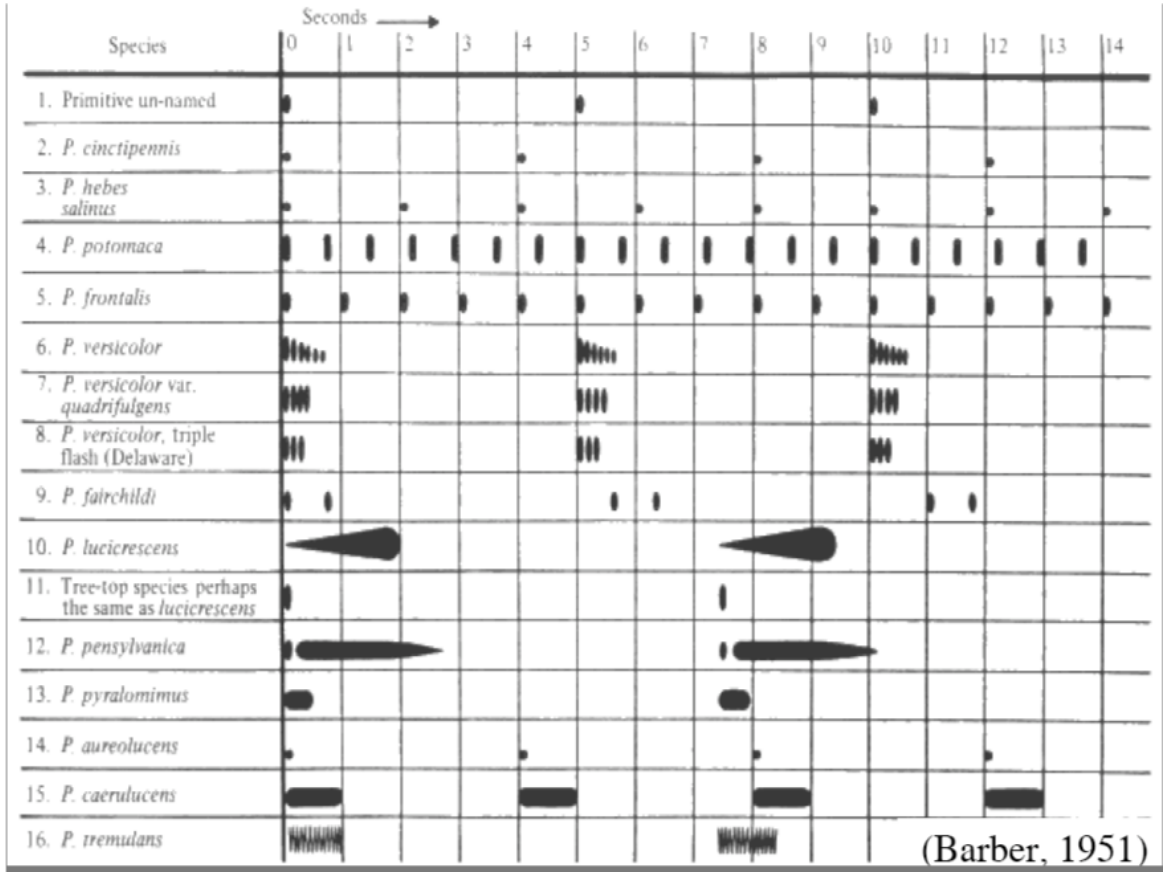
SWARM ROBOTICS: PART 3

Dr. Andrew Vardy
COMP 4766 / 6912
Department of Computer Science
Memorial University of Newfoundland
St. John's, Canada

SYNCHRONIZATION

FIREFLY SYNCHRONIZATION

- Fireflies exhibit a remarkable behaviour where they flash in synchrony, apparently for reproduction
- Since different species have different characteristic patterns, when the males flash appropriately the females can tell they have found an appropriate mate



A SIMPLIFIED VIEW

- The patterns of flashing in fireflies can be complex
- We take the simplified view of fireflies flashing in a single pulse
- Exemplified by this video from Thailand:

WHY IS THIS OF INTEREST TO ROBOTICS?

- Having a swarm of robots flash in synchrony has been proposed as a way to detect failures in other robots:
  - That one's not flashing right; It must be broken!
  - Christensen, A. L., OGrady, R., & Dorigo, M. (2009). From fireflies to fault-tolerant swarms of robots. *IEEE Transactions on Evolutionary Computation*, *13*(4), 754-766.
- In general, we're interested in how distributed agents can agree globally when each of them only has a very limited view of the world
  - e.g. Agree on a set of locations for coloured pucks

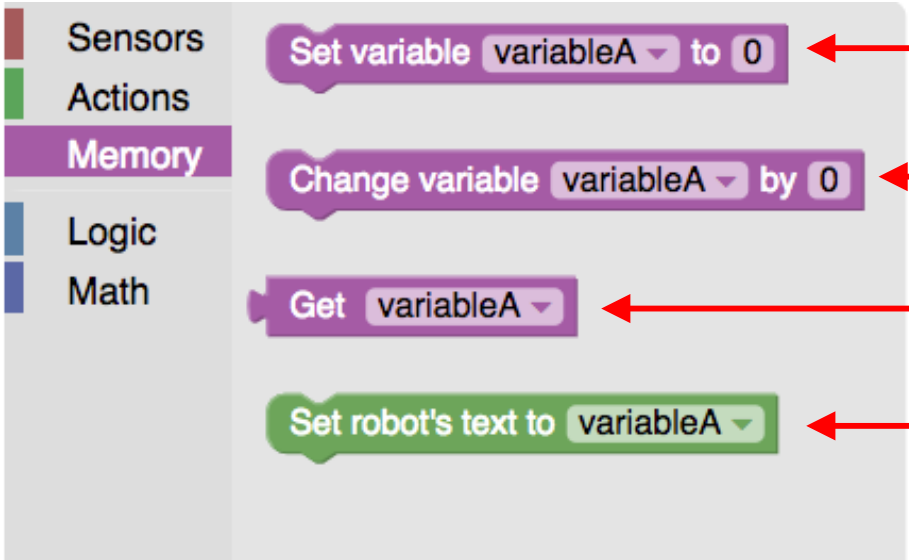
FLASHING IN WAGGLE

- Go to the main Waggle page (or just hit your browser's back button):
  - http://bots.cs.mun.ca/waggle/
- Select the **Fireflies** level. There are no pucks and the robot has lost its puck sensor. But it has a large sensor that can detect flashes from other robots.
- In "Sensors": Number of flashes
  This gives the number of robots within the sensor's area that were flashing during the last time step
- In "Actions": Activate flash Deactivate flash
  These illuminate and de-illuminate the robot

VARIABLES

- All of our controllers so far have been completely **reactive**---behaviour depends only on the present input
- But to make regular pulses of light the robots need to store something about the time since the last flash
- Variables are a fundamental element in programming, but their use in Waggle is pretty limited:
  - Only three variables available: variableA, variableB, and variableC
    - For our purposes we will only need one
  - They are integers, initialized to start at 0

"MEMORY" CATEGORY



Set the variable to the given integer

Modify the variable by adding the given integer

Get the variable's value as a number

Display the variable's value on top of the robot (useful for debugging)

EXERCISE #8:

SIMPLE FLASHING

5 MINUTES

Conditions:

Any number of robots (not trying to synchronize yet)

- Create a controller like this:
  - Add 1 to variableA
  - If variableA = 10
    - Activate flash
    - Set variableA to 0
  - Else
    - Deactivate flash
  - Execute

- See what happens when you increase the number of robots
- Play with "Timescale" slider to see things in slow motion
- Incorporate 

- Note that robots are "born" at different times, and therefore will not be synchronized by default (although those with the same "birthday" will automatically synchronize)

EXERCISE #9 / A6, TASK 4:

SYNCHRONIZATION

Conditions:

Any number of robots

- Start with the "fixed flasher" just created
- If a neighbouring robot flashes, how should you respond so that you flash at the same time during the next cycle?
  - **Devise your own strategy based on this idea**

- Drag your robots around so that they form groups
- You may wish to deselect "Show Sensors"
- You may not get good performance unless the robots "mingle"
  - Incorporate obstacle avoidance and keep the robots moving while they attempt to sync

PHEROMONES

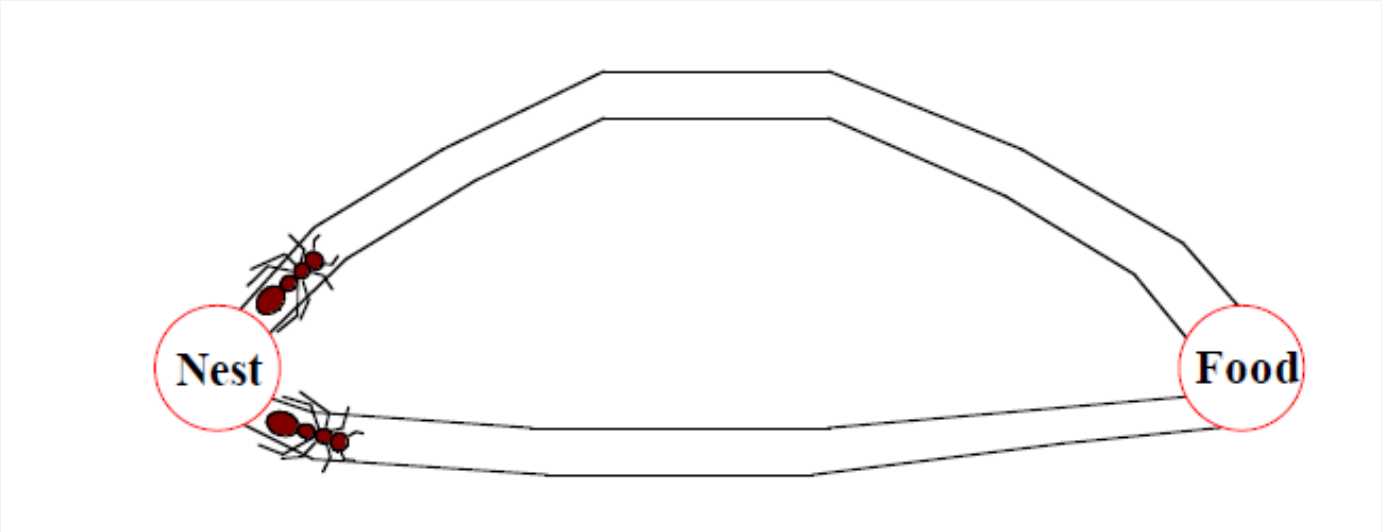
PHEROMONES: INSECT COMMUNICATION

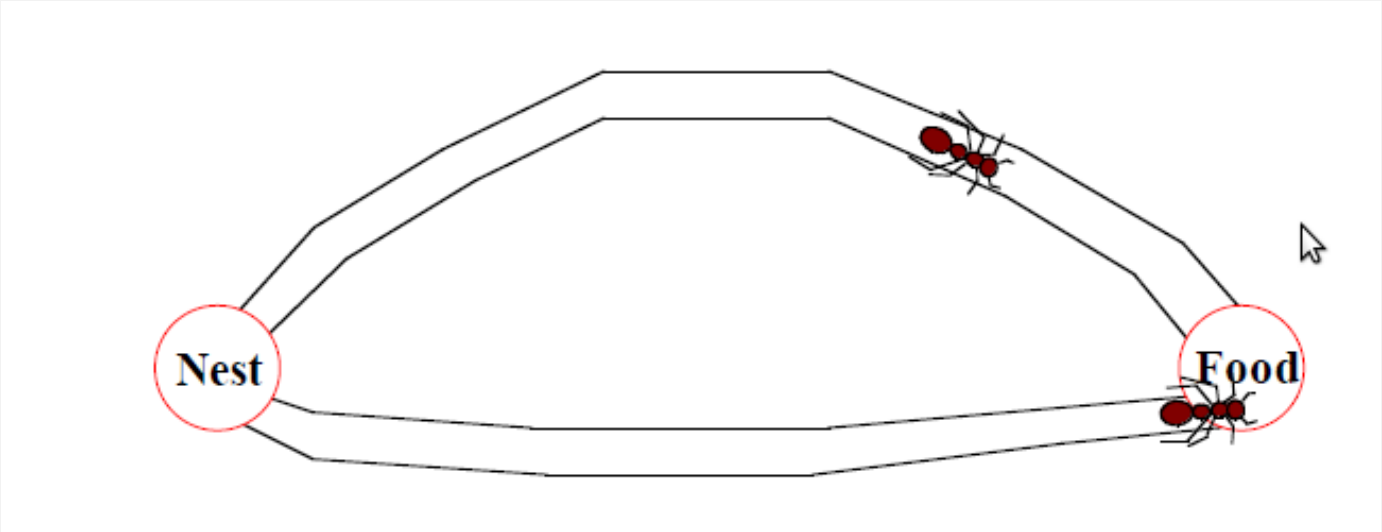
- In many cases, insects deposit chemical signals called pheromones which can indicate something about a certain place or thing
- The classic example of pheromone use is in ant foraging

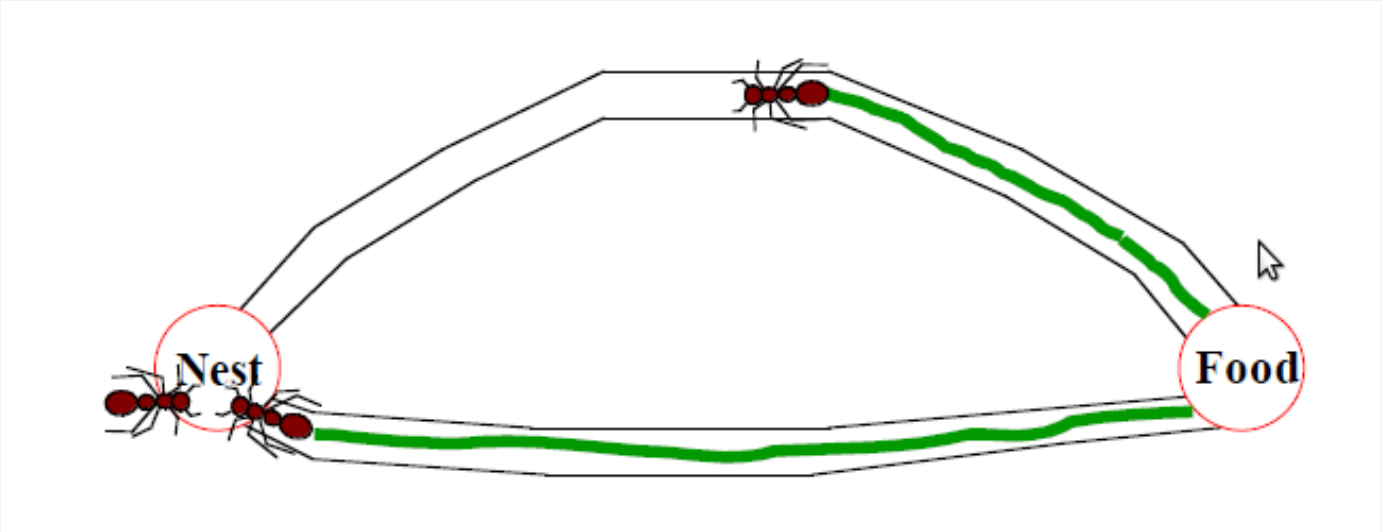
FINDING THE SHORTEST PATH

- When foraging for food, some ants leave behind pheromone trails on the ground:
  - The pheromone is deposited by ants on their way back to the nest
  - After the food source is exhausted the pheromone trail will dissipate
  - Generally ants find the shortest path between food sources and the nest

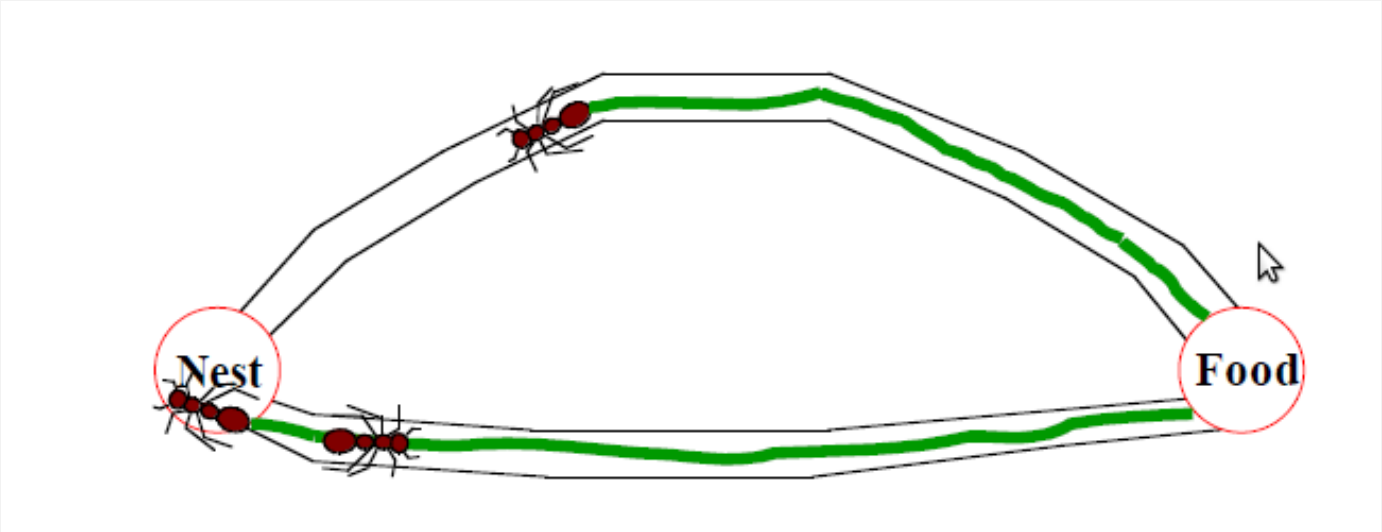
  - That's a big deal! How can these tiny insects solve the problem of finding the shortest path?

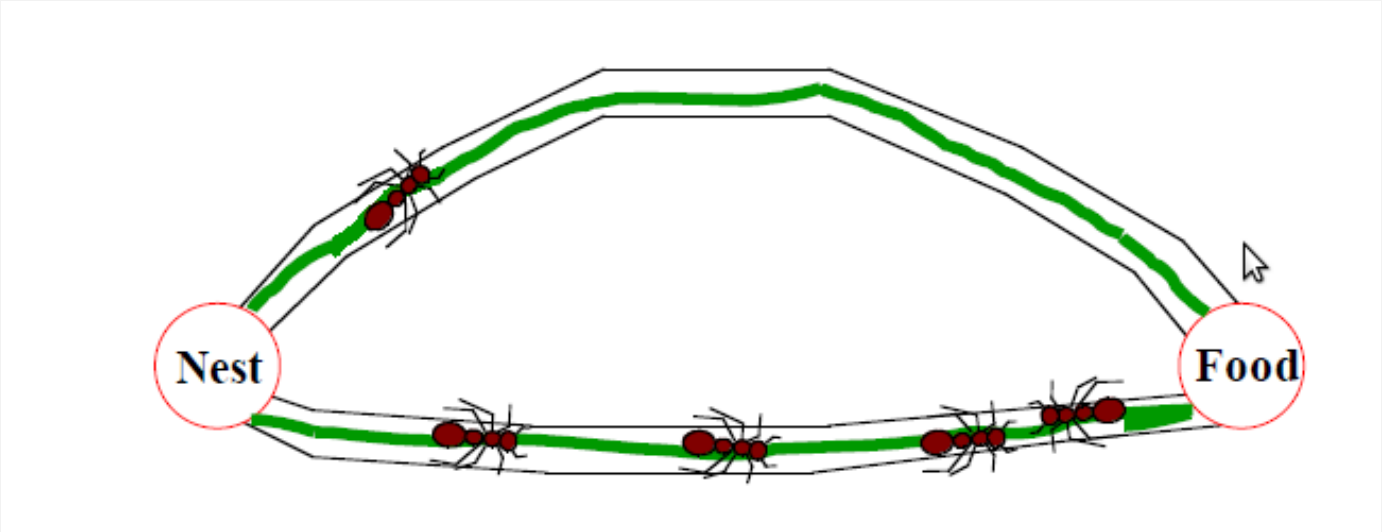

Nest
Food


Nest
Food

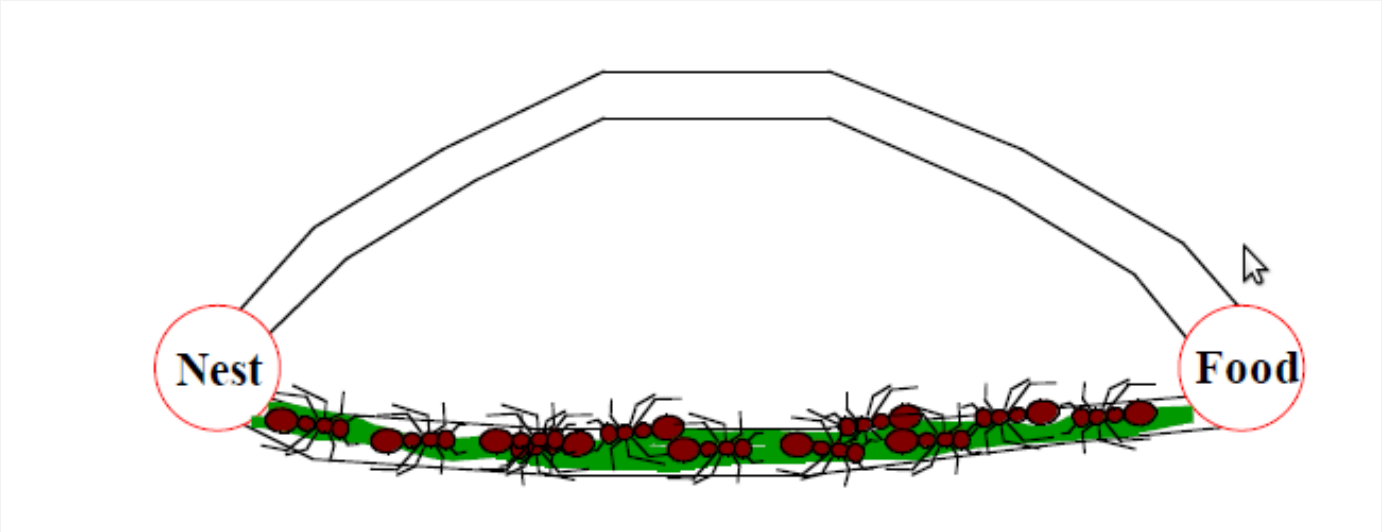

Nest
Food

This is an example of positive feedback! The shorter branch has more pheromone deposited and therefore attracts the new ant. The new ant will also deposit more pheromone, amplifying the shorter branch even more!




Nest
Food

The pheromone deposited on the longer branch evaporates—an example of implicit negative feedback. If the ants forcibly removed the pheromone from that branch then we would say the negative feedback was explicit.



- It is important to note that the ants will usually reach a state of consensus, with the majority adopting one of the two branches. This is true even if both branches are the same length.

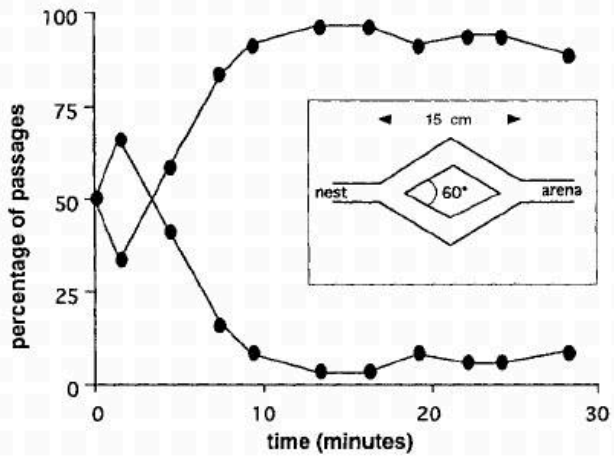
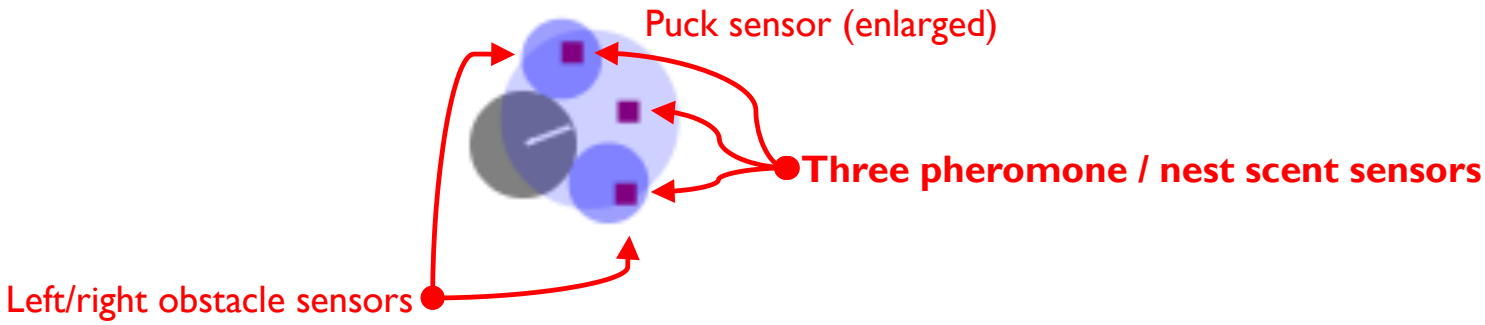


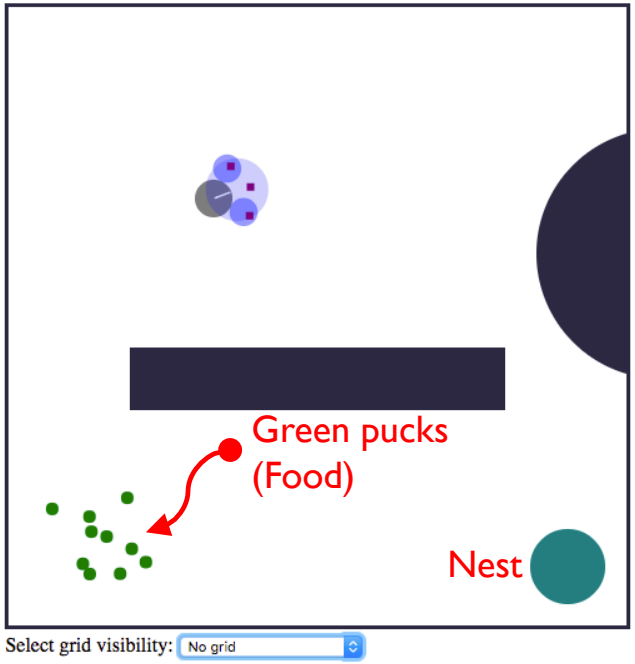
FIGURE 2.1 Percentage of all passages per unit time on each of the two branches as a function of time: one of the branches is eventually used most of the time. Note that the winning branch was not favored by the initial fluctuations, which indicates that these fluctuations were not strong enough to promote exploitation of the other branch. The inset is a schematic representaton of the experimental setup. After Deneubourg et al. [87]. Reprinted by permission © *Plenum Publishing*.

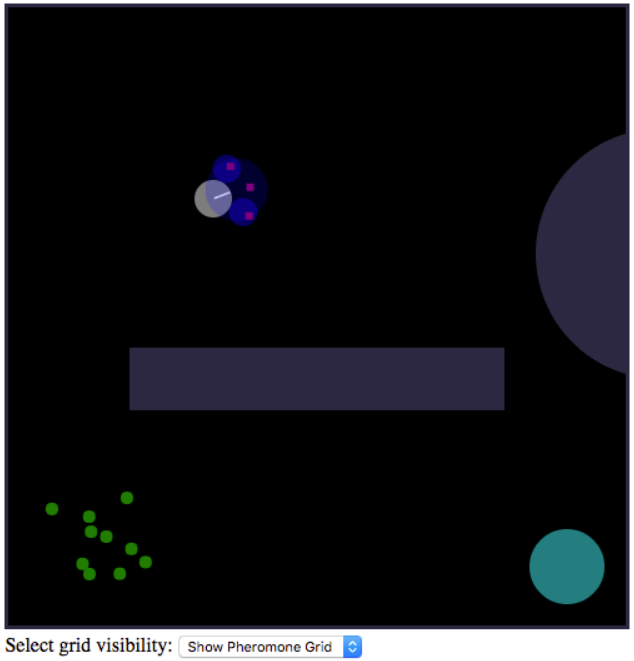
PHEROMONES IN WAGGLE

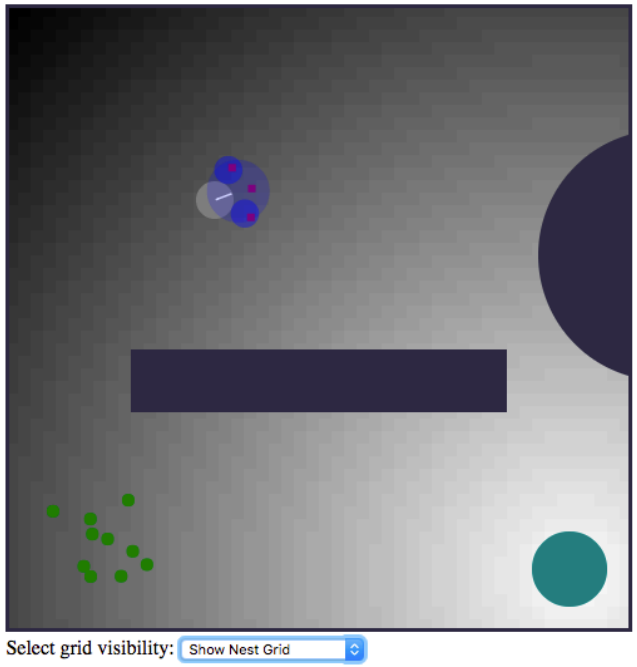
- Go to the main Waggle page (or just hit your browser's back button):
  - http://bots.cs.mun.ca/waggle/
- Select the **Pheromones** level
- The robot has a new layout of sensors:



- The "Select grid visibility" dropdown offers three choices:







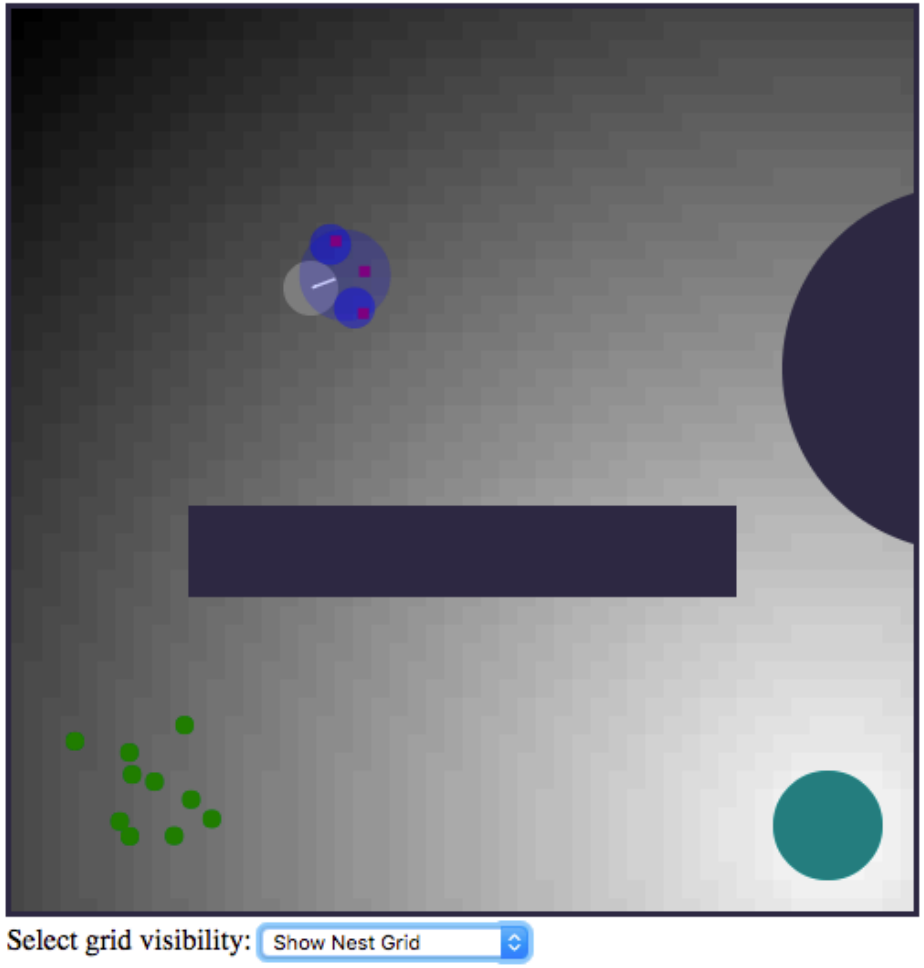
No grid:
- White background just as before

Show Pheromone Grid:
- Displays the quantity of pheromone:
  - 0: Black
  - 1: White
- Appears black initially because no pheromone has been deposited

Show Nest Grid:
- Displays the quantity of nest scent:
  - 0: Black
  - 1: White

NEST SCENT?

- Once grabbing a puck (food) the robot needs to get it back to the nest
- Real ants have a variety of navigation strategies:
  - Visual homing by comparing the current image with one captured at the nest
  - Learning their own visual path
  - Path integration: Integrating velocity over time
- We'll simplify and assume they can somehow smell the nest over a wide area



- Each cell has a value in the range [0, 1] corresponding to its distance to the nest:
  - 0 (Black): Cell is at maximum distance to the nest
  - 1 (White): Cell is at the centre of the nest (distance 0)
- So the nest scent grid forms a hill with the nest on top
- This hill can be climbed by always moving in the direction of the highest sensor value

NEW BLOCKS

- In the "Actions" Category:

  Emit pheromone quantity 10

  - Emit this quantity of pheromones (10 by default)
  - Value is divided by 10 internally

- In the "Sensors" Category:

  Nest scent quantity on left | Nest scent quantity ahead | Nest scent quantity on right

  Pheromone quantity on left | Pheromone quantity ahead | Pheromone quantity on right

  - Gives the quantity of either nest scent or pheromone underneath the left, ahead, or right sensor

EXERCISE #10:

HILL CLIMBING

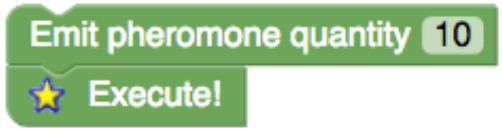
Conditions:

1 robot

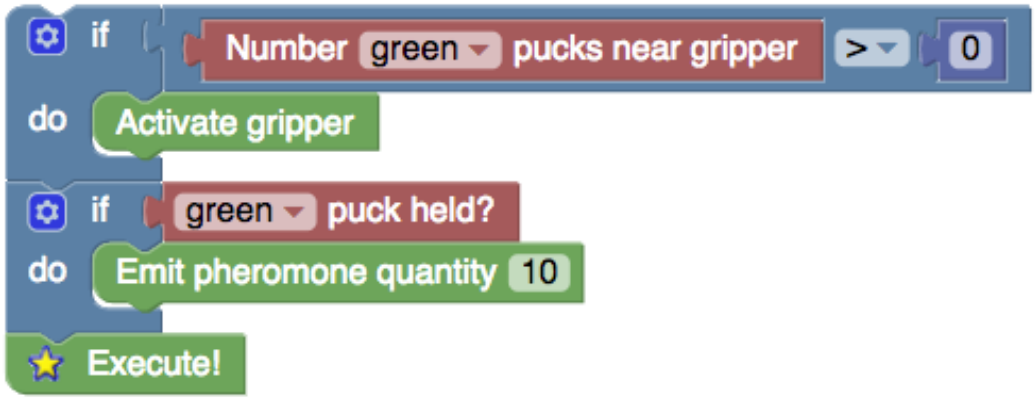
- Consider only nest scent, create a controller that moves towards the direction with the greatest nest scent
- Once working, incorporate obstacle avoidance

- CHALLENGE:
  - The robot should be able to reach the nest from anywhere

- Now experiment with this very simple controller by dragging the robot around:



- Our goal is to develop a foraging controller which will leave pheromones behind whenever the robot has collected a green puck and is on its way home
- Try this controller:



- Like the one above, a robot using this controller needs the human user to drag it around. Try using it to collect all of the pucks in each pile:
  - Notice how a new pile is generated

EXERCISE #11
A6, TASK 5:

PHEROMONE-AIDED FORAGING

Conditions:

Any number of robots

- Combine the following ideas to develop a controller that continually searches for green pucks using pheromones and returns with pucks to the nest:
  - Controller from previous slide
  - Hill climbing controller (ex. 10) for returning to the nest
  - Obstacle avoidance

- CHALLENGE:
  - 10 robots, 10 pucks: Once the controller is working, determine how long it takes to cycle through all three food piles
  - Contrast this with the same controller which deposits no pheromones