

Design Document for Braitenberg Vehicles (Draft)

Angela Almquist
University of Minnesota, Minneapolis, MN 55455
CSCI 3081W Spring 2019

March 6, 2019

Introduction

The purpose of this document is to identify the current and projected changes to be made to a provided Braitenberg Vehicle Simulation. An interface, along with functional modifications will be made to the current code base. A UML, or Unified Modeling Language, diagram will be delivered to show the relationships and between the interface and previously defined concrete classes.

Theory

The Braitenberg vehicle simulation was first introduced by Valentino Braitenberg. The simulation can be very interesting if implemented correctly. In a successful simulation, the behavior of these vehicles are dependent on their surroundings and user inputs. These robots can be set to cower from objects, avoid other vehicles or be particularly glutenous towards food.

A preliminary code has been provided by the University of Minnesota CSCI 3081W teaching staff. This code is sufficient to start a Braitenberg simulation with the help of a hard-coded .json file to define the initial conditions of the simulation. Eventually, the simulation should start with default initial conditions and allow the user to define any changes to the simulation such as number of robots and obstacles, robot behavior and color changes. This document outlines the changes that I plan to make to the code set in order to implement an interactive Braitenberg Vehicle Simulation.

Design Process

The first modification to the preliminary Braitenberg code is to implement an interface called `FactoryEntity`. With this interface, new robots, lights and food can be instantiated in the simulation. Using an interface will make it easy to implement certain features and/or changes to the Braitenberg objects locally within the factory scope without hard-coding changes in the large code-base.

Currently, with the provided .json file configurations, the robots can be set to "Explore" their surroundings. All other behaviors need yet to be implemented. To implement these behaviors, I plan on defining the behaviors in the `FactoryObject` class. The behavior should affect the motion and color of the objects. Each `FactoryObject` class will inherit from their respective classes in the preliminary code. For instance, the `FactoryBraitenbergVehicle` will instantiate a new Braitenberg Vehicle which inherits all of the vehicles default behaviors and dimensions. These behaviors, and how the motion of the robots will be affected, will be defined in the factory class.

Other changes that must be made, are modifications to the colors of robots, their speed, how long these robots linger near food and how frequently they return to food. The design process for implementing these methods are still a bit unclear.

Conclusion

Much work is left to be done. This is just a draft of the design process, and it is sufficient to say I will need some guidance from the 3081W teaching staff.