In this lab you'll write more pseudocode to control the robot. Today's challenges are more open-ended than last time's. There are many ways to solve them, so try to start brainstorming early. **Remember to bring your Scribbler and its cable to the lab!**

Hand in your lab report at the beginning of lecture on Tuesday, March 2nd. Include the programs you wrote (use "File -> Save As…" in the Scribbler Control Panel to save a program, and "File -> Print" to print it). Explain what your program from Part 4 does (if you chose to do it). Briefly analyze how well each program worked, and write about any problems you needed to overcome.

Part 1: Navigating a Maze

Teach the robot to escape from a simple maze. We will test your solution by placing Scribbler at a random location within a designated starting zone. This means the robot must use its sensors to navigate. Demonstrate your program for your TA. We'll give full credit for one successful escape.

Obstacles block the robot's path.

A heavy black line will warn the robot as it nears the lava.

Scribbler will start facing north at a random position in this region.

To the east of the maze is a pool of lava. The Scribbler will be destroyed if it strays into this area.

The robot's goal is to reach the maze exit, represented by this square.
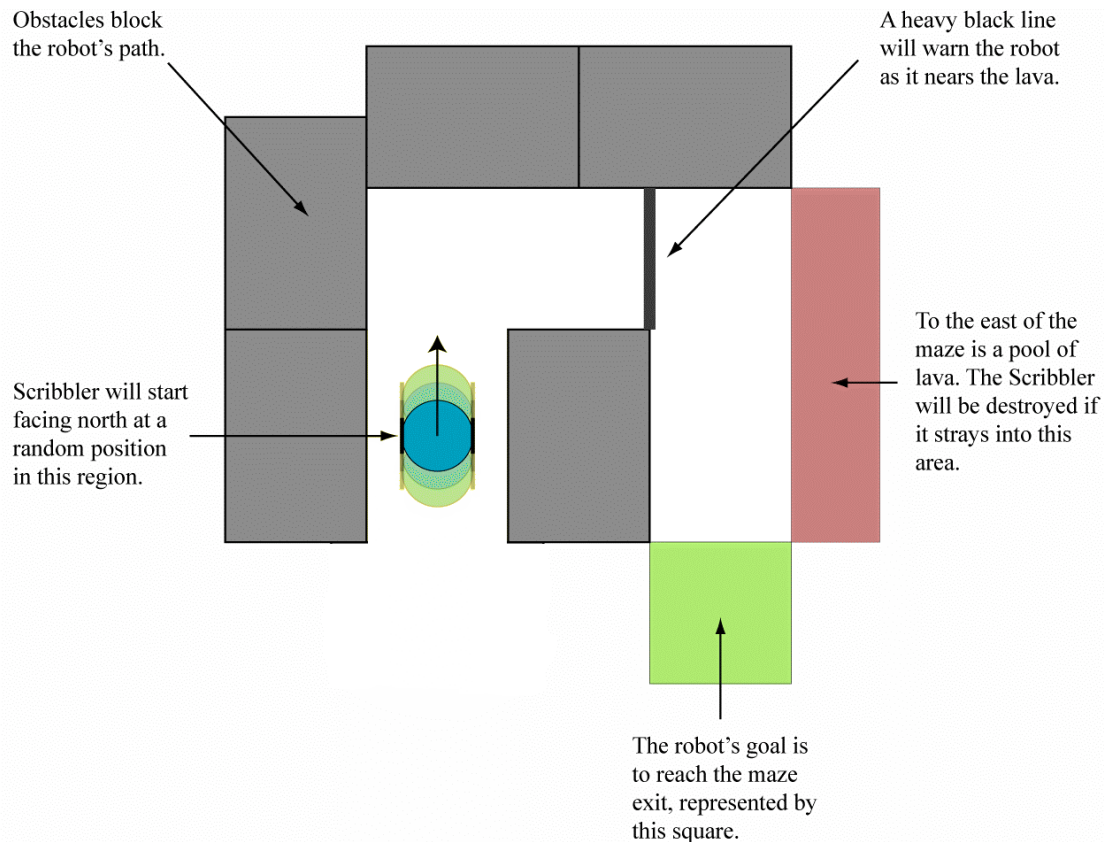
Part 2: Lights and Sound

A rapidly blinking light source appears to the eye to be completely steady. This phenomenon is called *flicker fusion*. Psychophysicists define the *flicker fusion threshold* as the frequency at which a person detects flicker in 50% of trials. The threshold varies among individuals and is affected by the brightness of the light, the location on the eye, and other factors. You can use the Scribbler to observe the flicker fusion effect and measure the flicker fusion threshold.

Experiment 5 — Measure your flicker fusion threshold

Instructions required: LED, Pause, Do Forever

Write a Scribbler program that repeatedly blinks the LEDs on and off. Insert equal length pauses after the LEDs turn on and after they turn off. Start with very brief pauses—0.001 seconds.

1. **Run your program and watch the LEDs. Do they appear to flicker? If not, alter the program to double the pause times, and try again. Repeat this process until you can see the LEDs flicker.**

2. **What is the briefest pause time where you can see flicker? What frequency does this correspond to?** (*Frequency* = 1 / *time*. Here, *time* is the length of one complete on-off cycle.)

Experiment 6 — A Scribbler siren

Instructions required: Do forever, Sound

Program the Scribbler to sound a siren by repeating two different tones. Experiment with different frequencies and durations to produce a realistic effect. Try frequencies that are close together and ones that are farther apart. Characteristics of the robot's speaker and the resonance of its plastic case cause some frequencies to sound louder than others. Choose loud ones for your siren.

Part 3: Non-determinism and Randomness

The robot's sensors detect the condition of the physical world, but there is another condition that can be used in branches and loops: the outcome of a random "coin toss." Of course, the robot doesn't toss an actual coin; rather, it uses a mathematical function with random-looking outputs to simulate a coin toss. A future lecture will revisit randomness in computation. Here we do a few experiments.

Experiment 7 — Testing the coin toss

Instructions required: If, Forward, Back, Pause

Program the Scribbler to toss a coin. If the coin toss comes up heads, move the robot forward a few inches; otherwise, move it back a few inches.

1. **Run your program, but first mark the Scribbler's starting position with a piece of tape. After the program completes, measure the distance *X* that the Scribbler travels.**

Experiment 8 — Random walk

Instructions required: Do for *n* times, If, Forward, Back, Pause

Write a program that adds a loop around the instructions in Experiment 7 so that the Scribbler tosses a coin 25 times.

1. **Run your program, but first mark the Scribbler's starting position with a piece of tape. After the program completes, measure the distance *Y* from the starting position to the ending position.**

2. **Using *X* (from Experiment 7) and *Y*, estimate about how many flips came up heads and how many came up tails.**

3. **You may find that the Scribbler does not stay on a straight line as it moves back and forth. How do you think this affected your estimate in Step 2? Specifically, do you think it caused you to overestimate or underestimate the disparity between the number of heads and the number of tails? How?**

Extra Credit: (Requires some familiarity with probabilistic thinking and infinite series.)

The coin toss is a way to make a uniformly random selection between two choices. But what if we want to choose uniformly among three choices?

Experiment 9 — Choosing uniformly among three choices

Instructions required: Do forever, If, LED, Pause

Write a program that randomly picks one of the Scribbler's three LEDs and turns it on for 1s, and keeps doing this forever. Make sure that at each step, each LED has probability exactly 1/3 of being chosen. (Hint: As a first step,

you could come up with a method so that the probabilities are very, very close to 1/3, rather than exactly 1/3.)

Extra Credit: Scribbler Art

Now it's time to get creative. Working alone or in a team, write pseudocode to show off the Scribbler's artistic side. Here are a few suggestions for inspiration, but the possibilities are nearly endless. If your art isn't something that you can submit with your lab report (see examples below), make sure you demonstrate your program for your TA.

- *Music:* Program a song with the sound commands. This web site lists the frequencies for musical notes: http://www.phy.mtu.edu/~suits/notefreqs.html. You might have several robots play different parts to achieve harmony.

- *Dance:* Use motion commands and loops to teach Scribbler a simple step. If you're working in a team, have another robot serve as a dance partner or provide a beat.

- *Drawing:* Give Scribbler a pen and let it run wild. Give the robot a fixed drawing path with motion commands, or combine motion with coin tosses for something more abstract. If you want, bring several different color pens and other kinds of paper.

We'll give 50% of the points for entering anything that works. The remaining 50% will be awarded for creativity (but we'll be generous). The best entry or entries will be shown in lecture.