CSCI 190: HW4: *Erosion*

Due on **Wednesday, March 9 at 8:00am** (before the start of class.)

Your solution should be submitted to the CSCI 190 drop box on middfiles. Your program MUST be named as follows**:YourName-HW4.nlogo**

Programs (from here on) will be graded on:

1. correctness (70pts)
2. structure (10pts)
3. style (10pts)
4. interface (10pts).

Style includes good commenting as well appropriate use of constants and variable names. Interface includes appropriate use of sliders, switches, and buttons. (Later, models will also be required to have an "information" session.) Correctness is based on how well it accomplishes the assigned task.

Overview: You will be writing a NetLogo model to simulate erosion. Your simulation should begin with a terrain: a set of patches with elevations (altitudes), colored with some scale-color to visualize the elevations (with darker = lower and ligher = higher). Drops of water will fall on that terrain and flow downhill. As water flows over a patch, it causes erosion: a lowering of the altitude of that patch. Continue to recolor the patches after each turn, thus as the elevations patches lower, they should darken. Your model should show the force of erosion through the darkening of the lines where the water flows.

Further Instructions and Details: You should use only the NetLogo commands already presented in class (including lab), plus the following which you can read about:

- turtle commands: `die setxy move-to`
- reporters: `random-xcor random-ycor`
- built-in constants: `min-pxcor min-pycor max-pxcor min-pycor`

Additionally:

- You should use a more finally grained world, 201x201 patches, with a patch size of 2 pixels, that does not wrap horizontally or vertically.
- The rain should be represented as agents/turtles that move around the world (flowing downhill, of course). A unit of rain (that is, a turtle) should be represented as a blue circle.
- Users should have some control over the rate of rainfall as well as the rate of erosion.

Note: There is an erosion model in the Earth Sciences section of the Models Library. You may *not* make use of this model. In fact, you will be taking a very different approach than the erosion model, and that approach (which doesn't have agents) would only be confusing. Your model should look more like the Grand Canyon model, but the Grand Canyon model shows the flow of rain and doesn't actually model erosion, and it has a fixed terrain.

For the Full 70 Pts on Correctness:

1. Your terrain should have a random element and should be uneven, but most of the water should eventually reach the edge as it flows downhill, without getting stuck in lake or pond.
2. Show the drops of water moving around the terrain, and over time the user should see the effects of erosion.

Partial Solution (Maximum 60): You could start with a single uniform hill – a cone shaped terrain where there is a peak patch and every other patch decreases in altitude with the distance from that peak. This should be easy to create, but it won't look as neat. You might also try a terrain such as the land-water terrain, but this will often allow the water to pool up rather than run off.

Hints and Suggestions:

1. Start raindrops in a random location using `setxy` and `random-xcor` `random-ycor`.
2. When drops of water reach the edge of the world—or get trapped in a local minimum—they should disappear. (Use the `die` command. A Turtle executing the `die` command ceases to exit.)
3. When moving drops of rain, having them `move-to` a neighboring patch every step, rather than moving using the `fd`. This will ensure that the drop causes the same amount of erosion on each patch.
4. For a different sort of terrain, try starting with a single mountain peak and running ridges down from that peak in a few different directions. *You can use a temporary turtle to wander in some direction away from the peak and create a ridge as it goes, and then have it die. That ridge-creating turtle would not actually be an agent in the simulation but would just be used to create the simulation.*

Extra Credit (up to 5 pts):

Find a way to mark ever local valley—a local altitude minimum from which water could not escape—as a lake. When a raindrop reaches a lake, it also dies as it would when it reached the edge of the world.