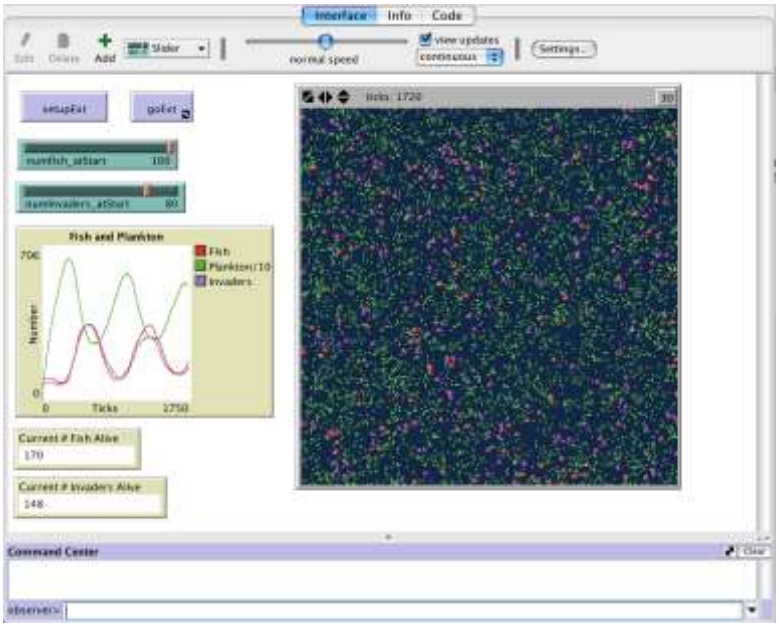
CS108L Computer Science for All
Module 6: Saving Nemo, the Invasion:
An Ecosystem Model EXTENSION



Since we are modifying the graph, you will need a separate program from your Lab 6 program (e.g. you must have a separate NetLogo file labeled as described in rubric 'A'), however you may copy your code from Lab 6 to the extension program.

In this extension an invasive breed of fish has found your ecosystem and is now competing with your original fish for the plankton, in order for both breeds to survive the invasive breed must adjust to its new environment.

Modify the program interface (buttons, sliders. etc…) as follows;

- Change the setup button to setupExt button.
- Change the go button to goExt button.
- Add a slider for the start number of invasive fish.
- Add a monitor for the current number of invasive fish alive.
- Modify the graph to track the number of invasive fish.

Modify the program as follows;

- Add a new breed that is the invasive fish (you may name them in anyway you like).

- The invasive fish still looks like fish but are a distinct color (match the color on the graph).
- Invasive fish move differently (turns left and right in the wiggle walk by a different maximum angle).
- Have different initial energy.
- Gain or lose energy differently. For example, the new breed might move faster but lose more energy from moving.
- Need a different energy to reproduce. 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 (for example).
- The original fish variables (plankton and breeding energy) settings are set to the best of the 3 trials you submitted for Lab 6. These are now constant they may not be changed.
- The original constants are still constant.
    - NOTE: If you are having issues finding a good balance, you may modify the plankton growth rate from your previous program, but keep it constant for all trials for the extension.

You will do the same trials as you did for Lab 6 and fill out the google sheet for the extension. The objective is to have both the original and the invasive fish survive to 10,000 ticks. To do this modify the invasive fish variables (plankton and breeding energy) so that at 10,000 ticks both fish have a remaining count > 0.

Module 6 Extension: Saving Nemo, the Invasion Grading Rubric (20 Points Total)		
Done	Points	Task

	2	A:
		• Submit a NetLogo source code with the file name: M1.*firstname*.*lastname*.nlogo.
		• The first few lines of your Code tab are comments including the following:
		`;Student's Name:`
		`;School:`
		`;Teacher's Name:`
		`;Date:`

	3	B: General Code Design • The code in the Code tab of your program is appropriately documented with "in-line comments". • Your program is logically ordered, formatted and indented so that it is easy to read the code and understand its function. • You program has separate procedures for fish eating, moving, reproducing and dying. You call each of these procedures in your "goExt" procedure. • The info tab describes the details of you model including a general description, how it works, how to use it and items from F below.

	2	C: Correct Setup: • Your program is setup the same as lab 6 with the following additions. • Change the setup button to setupExt button. • Change the go button to goExt button. • Create a new breed of fish (the invasive fish) • The invasive fish are a different color from the original fish.
	5	D: Your fish behave correctly: • Your original fish behave the same as Lab 6, the best of the 3 trials is used for their settings. • The invasive fish move differently than the original fish • The invasive fish use a different amount of energy to reproduce than the original fish (note that this may also affect the amount of energy a baby invasive fish starts with). Since this is an adjustable attribute, keep in mind it may not be the same as the original fish when adjusted.
	4	E: Input and Output: • You have a slider for each breed of fish that inputs the number of fish of a breed in the model. • You have a graph that shows the number of fish by breed and the plankton as the model progresses. • You have a monitor for each breed of fish that displays the number of each breed of fish currently alive.
	4	F: Model Parameters for a Stable or Oscillating Population: • When your model is run, neither the fish (of any breed) nor the plankton die out completely. Rather, the populations of each either oscillate (cycle around a value) or stabilize. To accomplish this, you may need to adjust the amount of energy lost or gained for moving, reproducing, 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! • **NOTE:** There are many variables that affect this. Therefore simplify the problem by setting the 'fish population', 'plankton population', 'plankton growth rate' and 'movement energy' (energy fish loses as it moves) to a constant (all trials use the exact same values for these items). Additionally, the original fish variables are set as constant according to the best of the 3 trials from lab 6. In the case of issues with stabilization, you may adjust the plankton growth rate. • Report in a tab on the google sheet for the extension (which can be reached from Module 6 web page) your above constants (which should be the same for all trials and includes your original fish settings), your variables of the invasive fish ('plankton energy' which is energy gained from eating plankton and 'breeding energy' which is

		the energy lost for breeding) and your result ('fish currently alive' at 10,000 ticks) for both the original and invasive fish. Report only the 3 best trials (highest number of fish still alive at 10,000 ticks, this must be greater than 0 for both breeds, but otherwise may be different). • NOTE: While you are only turning in the 3 best trials, you should aim to have at least 10 trials with different variable settings (not constant settings) that result in the fish currently alive at 10,000 ticks to be greater than 0 (e.g. at least 1 fish alive at those ticks). This will likely take many, many different runs that simply don't work out, be prepared to spend a large portion of your time just running the program and looking for those 10 trials with a result above 0. So start programming early!