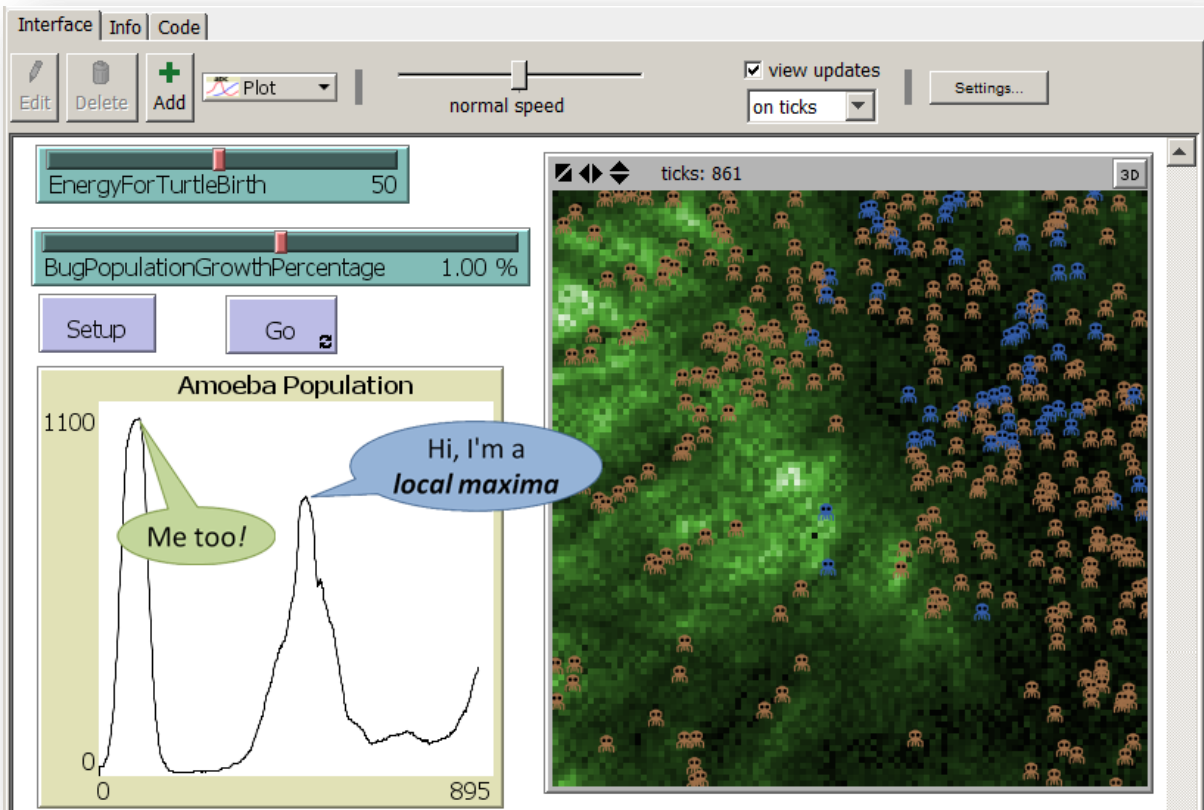


## LAB 7: EXPERIMENTING WITH LIFE AND DEATH



Augmented screen capture showing the required components:

- 2 Sliders (as shown)
- 2 Buttons (as shown)
- 1 Plot (as shown)

min-pxcor = -50, max-pxcor = 50, min-pycor = -50, max-pycor = 50

### Model Overview:

*Experimenting with Life and Death* is a Netlogo model with just a few modifications to the *Energizer Turtles* model developed in the last lab. The primary focus of this lab is gain practice using Netlogo to run a systematic set of experiments known as *parameter sweeping*. (see the video "Designing and Running Experiments in NetLogo").



### Sliders:

Three of the sliders from the *Energizer Turtles* model have been converted to global variables with values that are set in the setup procedure and never changed.

#### globals

```
[
  NumberOfTurtles
  MaxBugsTurtleCanEat
  MaxBugsPerPatch
  WiggleAngle
]

to setup
  clear-all
  reset-ticks

  set NumberOfTurtles 25
  set MaxBugsTurtleCanEat 10
  set MaxBugsPerPatch 15
  set WiggleAngle 5
```

↓

Of course, “variables” that never change can be replaced in the code with literals. For example, in the setup procedure, we could replace:

```
create-turtles NumberOfTurtles
[
```

with

```
create-turtles 25
[
```

However, and especially if the same quantity is used in more than one place in your code, it is usually easier for a human to read the code when meaningful identifiers are used rather than literals.

In place of the two sliders we made into global “variables” with unchanging values, a new slider has been added: `EnergyForTurtleBirth`. (with a domain from 2 through 100 in whole number steps).

Finally, the `BugPopulationGrowthPercentage` slider has had its maximum value increased to 2.00%.



### Turtle Setup:

- 1) With the exception of setting the global variables shown above and possibly changing the shape of the turtle, the setup procedure is unchanged from the *Energizer Turtles* model. Note: I picked the "monster" shape because in this model the turtles reproduce asexually, and the "monster" shape was the nearest predefined Netlogo shape I could find to an amoeba or some other single-celled organism. If you want to use the "monster" shape, or any of the extended Netlogo shapes, select "Tools" → "Turtle Shapes Editor" → "Import from Library..." and select your favorite. Make a note of the shape's name. Then, you can use that name in the code tab with Netlogo's `set shape` command.
- 2) I started each of 25 turtles with a random color, but "something happened" from when I started the model to when the screen capture was taken - leaving the run with just two colors of turtles (or amoebas or monsters or whatever).

### Patch Setup:

- 1) No changes form the *Energizer Turtles* model.

### Each Turtle on Every Tick (Eat, Turn, Walk, Birth and Death):

**Eat:** No changes form the *Energizer Turtles* model.

**Turn:** In the *Energizer Turtles* model, each student could choose the wiggle angle. In this model, the right and left turn in wiggle walk must be a uniformly distributed random number between 0 and the global `WiggleAngle` (set to 5 degrees). **NOTE: an extra credit option for *Energizer Turtles* was to make "seeker turtles". DO NOT use "seeker turtles" in this model.**

**Walk:** In the *Energizer Turtles* model, the more energy a turtle has, the faster it moves and when a turtle moves, it loses energy equal to the distance it moved. In this model, each turtle's speed is set in the same way as in the *Energizer Turtles* model, however, each turtle loses 1 unit of energy each tick - regardless of how quickly it is moving or even if it is not moving at all.

**Death:** In the *Energizer Turtles* model, a turtle could never have less than 0 energy and could never die. In this model, a turtle dies if it has less than or equal to 0 energy ☹️



**Birth:** In this model, when a turtle has energy greater than or equal to the slider value of `EnergyForTurtleBirth`, then two things happen (in order):

- 1) The turtle's energy is divided by 2.
- 2) A new turtle is hatched. This is done with Netlogo's `hatch` command. The `hatch` command creates a new turtle which inherits of all its variables, including its location, from its parent. This is analogous to asexual reproduction in the natural world by which offspring arise from a single parent as an exact genetic copy of that parent.

**Each Patch on Every Tick (Multiply, Spread and Color):**

**Multiply:** No changes form the *Energizer Turtles* model.

**Spread:** No changes form the *Energizer Turtles* model.

**Color:** No changes form the *Energizer Turtles* model.

**Plot:**

Your program must include a plot of time verses total turtle population (I called it "Amoeba Population" in the screen capture because if you half close your eyes and turn your head mostly away from the display, the shape I picked looks just like an amoeba). I set the plot to "auto scale" with the "Pen Update Command":  
`plot count turtles`

**Grading Rubric [20 points total]:**

**[A: 1 point]:** Submit Netlogo source code named: `w7.firstname.lastname.nlogo`.

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

**[C: 1 point]:** The code in the code tab of your program is appropriately documented with "inline comments".

**[D: 2 points]:** Your program's interface includes the required 2 sliders, and 2 buttons and the required plot. As usual, your layout must be neat, but can be organized and labeled as you like.

**[E: 2 points]:** Your setup procedure must set up the turtles as required above. Note that the turtle shape is not a requirement - pick the shape that most pleases you.



[F: 2 points]: Your go procedure must *Eat*, *Turn*, and *Walk* the turtles as required.

[G: 2 points]: Your go procedure must kill the turtles as required. Note: as the "turtles" in this model are nothing more than bits of light and electrons in a computer, "killing" them does not actually cause any harm to anything. You may sleep peacefully the night after conducting this experiment. However, please reframe from sleeping in the classroom.

[H: 2 points]: Your go procedure must birth the turtles as required.

[I: 5 points]: Pair up with a classmate. You and your classmate must run a total of 18 experiments: 6 (each with different parameters)  $\times$  3 (repetitions to be averaged). You and your labmate must each choose a different set of 6 parameters so that the two of you together produce a parameter sweep of the model with a total of 12 different settings (each setting must be repeated 3 times). Record the results in a "Things We Noticed" section of your program's info tab. It is up to the two of you to choose settings that provide a "good" sweep of the parameters. The three repetitions with the same parameter settings is required since the model is stochastic (involves randomness). You must run each experiment for 2,000 ticks or until all the turtles are dead - whichever comes first. For each set of 3 repetitions of the 12 parameter settings, record the following:

- a) Name of person who ran the experiment.
- b) `EnergyForTurtleBirth` setting.
- c) `BugPopulationGrowthPercentage` setting.
- d) Total number of ticks in repetition 1, 2, and 3. Also, the *average* total number of ticks in the three repetitions.
- e) Final turtle (or whatever) population in repetition 1, 2, 3 and the *average* final population in the three repetitions.
- f) Number of population hilltops (more properly called "local maxima") in each of the three repetitions and the *average* number of local maxima.

[J: 2 points]: In a section of your program's info tab titled "Hypothesis", write a paragraph or two explaining any patterns you believe exist in the data and your guess as to why those patterns are as they are.