Institute of Technology, University of Washington Tacoma
TCSS 437 Mobile Robotics, Spring 2017
Robot Challenge 1
Submissions due: 13 April
Value: 15% of the course grade

Purpose:

This project has a number of purposes:

- Construct a robot using the Lego education EV3 kit
- Program your robot to display multiple layered behaviors
- Work in a team to creatively design and implement a project

Project Description:

Design, construct, program, test, and demonstrate a mobile robot with the following behaviors:

- **Wandering**:
  This behavior emulates the non-deterministic way in which various animals explore an unknown environment. Animals do not move in a straight line until they sense something that they want to obtain. Rather they spend a great deal of time zig-zagging in a semi-random pattern called a biased random walk (or drunken sailor walk) as opposed to a pure random walk. This pattern has a generally forward motion with swings to the right or left that are at random intervals and for random amounts of time. Your robot, when not encountering an obstacle, should demonstrate a biased random walk as shown in the diagram at the end of this document. For full credit the biased random walk should consist of smooth curves, not connected line segments.

- **Object Detection and Reaction**:
  Bumpers are used to detect obstacles in front of the robot. The required reaction is relatively simple. If an object is detected on the right side (only the right bumper is depressed) then the robot should back up a short distance, turn to the left and resume wandering. Similarly, if an object is detected on the left, it should perform a mirror image response. If both bumpers are depressed (maybe within 10 milliseconds of one another?), the robot should make a sound, back up a short distance, pause for 2 seconds without moving, then turn in a random direction and resume wandering.

Getting Started:

- Construct a robot with touch sensor bumpers mounted front/left and front/right. You may use or modify an existing design or you may create your own design.
- Develop a program to provide the robotic behaviors described above. RobotC has many example programs from which you can get ideas and inspiration.

Performance Requirements:
Your robot will be tested in 2 ways. First, we will set up an arena (a test table). There will be several objects placed in this arena. The robot should be able to explore this arena for 4 minutes without getting stuck, flipping over, or having any pieces fall off. During the 4 minute test, the robot should ideally explore all parts of the test arena. Second, when traversing an open area of the lab floor, your robot should use the biased random walk pattern (NOT a purely random walk and NOT a walk with ANY sort of predictable repeated pattern. Be sure you understand what is meant by a biased random walk.) Your robot should use the same program for both parts of this test. You will not be allowed to stop, restart, or change your code when moving your robot between the two parts of this performance evaluation.

Lab Journal:
Maintain a lab journal as you work on this project – a basic format is provided on Canvas.

Record all design and implementation options that you consider (even ideas that don’t work and are later abandoned). Give credit to all sources that you use for robot construction and programming ideas.

Test each behavior independently and record your findings, then include both behaviors in a program and test that. Remember, this is experimental. At each stage of project development generate hypothesis about what results you expect your current hardware or software change will produce, set up an experiment to test your hypothesis, record and analyze the results. Repeat this pattern as many times as necessary to refine your hardware and software designs. Every detail of your process should be recorded in your team's lab journal. Try many configurations of both hardware and software.

Your journal can include things other than text. You may include diagrams, spreadsheets with data, images, video, etc. Include whatever helped you during the development of this project and anything that will help communicate your efforts and ideas to me. I specifically appreciate at least one image of your robot.

Coding Technique:
Apply all that you have learned previously to the development of high quality code. (Don’t abandon good practices now that you are working on a robot!)

For example, do not put your entire program into one huge function or task. Decompose the problem in a reasonable fashion and implement the code accordingly.

Use meaningful names in your code and provide adequate commenting in your code to orient a programmer who is looking at your code for the first time (like me when I grade your code). Use comments in your code to clarify any parts of your code that are complex or could easily be misinterpreted.

Submission and Grading:

When you are ready to demonstrate your robot, notify me and we will schedule a test (this takes about10 minutes). You will get three tries, if needed, to meet the performance requirements.
After a successful demonstration, use the link on Canvas to submit your lab journal, your code, and any other files you wish me to consider in single .zip file. (Just one submission for each team).

Each team member must also separately submit an individual report. Please see the detailed instructions for this on Canvas.

Grading will be based on the following:

- Successful demonstration of the wander behavior. 25%
- Successful demonstration of the object detection and reaction behavior. 25%
- Completeness and quality of the lab journal. 30%
- Coding technique. 5%
- Individual team member reports 15%

Hints and Suggestions:

An enormous amount of information exists on the internet that can help you with this project.

There are many complete robot construction plans published online for the EV3. RobotC contains numerous example programs but you could look online for other robot programs in RobotC or in other languages and then translate those to RobotC (give credit to original sources in your code and in your journal).

Read about random walks online to supplement what the lecture discussions provide. You may find algorithms, both simple and complex, described online. (Give credit for the source in your code and in your journal.)

The EV3 hardware supports concurrency and you may need to program multiple threads for this robot challenge (this is highly recommended since the other challenges will require concurrency). Look over notes that you have from previous courses about concurrency. Look for RobotC example programs which implement concurrent tasks.

Extra Credit:

(Don’t spend time/effort on this until you have a solid start and high confidence of success on the basic project requirements!)

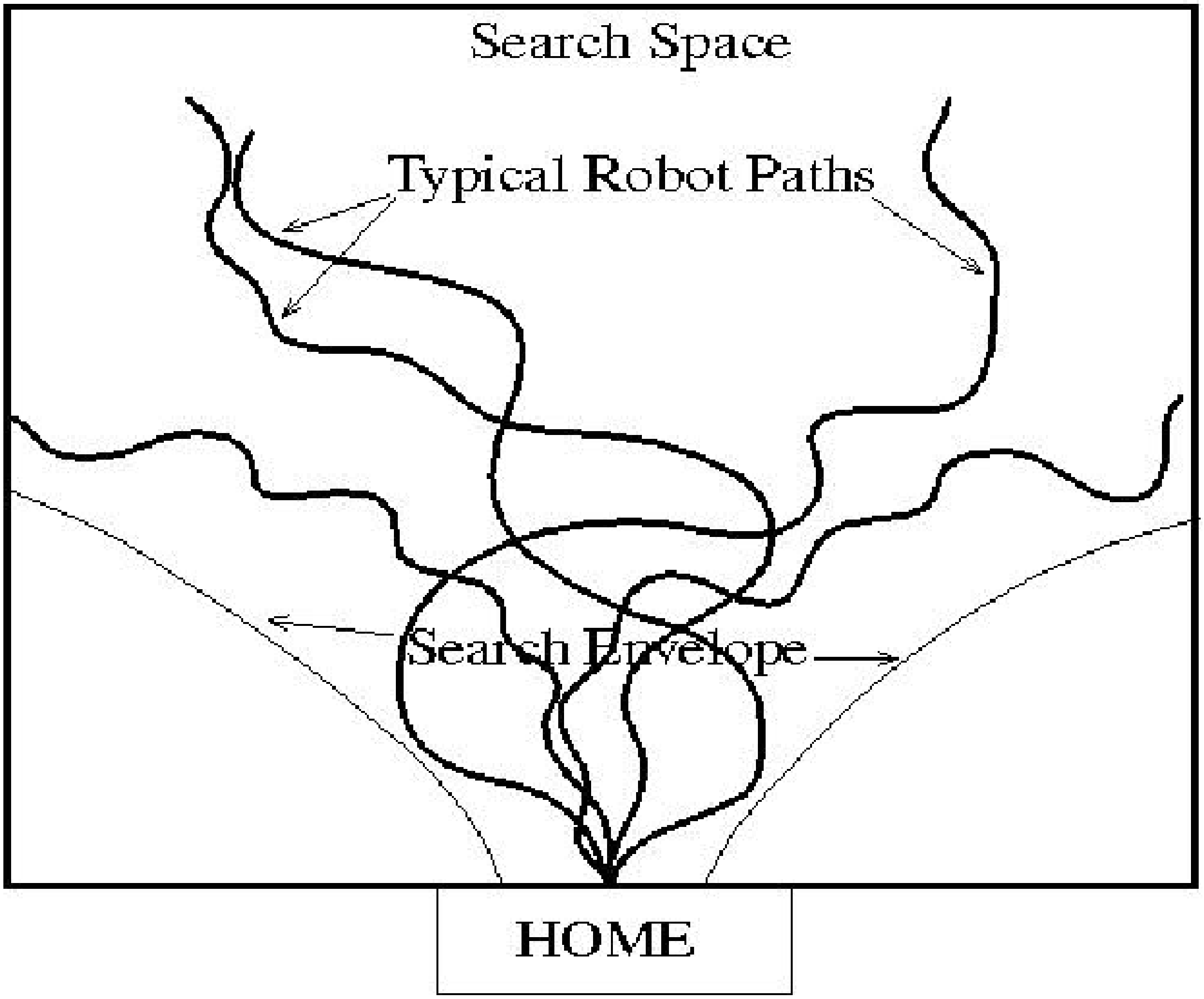
I will award extra credit to teams who submit any of the following:

- An additional implementation in a language other than RobotC – 5%

(If you want to work on this ask for an extra intelligent brick to work with.)

- A CAD file of your robot design with building instructions - 5%

Examples of paths taken during a biased random walk:


Search Space
Typical Robot Paths
Search Envelope
HOME