Institute of Technology, University of Washington Tacoma
TCSS 437 Mobile Robotics
Robot Challenge 3
Value: 30% of the course grade

For this final challenge teams will be of size four. Challenge 3 teams will be formed as teams finish challenge 2. I will try to accommodate as many stated preferences as possible.

Your robot will use (at least) 5 sensors. Therefore you will need two EV3 bricks working together collaboratively (communicating) to provide the required behaviors. Each team will have two complete robot kits to use. Alternatively teams may choose to use the (somewhat older) NXT kits (RobotC also works with the older kits).

Behavior Requirements:

Your robot will explore the lab by wandering as in previous challenges. It will look for white patches/lines which will represent "food". It will react to objects that it bumps against as was done in Challenge 1. It will detect and approach objects as was done in Challenge 2. It will also attempt to escape from bright flashes of light which it will interpret as a possible threat.

- **Wandering**: As in the previous challenges.
- **Object Detection and Approach**: As in Challenge 2. The robot will approach any nearby object it detects with its sonar sensor.
- **Avoid obstacles**: As in Challenge 1, the robot will react to objects that it detects with touch sensors.
- **Gradient following**: The robot will perform line following but in a different manner than was done in challenge 2. All lines will have a skinny end that gets broader over some reasonable distance. The thickening end of the line will lead to a large white patch representing food. The robot must determine the direction it should follow to get to the food patch by detecting the gradient (increasing width of the line) and following it to find the patch. If the robot starts to backtrack along the gradient path (out of the patch) it must detect this and turn back to the patch. It should not 'feed' while determining the direction of the gradient. Feeding starts when the robot is convinced that is not heading in the wrong direction on a gradient.
- **Patch feeding**: Once a robot is in a patch it should not leave unless it detects an object, is frightened by a flash of light, or has fed until it is "full". If it feeds until it is "full", it must return to wandering. While the robot is feeding, a variable called "energy_level" (or something similar) should be incremented. If the robot feeds for a full two minutes this variable should be at its maximum "full" value. After the robot leaves the patch, for whatever reason, this variable should decrement slowly. Low values for this variable are interpreted as "hunger". When the energy variable is below a certain point (no more than two minutes after feeding) the robot should be ready to feed again "hungry". Feeding involves continuously moving in a non-repetitive way (wandering) on the patch (your robot should *always* be moving except when it pauses briefly at an object which it has approached using sonar).
- **Escape:** The robot dislikes bright flashes of light and considers them a threat. The robot will react to light flashes as detailed later in this assignment description.
- **Death:** If a robot's "energy_level" variable ever reaches 0, the robot is "dead" and should cease all activity after emitting a final sound.

Clarifications and details about the energy level:

Consider your robot’s appetite for food ("energy_level”) to be on a scale such as the following:

```
dead              danger            hungry                                  full
  |-----------------|-----------------|----------------------------------------|
     must feed     non-deterministic          not interested in food
```

It should take the robot approximately 2 minutes of continuous feeding to go from nearly “dead” to “full”. NOTE: It may take less than 2 minutes for a robot to fully restore its energy level (maybe the robot’s energy level was just a small amount below ‘hungry’ when it found a food patch). You should consider the 2 minute time frame as a 'worst case' scenario for a nearly dead robot (a robot with energy level close to zero).

It should take approximately 2 minutes of no feeding to go from “full” to “hungry”. It should take approximately one minute of no feeding to go from “hungry” to “danger”. It should take approximately one minute to go from “danger” to “dead”.

If the energy level is between hungry and full, the robot wanders without interest in food (ignoring food patches). It reacts to any of the other stimuli as described. As it wanders, the energy level is steadily decremented.

When the energy level is between “hungry” and “danger”, if the robot finds a patch (or is on a patch) it should feed and increment the variable until it reaches full (unless interrupted by another stimulus).

If the energy level falls below “danger” it should ignore other stimuli except the bumpers and look for food.

Have the robot emit a sound whenever it starts feeding and some other sound whenever it stops feeding and returns to wandering. If the energy variable ever reaches 0, the robot is “dead” and should cease all activity after emitting a final sound.

Clarifications and details about the Escape behavior:

The robot must have a light sensor pointed upward that looks for flashes of light. Under normal circumstances, if the robot detects a flash, it will back up, turn in a random direction, and run away for approximately 4 seconds as fast as it can. However, if the robot is "frightened" again within 1 minute of a previous light flash, it should execute this same behavior, but at a somewhat slower speed and for less time. If a series of flashes occurs (each within one minute of a previous flash), the robot should gradually habituate and eventually ignore the flashes (other behaviors such as gradient following and object detection and approach now have higher priority). 1 minute after the last light flash seen by a robot, the robot will begin to “recover” from its fright and gradually return to its initial response level (the level of response it had prior to habituation). If there is a 4 minute delay after the last flash, the robot should completely restore its initial response behavior even if the robot had previously become completely habituated to light flashes. Consider storing a variable (perhaps “habituation_level” and incrementing and decrementing that value in a manner similar to how you are using the energy_level variable”). NOTE: It may take less than 4 minutes for a robot to fully recover if it was not fully habituated by light flashes (maybe the robot only saw one or two flashes). You should consider the 4 minute time frame as a 'worst case' scenario for a fully habituated robot (a robot that saw many flashes).

The robot should be able to mediate between all of these behaviors smoothly. If your robot starts to wander away from the demonstration area, you may pick it up and turn it around.

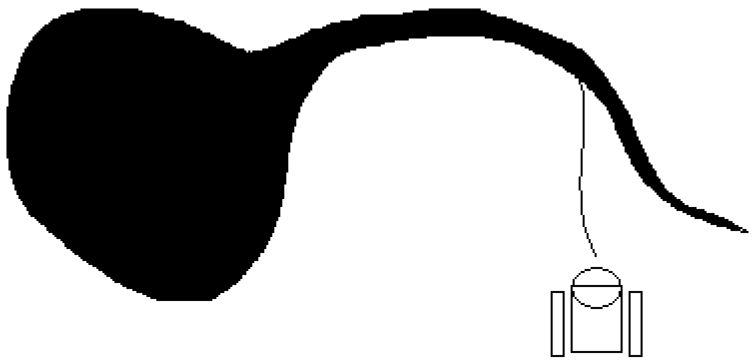

Fig. 1. Robot approaching a gradient path. The robot should be able to detect which direction leads to the patch of food. (White and black reversed in this diagram)

Lab journal

The lab journal should start with an architectural design description with the following:

1. The overall behavior composition architecture - provide a diagram of your prioritized behaviors including information about which sensor values and internal variable values trigger various behaviors and provide finite state diagram(s) for each behavior which show transitions and the trigger causes for the transitions. Optionally also create a ‘complete’ finite state diagram in which all behaviors are represented in a single diagram.
2. A description of the distribution of tasks between the two EV3 bricks and a description of the communication between the two bricks.

Keep your journal as in prior challenges. However, in this challenge you must be very explicit in recording the agreed distribution of work among the various team members. Record the names and contributions of who did what tasks. If all participated in a particular task then indicate so.

In addition, each team member will produce an individual report as in previous challenges.

Turn-in and Demonstration: When you are ready to demonstrate your robot, notify me and we will schedule a run. Turn in your journal and code as a team submission on Canvas as in the previous challenges. Turn in your individual report on Canvas.

Grading will be based on the following:

- Successful demonstration of the reaction to bumper contact. (as was done in challenge 1) 3%
- Successful demonstration of the wander behavior. (as was done in challenges 1 and 2) 3%
- Successful demonstration of object detection and approach. (as was done in challenge 2) 4%
- Successful demonstration of gradient path following. 12%
- Successful demonstration of patch feeding. 8%
- Successful demonstration of escape behavior (from light flashes). 10%
- Successful demonstration of correct mediation among behaviors. 15%
- Completeness and quality of the lab journal. 20%
- Coding technique. 5%
- Individual team member reports 10%
- Disassembly of robots and inventory of kits - checked by instructor. 10%