

CS 5110/6110 Program 2

(30 points)

We will be experimenting with multiple reactive agents.

Using Program 1 as a starting point, adapt the code to do the following. Set the parameters (shown in quotes) as you see fit.

The Roomba knows its location, and the location of the charger. The Roomba can sense if it is picking up (a) any dirt at all (b) excess dirt. The Roomba can sense another Roomba when it runs into it. The vacuum will have to go “much slower” on dirty areas. The Roomba can detect obstacles and drop-offs.

Develop a method to determine the goodness of your method. Given a specific amount of time, how well did it do? What percent of the accessible cells were reached? If cells had excess dirt, were they cleaned?

Part 1: One reactive Roomba

“Possible Rules” for Roomba are as follows

1. Calculate room size (and take time proportional to room size)
2. While sense drop off (going down steps, for example):
 - a) Bounce back
 - b) rotate,
 - c) move forward.
3. If sense wall, follow close to wall without touching
4. Determine own cleaning path
 - a) “Outward cleaning spiral” or “back and forth”
 - b) When hit obstacle, “clean perimeter or move randomly”
5. If battery gets low, return to charger
6. Clean for estimated needed time
7. If sensing lots of dirt, repeat over same area.

Create a room that contains the following features:

- a. Drop off
- b. Odd shaped edges of room
- c. Obstacles (like furniture)
- d. Moving obstacles (like dogs)
- e. Varying amounts of dirt

Try variants of the reactive rules and draw reasonable conclusions, such as “when there are few obstacles, these rules work well”.

Part 2: Multiple reactive Roombas.

If you have more than one Roomba, does your algorithm work? How can you make it better? If a Roomba stops working, can your algorithm clean well?

Create a short video to create a demo of your project. Submit the video with your project.

Submit the entire project to canvas as one zip file.