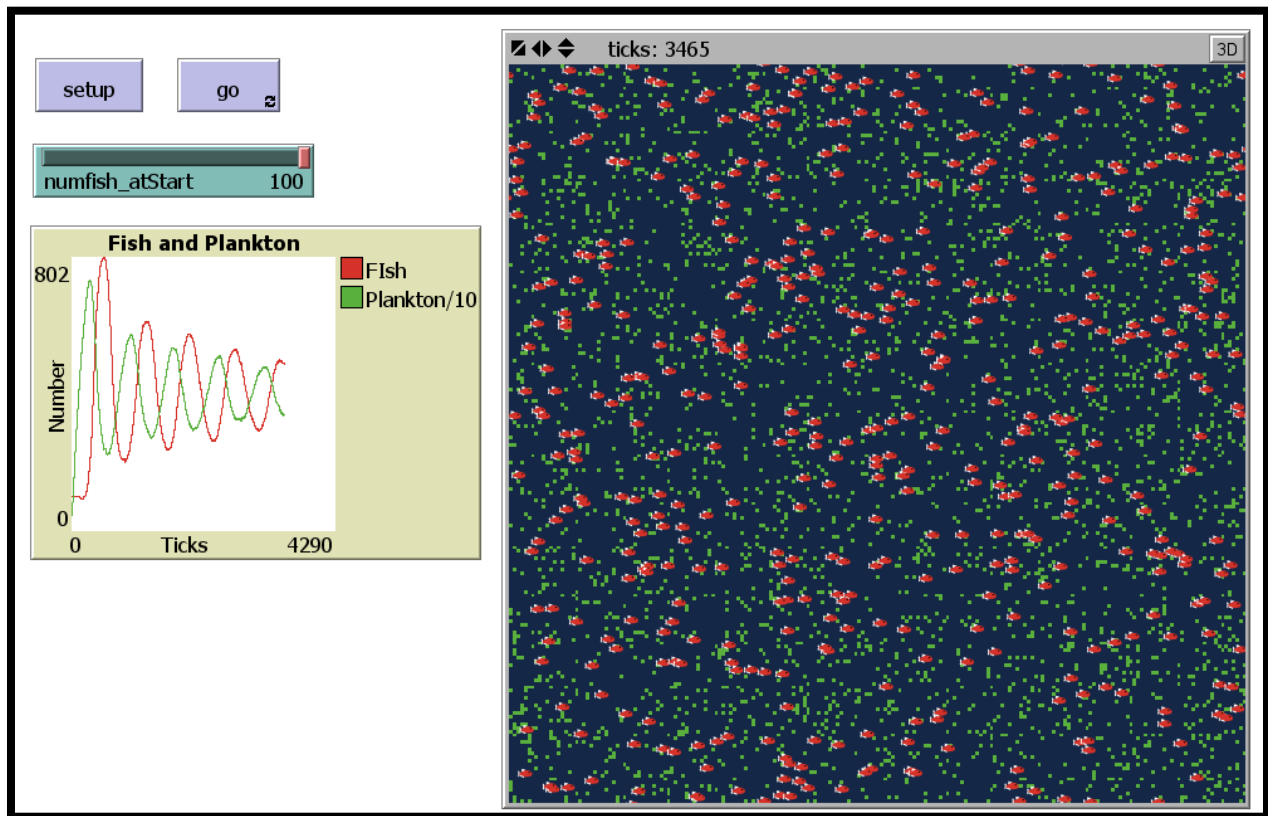


LAB 13: SAVING NEMO: AN ECOSYSTEMS MODEL



In this lab, you will be creating a model that represents a simple ecosystems model. You have fish swimming around eating plankton and having fun! The fish die if they can't find enough plankton and have fishlettes if they can eat enough food. Your basic ecosystem!

The Basic Ecosystems Model:

The basic model has the following requirements:

- Change your world!
 - On the interface tab, choose settings and change the max-pxcor, and max-pycor (suggestion: 100). Also change the size of your patches (suggestion: 3). The world wraps around! Your agents live on a doughnut!
 - Change the color of your world to a deep blue because Nemo lives in the ocean!



- There are Fish in your world!
 - Create a breed of agent (fishes/fish or something like that)
 - Add a slider that allows the user to change the initial number of fish.
 - Fish are big, but NOT too big (size 5 or less)
 - The fish are all the same color
 - Each fish has its own variable to monitor its energy and the initial energy is not zero.
- There is plankton in your world. Plankton make the patches green! Not all the patches are green though! Say about 1 in 100 patches are green at the start!
- Fish do their own thing...
 - Each tick, each fish moves in a wiggle and loses a bit of energy. Remember that the fish have an energy variable that needs to be set when a fish wiggles.
 - Fish eat plankton when they get to a green patch and then the patch becomes a deep blue. Also, when a fish eats plankton, it gains energy so its energy value increases.
 - Fish die if their energy falls to zero or below.
 - Each fish reproduces if it has enough energy to do so. Again, how much energy the fish needs to reproduce is up to you to decide so as to have the model behave as required. Note: Real fish are generally either male or female. They reproduce by the female laying unfertilized eggs, often in a gravel bed. Male fish then fertilize these eggs which (if all goes well for the fish) hatch sometime later. This model will be skipping these details (i.e. using abstraction).
 - When a fish reproduces, it loses energy. Remember that the fish have an energy variable that needs to be set when a fish is hatched.
 - How much energy the fish gains and loses for each activity (moving, eating, and reproducing) is up to you. However, too much or too little makes a difference! There is not one right answer; it's about BALANCE! You will need to decide what those values are so that the model behaves in the required manner (see Stable or Oscillating Population in the rubric).



THE UNIVERSITY *of*
NEW MEXICO

- Plankton happen! Each deep blue, empty water patch randomly grows plankton (and changes the patch color to green) with a probability of 1 in 1000 each tick. Note: To keep the model simple we use abstraction to simplify this so we are modeling the presence of significant concentrations of plankton in the water as a simple green patch. In the real world, plankton are very small and as the density of plankton increases or decreases within a patch of water, the water would gradually change color. In the model, when an empty (deep blue) patch spontaneously generates plankton, you can think of this as a small population of plankton having the right conditions to increase their population density and that this increased density is represented in the model by changing the patch color to green.
- You are also very anxious about your fish and plankton. You create a graph that tracks how many of each are alive in the model as a function of time. Hint: to do this, don't forget about ticks!

That's it for the basic model!

Have Fun Modeling!



THE UNIVERSITY of
NEW MEXICO

Grading Rubric [20 points total]:

[A: 1 points]: Submit the NetLogo source code named:

`W13_firstname_lastname.nlogo` to your instructor:

[B: 1 points]: The first few lines of your code are comments including your name, the date, your school, and the assignment name.

[C: 2 points]: The code in the code tab of your program is appropriately documented with "inline comments".

[D: 3 points]: Correct Setup:

- Your program is set up with the required world settings (many more than the default 33×33 patch world).
- The patches on your world are deep blue except where there are green plankton.
- You start with some plankton (around 1 in 100 patches are green)
- You created a fish breed.
- The fish are all of the same color (unless you do the extra credit with an extra fish breed which will have a different color).
- A slider is used to specify how many fish are created with your setup procedure.
- Each fish is set to a larger than size 1 but smaller than size 5 and has an energy variable which has an initial value that is NOT zero.

[E: 4 points]: Your fish behave correctly:

- Fish lose energy when they move.
- Fish eat plankton and gain energy when they get to a green patch, turning the patch deep blue.
- Fish die if they do not have enough energy and reproduce if they have enough energy.

[F: 3 points]: Your plankton behave correctly: They disappear (patches turn deep blue) when they are eaten and they reproduce once in a while (patches turn green).

[G: 3 points]: Input and Output:

- You have a slider that inputs the number of fish in the model.
- You have a graph that shows the number of fish and the plankton as the model progresses.



[H: 3 points]: Stable or Oscillating Population: When your model is run, neither the fish nor the plankton die out completely. Rather, the populations of each either oscillate or stabilize. In order to accomplish this, you may need to adjust the amount of energy lost or gained for moving, birthing, eating and at birth. There should be fish and plankton after 10,000 ticks. Remember you can change the speed at which the program runs!

[EXTRA CREDIT: 5 points]: Your program has separate procedures for fish eating, moving, reproducing and dying. You call each of these procedures in your "go" procedure.

[EXTRA CREDIT: 2 points]: Set the shape of the fish breed to something other than the default arrowhead shape.

[EXTRA CREDIT: 10 points]: You add a second breed of fish. The new breed must have a different color. It must move, eat, reproduce and die as the original breed; however, the new breed must have a different initial energy value, and gain or lose energy differently than the first type of fish. The new breed also moves differently than the first breed (turns left and right in the wiggle walk by a different maximum angle). For example, the new breed might move faster but lose more energy from moving. Also, the new breed might need less energy to reproduce, but the baby fish start their life with less energy than the baby fish of the first breed.

As part of creating a new breed, modify the chart to plot three lines: one for the population of each fish breed and one for the population of the plankton.

To gain these extra credit points, you must choose energy gains and losses so that the populations of all breeds remain stable or oscillating (none of the breeds die out).