Due 5/27/20 at 9 PM.

GLE Competition 2

This is a GLE Group Competition. The activity is graded out of 100.

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 file) 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.

Competition 2: Guidelines

- CSC 475 students are graded at a stricter criterion (10%) than the CSC 375 students. That means similar competition performance will earn 90% of the points than a CSC 375 student.
- Each group gets 2 attempts. Only the best performance (designated by you) will be considered for grading.
- You can try different algorithms or make changes to the algorithm between the attempts. However, you only have 5 minutes to complete the demonstration.
- After all the demonstrations, one student from each group will briefly explain your state machine-based implementation (3 minutes).
- Following implementation explanation, one Depaul student and one UNESP student (chosen by the group) will summarize the key takeaways from the nature of collaboration between the Depaul students and UNESP students (2 minutes total).

Track Challenge 2

In this competition, each group will develop a state machine-based solution to solve the custom Track Challenge of the Robot Virtual Worlds Level Builder (see Figure 1). See the section on Importing the Custom challenge section to know how you can import the challenge arena. This challenge has a track bordered by two black lines on a white surface. Along the track, there are movable obstacles. The robot is to move past these obstacles without affecting its course within the track. There is a dead-end (identified by a wall). In order to continue the challenge, you should locate the track that is 4ft from the dead-end wall. Also, there is a wall within 1ft on the opposite side of the next track. Your robot should reach the STOP tile as quickly as possible.

Notes:

- The robot should complete the challenge.
- On 5/27, in the competition, each group will demonstrate their solution during the synchronous meeting.
- The robot should follow the track at all times without veering off track.
- You are free to use any sensor you want.
- You are expected to follow the task-based implementation that interacts with peripheral individually and in parallel. Use appropriate Task rates with justification (on the report).

Exercises:

a) Draw the state diagram for this competition challenge. Use descriptive state designations. Identify what actions you are taking under each state. Identify the sensor values that you used to make the state transitions. Include the state diagram in your report (in the PDF file).

b) Implement the state machine you developed in part a) in RobotC. Adhere to the task-based development philosophy, wherein separate tasks are implemented to interact with each sensor and motors. Decide on the rate at which each task should run (you may rely on the experience you have gained programming the EV3 robot so far.

c) In the report, include a one-page section that discusses GLE collaboration (use 1-inch margin, single-line spaced 12 size Times New Roman font). This section has two parts, the perspective from the DePaul students and UNESP students. The discussion should include the nature of the collaboration (the technology used, the frequency, the contribution from each group, etc), the positives, and the things that could have been done better. Please indicate how the collaboration has been improved since the last challenge. If there are challenges still, indicate how you would address those challenges moving forward.

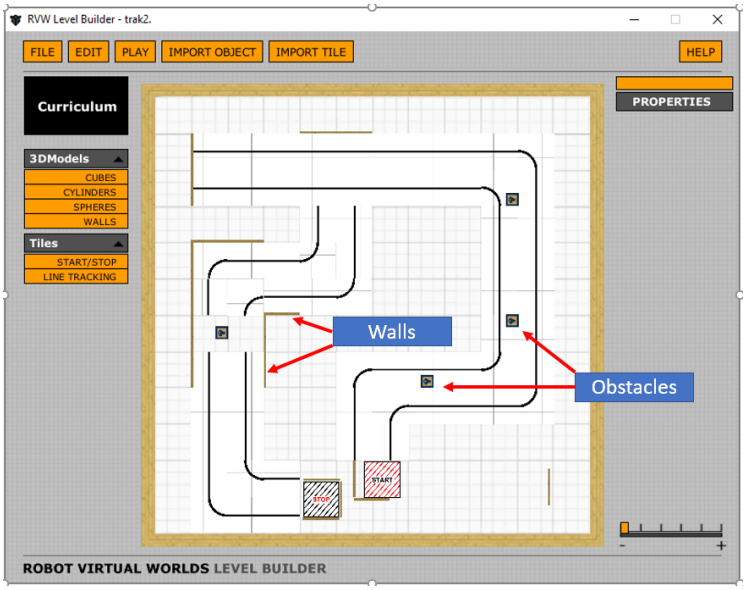

Figure 1: Line-tracking challenge arena.

Grading Criteria

a. Complete the competition (20%)

This will be normalized to how much you complete.

b. Task-based State machine implementation (20%)

i. Independent peripheral interaction (20%)
ii. Appropriate rates of sensor reading and actuation with justification (in the report). (20%)
iii. Using variables to communicate among Tasks without cross calls. (20%)
iv. Correct and precise state-machine based implementation (40%)

c. Following the track (20%)

Time remains outside track will cost you points. The time outside will be calculated as a percentage of the total time taken to complete the task (i.e., 100 * time outside / total time)

i. Less than 5% (0% deduction)
ii. 5%-15% (10% deduction)
iii. 25-35% (25% deduction)
iv. 35-60% (50% deduction)
v. More than 60% (100% deduction)

d. Time to complete the entire trajectory (10%)

i. Completion time will be recorded and normalized.
ii. Points are awarded at a linear scale maximum time (100%) to minimum time (120%). For instance, the slowest group will receive no penalty, whereas the fastest group will receive 20% extra credit.

e. Report (30%)

i. Detailed state diagram with clearly identified states the actions taken in each of the States sensor information used to determine the state transition. (50%)
ii. A section on the of GLE collaboration. (50%)

Importing the Custom Arena

1. Install the RVW Level builder pakcage. Goto Help > Manage RVW packages and select RVW Level builder (Figure 2).

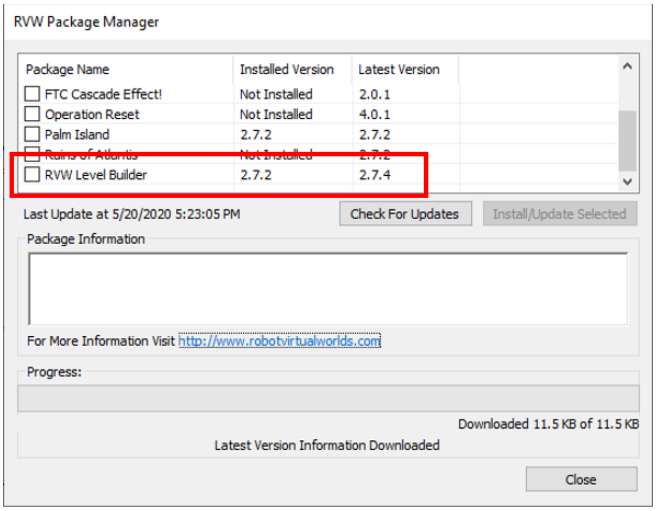

Figure 2: Download RVW level builder

2. Change the Virtual world to use from Challenge pack for EV3 to RVW Level Builder (Figure 3).

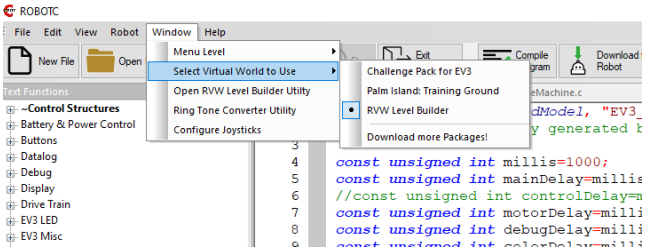

Figure 3: Select RVW level builder

3. When you compile and download your program, it will prompt the RVW level builder interface shown in Figure 4.

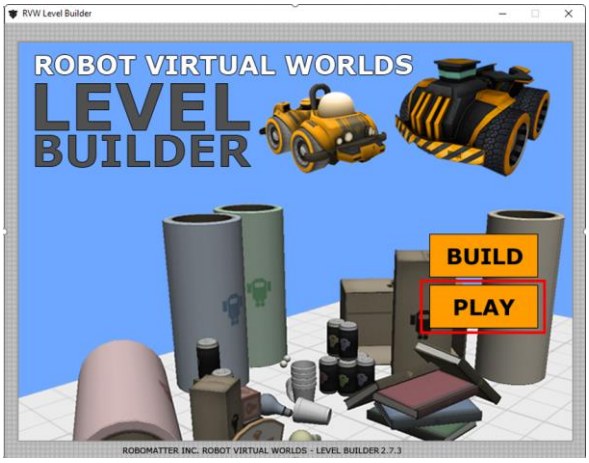

Figure 4: Select Play

4. To import the custom arean, select ther drop-down menu shown in Figure 5 and click Browse.

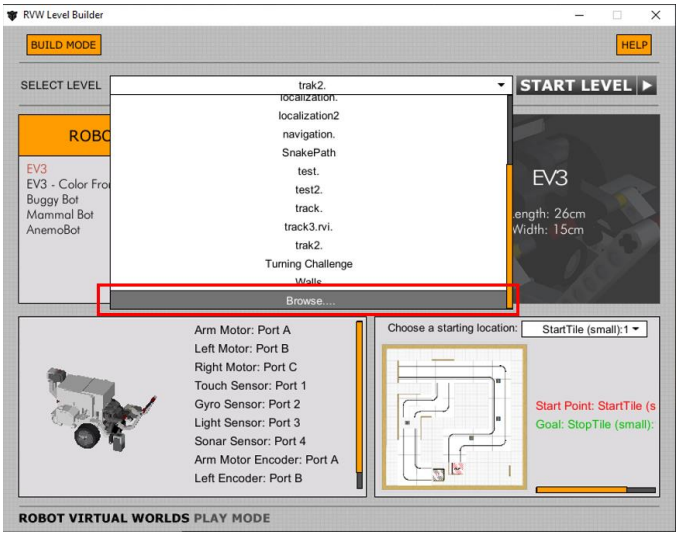

Figure 5: Select browse

5. Navigate to where you download the custom arena and click open (Figure 6).
6. To run the program on the custom arean, click start level (Figure 7).
7. You will then be taken to the simulation framework (Figure 8) similar to RVW EV3 challenge pack.

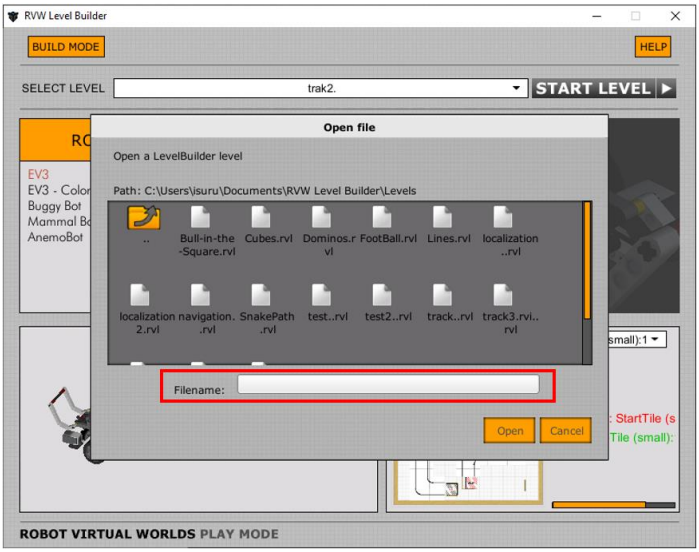

Figure 6: Select the file

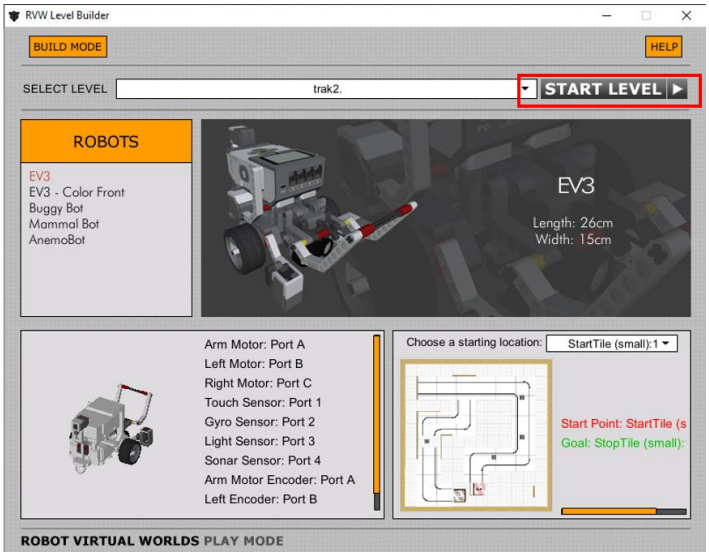

Figure 7: Select the start level

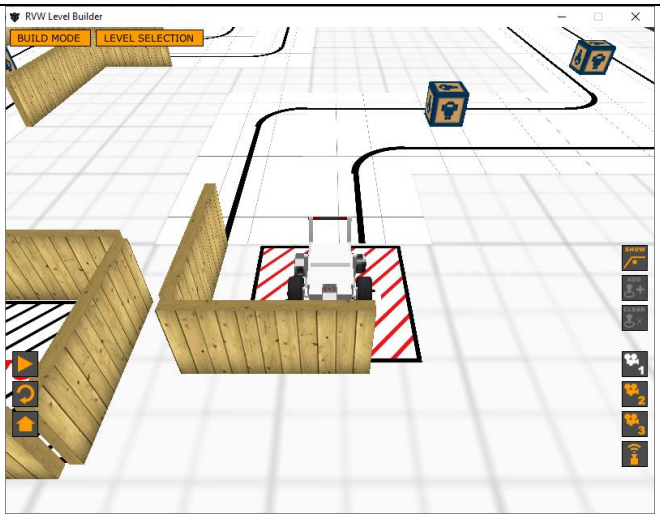

Figure 8: Robot is ready to run in the custom arena.