

# Recycle Swarm – Tutorial

## Introduction

This tutorial describes how to run the Recycle Swarm framework with the simple solution included in the SDK, and how to get started on developing your own solution. It is based on Eclipse IDE, version 3.6 (Helios).

The Recycle Swarm framework uses [JADE](#), a free and open source platform for developing multi-agent systems. Developing a solution requires that you are familiar with JADE programming. JADE comes with extensive [documentation](#), including tutorials, to help you get started.

## Pre-conditions

The following assumes that you are already familiar with [Eclipse IDE](#).

## Running the Framework

There are several methods to run the Recycle Swarm framework:

### 1. Using scripts

You may run one of the ready made scripts, which load the sample scenario and agent:

- `start.bat` for Windows
- `./start.sh` for Linux

### 2. Command line

Enter one of the following commands:

- `java -cp lib/recycleswarm.jar;lib/jade.jar;lib/commons-codec-1.3.jar;lib/log4j-1.2.16.jar;lib/log5j-2.1.2.jar;lib/kd.jar;lib/jade.Boot -gui -agents scenario:org.aimas.wo.recycle swarm.scenario.ScenarioAgent (Scenario, Agent1, Agent2)`

for Windows

- `java -cp lib/recycleswarm.jar:lib/jade.jar:lib/commons-codec-1.3.jar:lib/log4j-1.2.16.jar:lib/log5j-2.1.2.jar:lib/kd.jar:lib/jade.Boot -gui -agents 'scenario:org.aimas.wo.recycle swarm.scenario.ScenarioAgent (Scenario, Agent1, Agent2)'`

for Linux

The parameters for the ScenarioAgent are:

- Scenario – path to scenario XML file; you should make sure that the map file referenced in the scenario file is also accessible
- Agent1 – fully qualified class name for the agents of the first team
- Agent2 – fully qualified class name for the agents of the second team

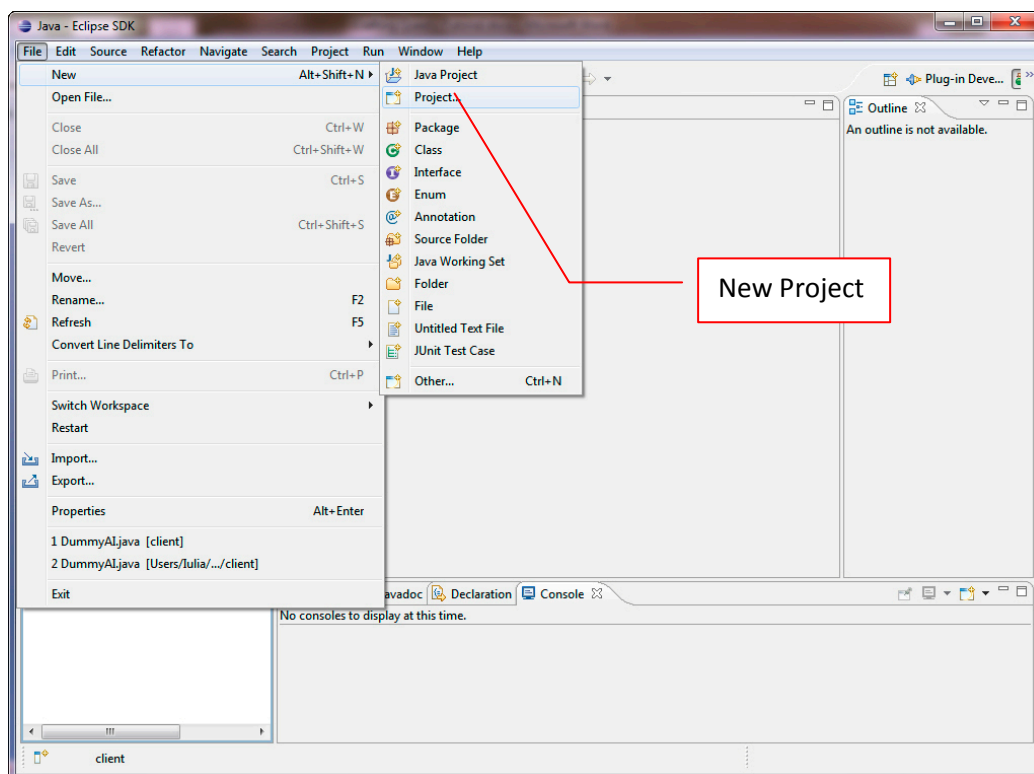
The SDK includes an example scenario (*test1.xml*) and agent (*DummyAgent* from the *org.aimas.vo.recycleswarm.agents* package).

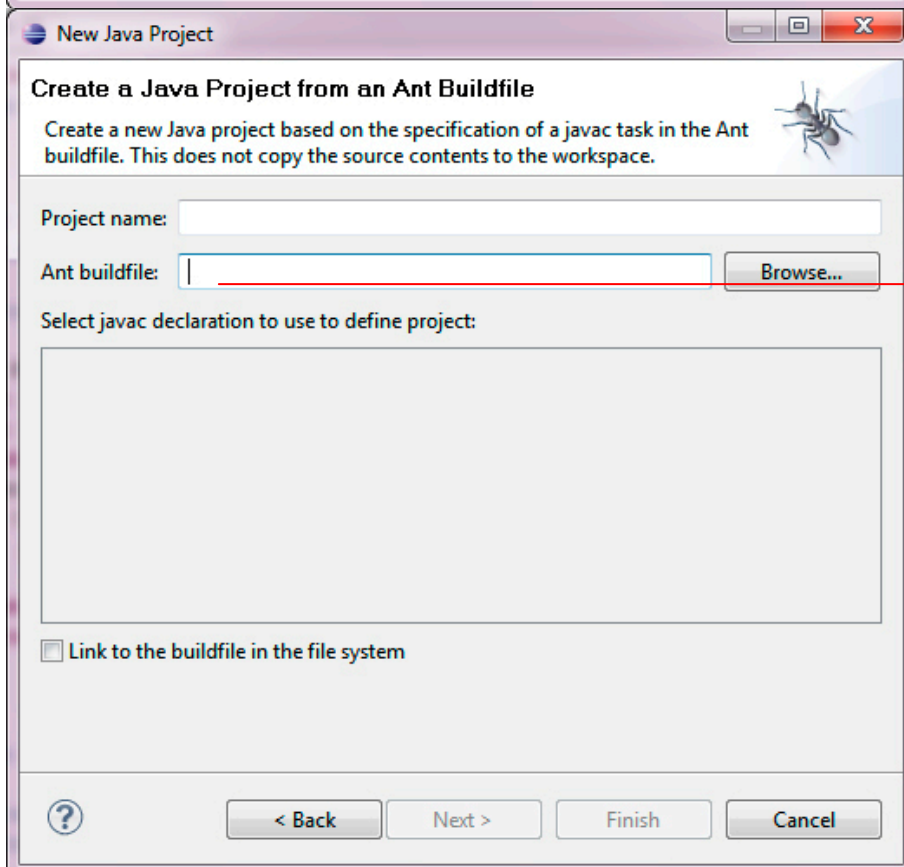
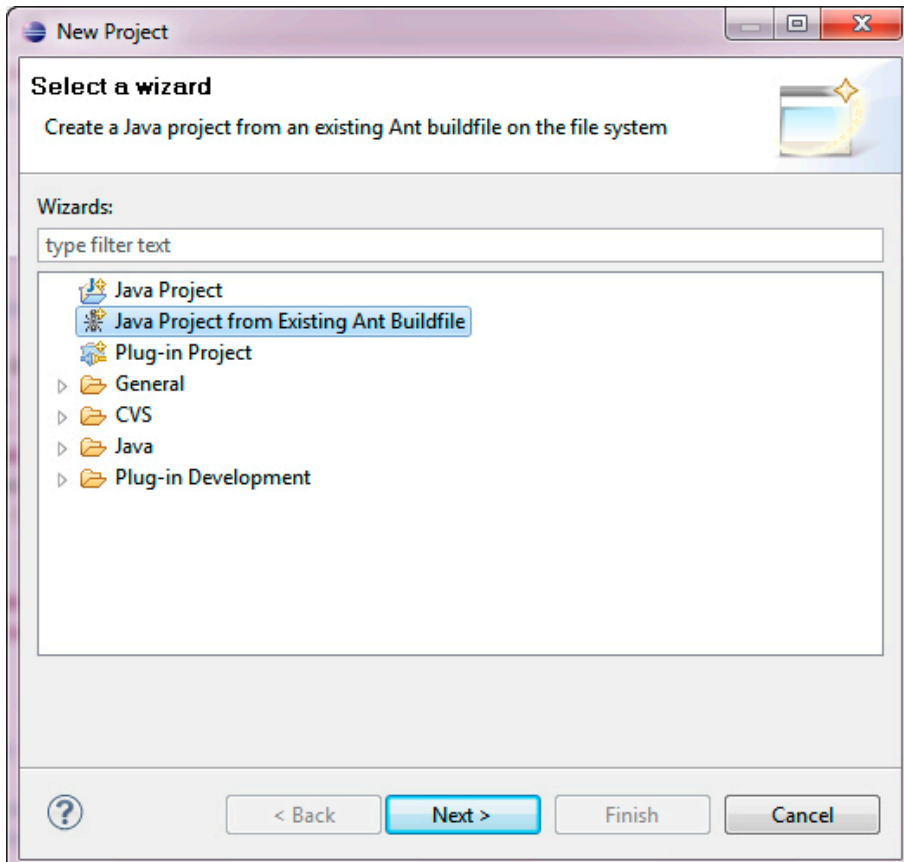
To aid in debugging, the framework uses the [log4j](#) library for logging messages (specifically, the [log5j](#) library, which runs on top of log4j and provides a few more advanced features). You can also use log5j when developing your solution.

You can configure what types of messages you are interested in seeing (eg. debug, info) by adding a log4j.properties file in one of the folders from the classpath of the java application. More information on the log4j.properties file in the log4j [manual](#).

### 3. Eclipse IDE

You can also run the framework by compiling it from source. In Eclipse, create a new Project -> *Java Project from Existing Ant Buildfile*. Set the *build.xml* file from the recycleswarm-framework-r0.zip archive as ant buildfile and press *Finish*.





Insert the path to build.xml here

The advantage of this method is that it allows you to recompile the entire platform and tweak it as you like. Please note, however, that submitted solutions will run on the official Recycle Swarm platform, so your solution should not depend on any tweaks you make.

In order run (as an Application), you should set a *Run Configuration* for the project (*Run -> Run Configurations*, or right click on project and select *Run -> Run Configurations*) having your solution in the ScenarioAgent's arguments. The arguments in the RunConfiguration are:

```
- gui - agents  
scenario: org.aimas.vo.recyclewarm.scenario.ScenarioAgent  
(Scenario, Agent_1, ..., Agent_n)
```

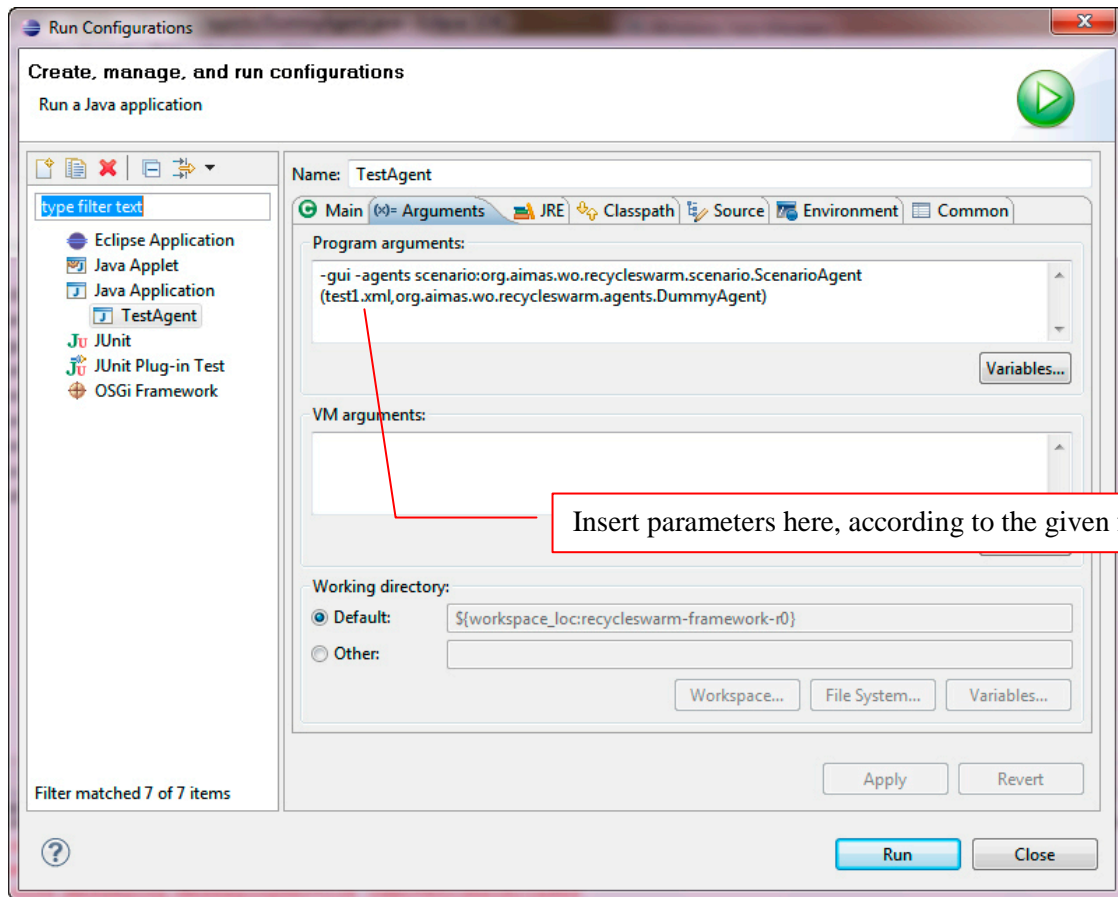
The last part must **not** have any spaces. Make sure the parameters for the agent are separated by commas “ , “ with no space before or after the comma. Also, there should be no space or newline before the opening parentheses.

The parameters for the ScenarioAgent are:

- Scenario – path to scenario XML file
- Agent\_i – fully qualified class name for the agents of the team (one, two, or as many teams as the scenario defines)

You are given an example for both the scenario (*test1.xml*) and agent (*DummyAgent* from the *org.aimas.vo.recyclewarm.agents* package).

The image below contains the arguments for running our Dummy solution.



- **Click *Run***

You should see the Recycle Swarm GUI, and the JADE Remote Agent Management GUI.

When you're done, make sure you completely shut down the JADE platform, either by selecting *File -> Shut down Agent Platform* from the JADE Remote Agent management GUI, or by terminating the process from Eclipse.

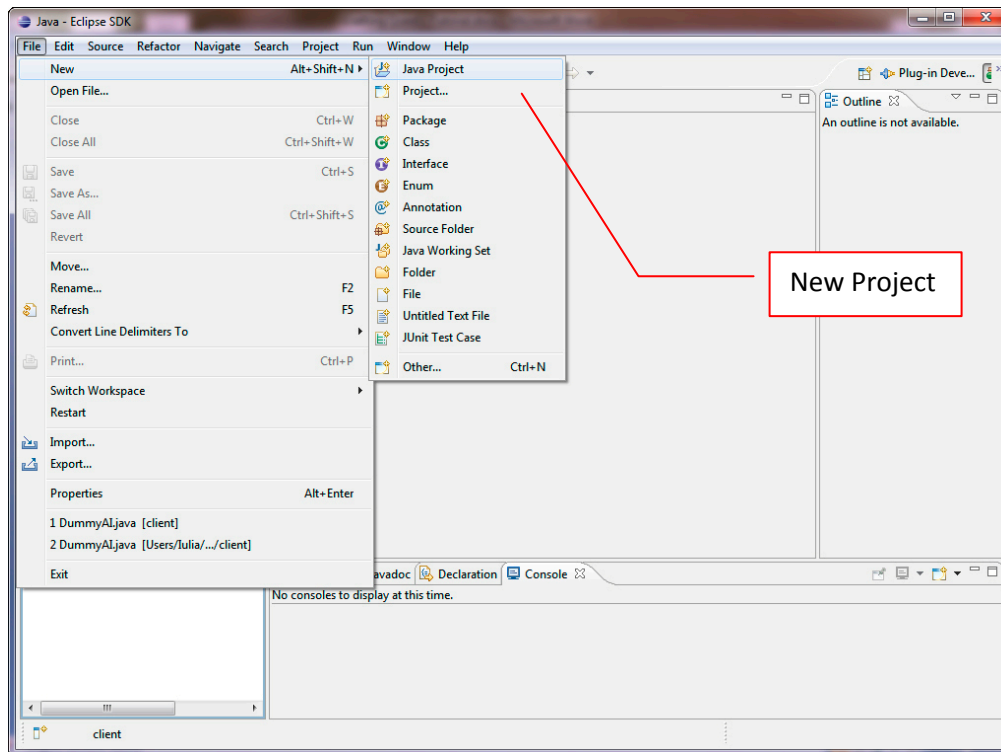
## Submitting Solutions

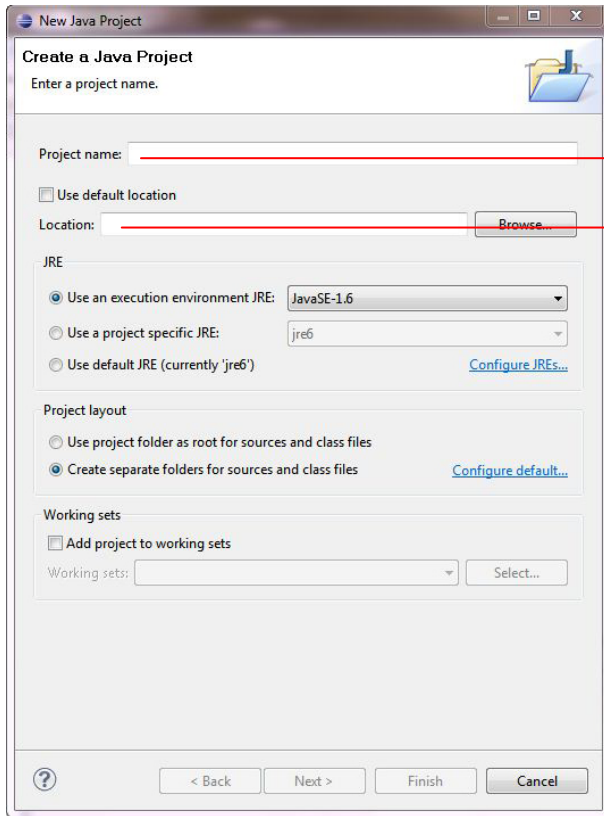
In order to develop and test your solution, you should create a new Java project. Start by creating a new folder on your computer, with the following structure:

- MyTeam
  - lib
    - recycleswarm.jar
    - jade.jar
    - commons-codec-1.3.jar
    - log4j-1.2.16.jar
    - log5j-2.1.2.jar
    - kd.jar
    - log4j.properties
  - test1.xml
  - test1.txt

The *lib* folder is the one from the recycleswarm-framework-r0.zip archive (or the latest release). The other two files, test1.xml (scenario parameters in XML format), and test1.txt (map data), are also from that archive.

Then create a new project in Eclipse, setting the folder you created earlier as default location (uncheck *Use default location* and set the *Location* field).

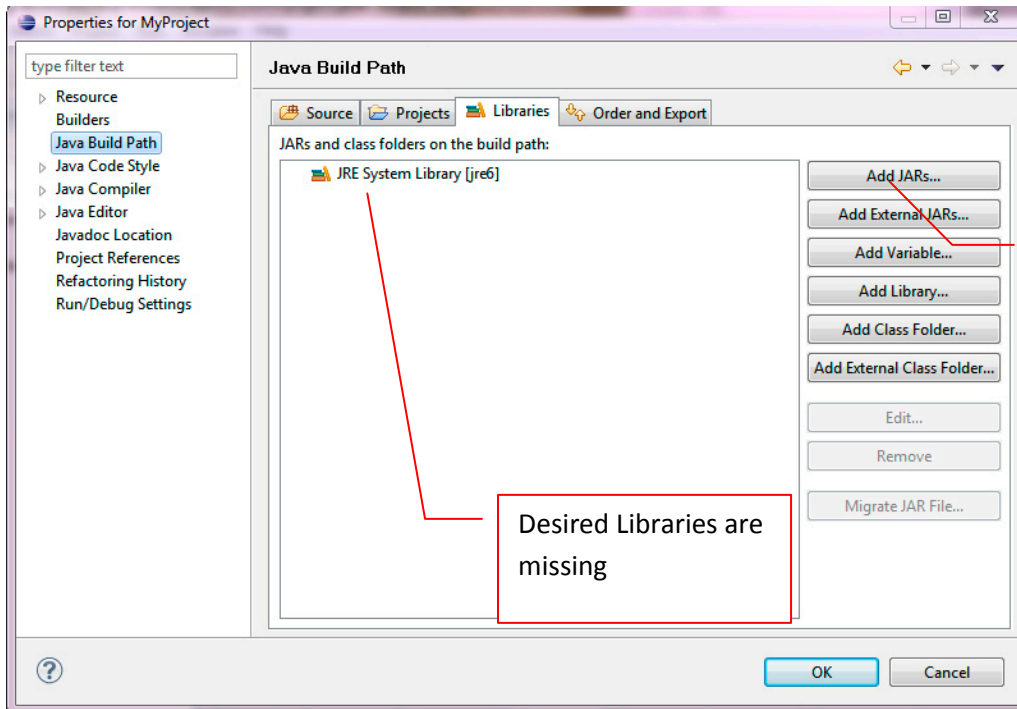




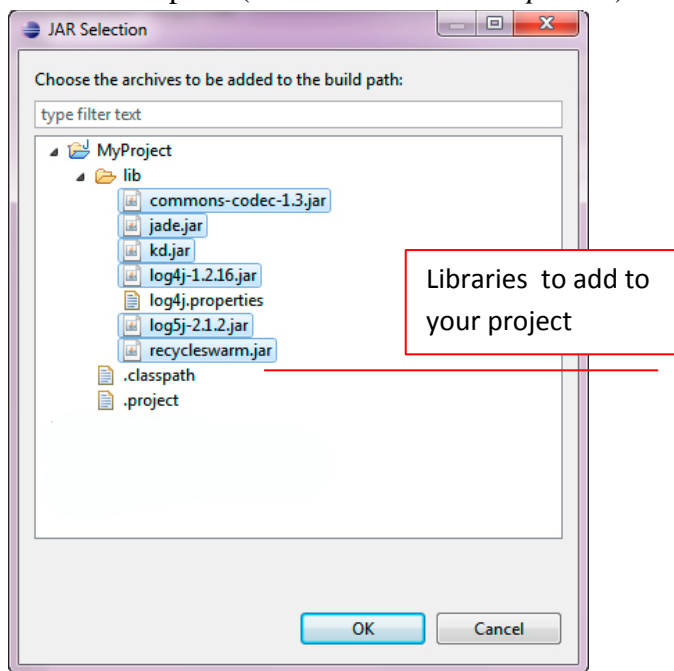
Insert desired name for the project here

Insert the folder created earlier here

You need to have the libraries from the *lib* folder in your project. Check if they are included (select project, right mouse click and select *properties*, then go to the *Java Build Path* tab and look in the list for them). If they **do not appear** in the project, you need to add a reference to them.



In this case, select your project, right mouse click and select *properties*. In the *Java Build Path* tab, under libraries select *Add JARs* and choose the three libraries from the *lib* folder. Hit *OK* twice to complete (*Jar Selection* and *Properties*).



It is recommended to organize your solution into a package having a suggestive name, and following Java best practices (example: *teamname.AgentName*). The agents' classes must extend the *jade.core.Agent* class and override the *setup* method, where the agent is initialized.







Although in the game you will be controlling three types of vehicles, and there will be multiple instances of the same vehicle, all agents in your team will use the same code. There will be exactly one instance of your agent for each vehicle in your team, and the only way to communicate between these instances is by sending and receiving messages.

You are given an example (*DummyAgent* from the *org.aimas.wo.recycleswarm.agents* package) to help you understand.








Test your agent as described before.

## Recycle Swarm GUI

In order to understand the GUI note that: each team has a color (orange for the first team and cyan for the second team), and each of its vehicles are represented by circles according to the following rules:

- *Scouts* are discs colored in the team color (   )
- *Harvesters* are discs colored in the team color and having a thick black outline (   )
- *Carriers* are discs colored in the team color and having a thick black outline and a black concentric circle of a smaller radius (   )

The other elements on the map:

- *Recycling Centers* are red triangles (  )
- *Garbage Piles* are blue squares (  )
- *Garbage Packets* are magenta squares (  )
- The terrain types are colored as follows:
  - Normal terrain – Green 
  - Muddy terrain – Light gray 
  - Rough terrain – Dark gray 
  - Obstacles – Black 
- The route each vehicle is travelling is represented by white lines, with the initial waypoints set by the agent represented by white squares
- The visual range of a certain vehicle is a semi-transparent circle

