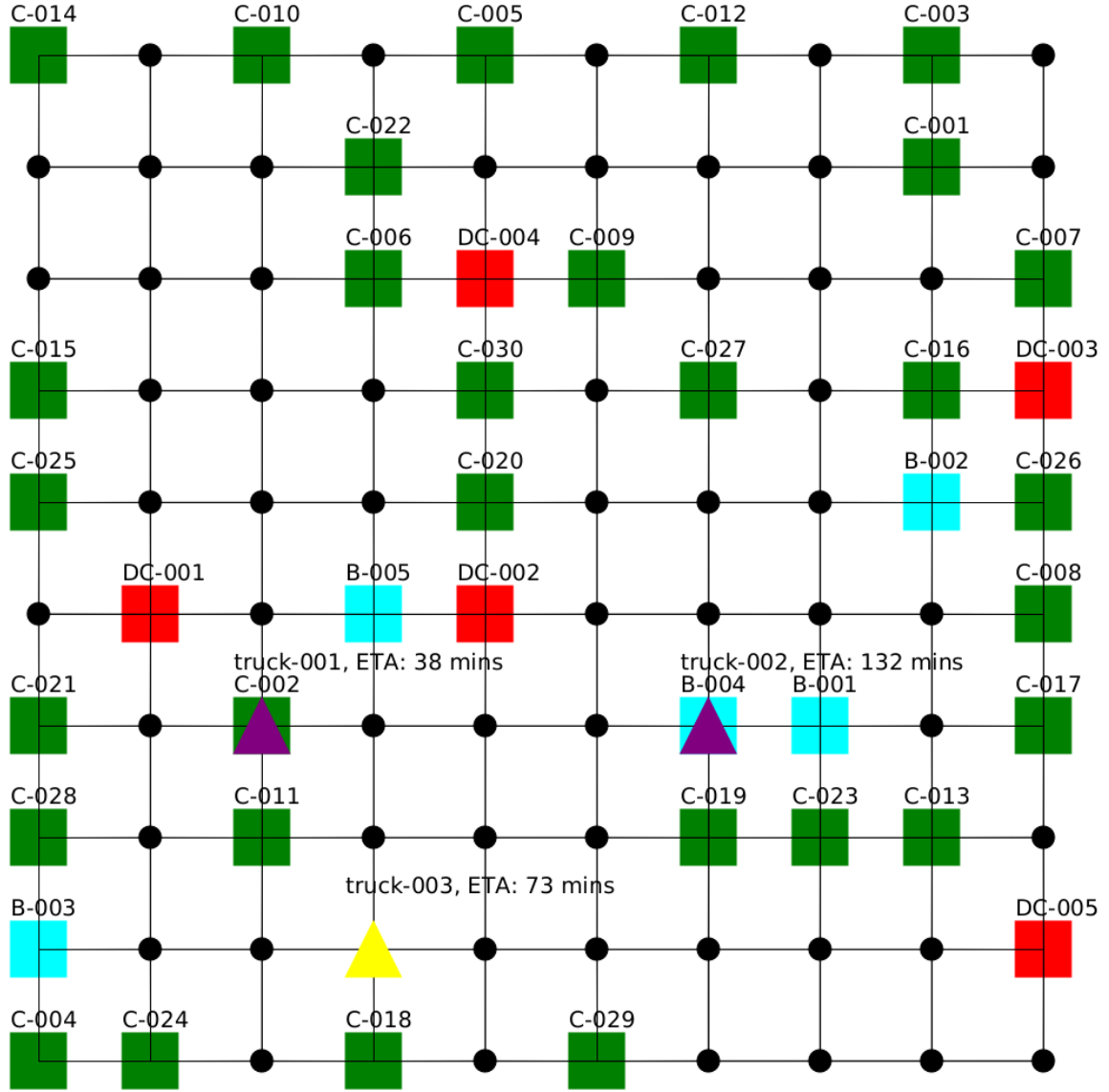


Figure 1: Screenshot of Delivery Stage Visualization

Figure 2 shows an example of the contents of the message.

(Incoming) Truck Message:

Description: The Graph Visualization Agent receives the truck status information from the Truck Agent in a message of the following format:

Performative: INFORM

Sender: AID of the TruckAgent

Receiver: AID of the GraphVisualizationAgent

Conversation ID: "TruckPosUpdate"

Figure 3 shows an example of the contents of the message.

```

{
  "nodes": [
    {
      "guid": "node-014",
      "company": "customer-009",
      "location": {
        "x": 5,
        "y": 2
      },
      "type": "client"
    },
    ...

    {
      "guid": "node-039",
      "company": "delivery-company-004",
      "location": {
        "x": 4,
        "y": 2
      },
      "type": "delivery"
    }
  ],
  "edges": [
    {
      "source": "node-083",
      "target": "node-050"
    },
    ...

    {
      "source": "node-083",
      "target": "node-052"
    }
  ]
}

```

Figure 2: Incoming Street Network Message Example

```

{
  "id": "delivery-company-001_truck-001",
  "x": 5.2,
  "y": 3.6,
  "state": "IDLE",
  "eta": 5
}

```

Figure 3: Incoming Truck Message Example

4 Running the Visualization

The visualization of the delivery phase can be run by simply providing the ‘*-visualization*’ argument to the common *gradle run* command. This automatically sets up the graph visualization agent which then communicates with the street network and the truck agents. In addition, the GUI window appears.

The same commands described in the team's report must be followed:

1. Start the program using the command

```
gradle run --args='-packaging -delivery -visualization'
```

2. If you get an error saying gradle does not support '*args*'. Then use the below command instead to start the program

```
./gradlew run --args='-packaging -delivery -visualization'
```