

Lab 7: Week 8

Due on 5/27/2020 at 5 PM.

This group lab assignment has three questions. Develop the following RobotC programs.

Read the instructions carefully.

- Provide solutions to written questions in a single PDF file. In the title page of this document, identify group members, their affiliations, and percentage-wise contribution.
- Use separate c files for programming questions with filenames in the form Qnm.c, where n is the question number and m, is the part number.
 - E.g., **Q1b.c** is the solution file for the Q1 part b programming question.
- **Zip** your solution files (including the PDF) in a file named Gn.zip where n is the group number and upload on D2L. Upload only one solution per group.
 - E.g., G3.zip is the solution by group 3.
- If you have any questions, contact the instructor.

Q1: Implement the simple Wall Following to avoid obstacles and reach a desired target and test on custom Robot Virtual World (RVW) maps provided. Read Section 7.1 of the textbook to learn more about the algorithms we are implementing in this lab.

- a) Design the state diagram for the simple Wall following algorithm.
- b) Write the pseudocode for a **Controller Task** that will implement proportional control for turning as well as stopping the robot 20 cm before the obstacle. Also, use the slew rate algorithm to control the initial acceleration of robot motions.
- c) Implement the RobotC program and test on the **ObstacleAvoidance_1.rvi** (Figure 1) and **ObstacleAvoidance_2.rvi** (Figure 2) arenas provided. Report your observations in the accompanying report.

Notes:

- Refer to the Competition 2 instructions to know how to import RVW maps.
- Make sure that you use a reasonable proportional-gain and slew-rate values.
- Use +/- 2 degrees turning error tolerance.
- Use +/- 2 cm distance error tolerance.
- When turning, make point turns to avoid heading or clearance errors.

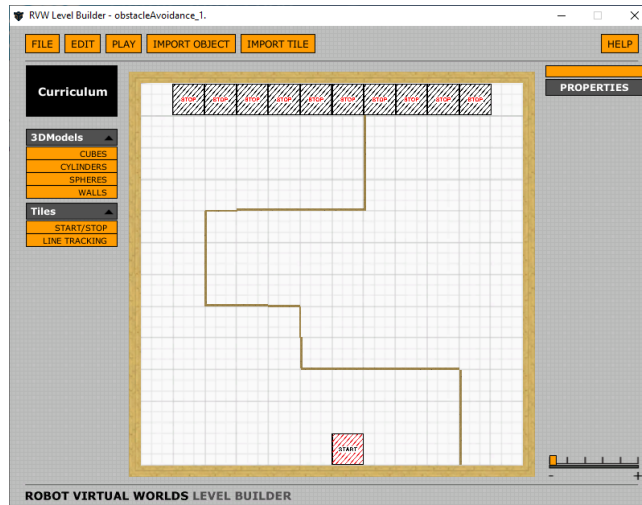


Figure 1: First arena.

Q2: Improve the simple Wall following algorithm developed in Q1 to incorporate the heading direction (the algorithm is termed Wall following algorithm with direction).

- Draw the state diagram (based on Q1-a) to add the direction following the feature.
- Modify the pseudocode in Q1-b to reflect the changes.
- Implement the RobotC program and test on the `ObstacleAvoidance_1.rvi` (Figure 1), `ObstacleAvoidance_2.rvi` (Figure 2), and `ObstacleAvoidance_3.rvi` (Figure 3). arenas provided. Report your observations in the accompanying report.

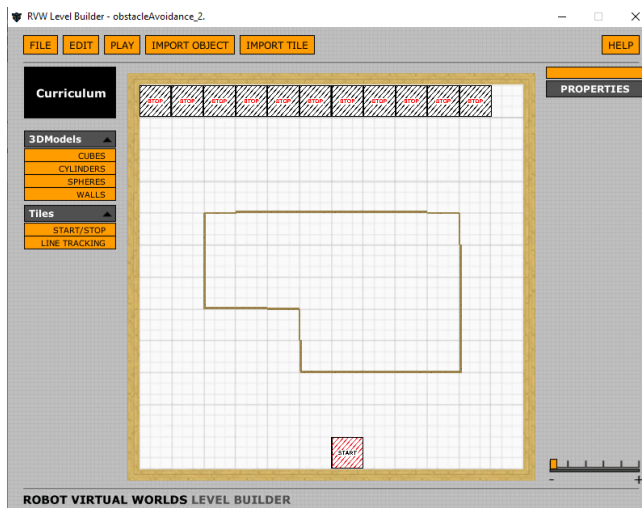


Figure 2: Second Arena.

Q3: Modify the Wall following algorithm with direction developed in Q2 so that it will avoid getting to an endless wall-following loop (the algorithm is termed Pledge algorithm).

- a) Draw a new state diagram (based on Q2 state diagram) to add the pledge feature.
- b) Modify the pseudocode in Q2-b to reflect the changes.
- c) Implement the RobotC program and test on the **ObstacleAvoidance_1.rvi** (Figure 1), **ObstacleAvoidance_2.rvi** (Figure 2), and **ObstacleAvoidance_3.rvi** (Figure 3). arenas provided. Report your observations in the accompanying report.

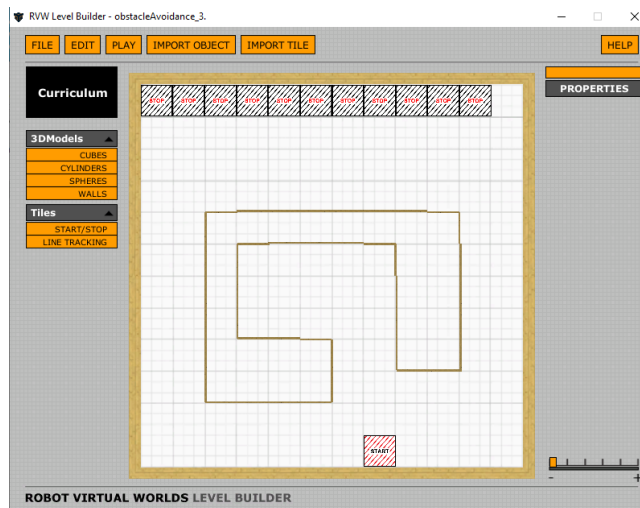


Figure 3: Third Arena.