

Design Document for Braitenberg Vehicles

(Draft)

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

March 9, 2019

Introduction

The purpose of this document is to identify the current and projected changes to be made to a provided Braitenberg Vehicle Simulation. The pros and cons of implementing an interface verses a concrete class to add new robots to the simulation will be considered. The current update to the working code incorporates a braitenberg factory interface.

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. The downside to using an interface is the programming part can be tricky to get correct. The tests are a good indicator at how complex an interface can be. It will be tricky to implement the subtle changes due to user interaction. Alternatively, a concrete class would be easy to add changes within the provided code however, these changes may be substantial. I'm choosing the interface method because I think this is a good opportunity to benefit from the elegance of an interface, and I feel there will be many more hurdles to overcome if one must hard code each change/option to each class as this program is developed.

With the provided .json file configurations, the robots can be set to "Explore" their surroundings. Other behaviors have been implemented by me. The behavior should affect the motion and color of the objects. Color changes due to what the robot is sensing has also been implemented. Eventually, The factory interface will be used to instantiate objects in the simulation based on user input. It will be expected that the user can add robots during the simulation.

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.