Week 2 Boot Camp D term 2018 (100 points)

Due on Friday, March 23 in class.

Each of the 4 problems is worth 25 points. As before, Ashley and I will come and check off your group when you have each one completed.

**Background reading: the Netlogo Programming Guide (online), the Clock Tutorial (posted on the site) and Netlogo 3-D**

* Now that you have some practice with Netlogo programming, it is helpful to read through the Netlogo Programming Guide if you haven’t already.

<https://ccl.northwestern.edu/netlogo/docs/>

Read at least the initial sections on Agents, Procedures, Variables, and the Tick Counter – you will need those for this bootcamp.

* Read the Clock tutorials posted on the site, and make sure you run and understand the bee-demo-2018-v2 simulation.
* Although you are not required to program using Netlogo 3-D for your projects, I want to introduce you to it in case you want to use it. Try running Netlogo 3-D; it should have downloaded to your computer when you downloaded Netlogo.
* To learn how Netlogo 3-D works, take a look at some of the models found in the 3-D Sample Models folder in the Models library. I recommend Fireworks 3D in particular. Try it with the ‘trails’ switch off and on. Try turning off the spin switch, and moving the 3-D view around with the mouse or touch pad. Play around with the sliders.
* Also check out Bouncing Balls Example 3D in the 3-D Code Examples folder.

1. **Create a simulation in Netlogo 3-D with agents that do something different in the morning, afternoon, and nighttime.** Your simulation will demonstrate that you understand how to use the clock, make your agents perform 3-D movement, and create procedures with input parameters.

* Create a new simulation in Netlogo 3-D. Create an ‘hour-of-day’ global variable similar to what was done for the second-counter in the Clock tutorial. Create a monitor to show the value of hour-of-day.
* Now create some agents. Using logic statements, write some code that allows the agents to test what time of day it is – morning, afternoon, or night.
* Write a procedure for each time of day that makes your agents do something obviously different for each time. Make sure at least one of your procedures has parameters, and one of them has your agents do something using the z axis. Have fun with this – don’t feel your agents have to do something realistic.

**2) Using the clock to introduce delay.** The clock can also be used to introduce delay. This can be extremely useful. In class, I will show you how I make simulated people stop when they collide with houses, simulating a door-to-door salesperson. You could use this idea to have a ligand stick to a receptor, two cells adhere in culture, or two organisms mate, to mention just a few possibilities.

* Modify your mushroom hunter simulation from week 1 to make mushrooms die after they turn color, with a delay. As I did with the people, you will need to create a variable or two for your agents, in order to keep track of how much time has passed since the time of the color change. Don’t forget to initialize your variable(s) appropriately!

1. **Creating a membrane.**  Membranes are very important in biology! Knowing how to simulate a membrane will come in especially handy for molecular and cellular simulations. Similar ideas can also be used to create ‘boundaries’ for agents in ecological simulations.

* For the first version of your Membrane model, create a membrane that runs across the view, from left to right. Populate the upper half of the view with molecules, and make them bounce off of this membrane. When the molecules get to the horizontal edges, you can wrap them around, but the molecules at the top of the view will need to ‘bounce off’ that edge (note what happens if you make the world wrap both horizontally and vertically.

Because Netlogo has such an extensive library of sample models, it is always a good idea to look there first when starting a new project, to see if someone else has already come up with what you need, or something close to what you need. In this case, I recommend going to the Models Library and opening the Code Examples folder, and looking at Bounce Example and Look Ahead Example.

* In cells, how do molecules typically get across membranes? Think about ions, for example, that normally would not be able to cross a lipid membrane. How do they do it? Add this mechanism to your simulation. Hint: you will need to create a new type of agent. Create this type of agent in your simulation using a button rather than in the setup code, so that the user decides whether it is present or not. I am not looking for anything complicated. Feel free to brainstorm with your group.

**4) Representing Diffusion**

* Write a program that simulates diffusion (in 2 dimensions) from a point source. Think of this as someone placing a pipette in a cell culture dish that provides a continues stream of some solution (maybe high sugar or salt). Represent the movement of the salt or sugar molecules as following a very simple Brownian-like motion. Be sure to include a way of regulating the number of molecules so that your population doesn’t explode.
* Now add a second point source that generates a different molecule to your simulation. Allow the user to regulate the speed of diffusion of each molecule independently using sliders.
* Add a membrane to your simulation that is permeable to one molecule and not the other.