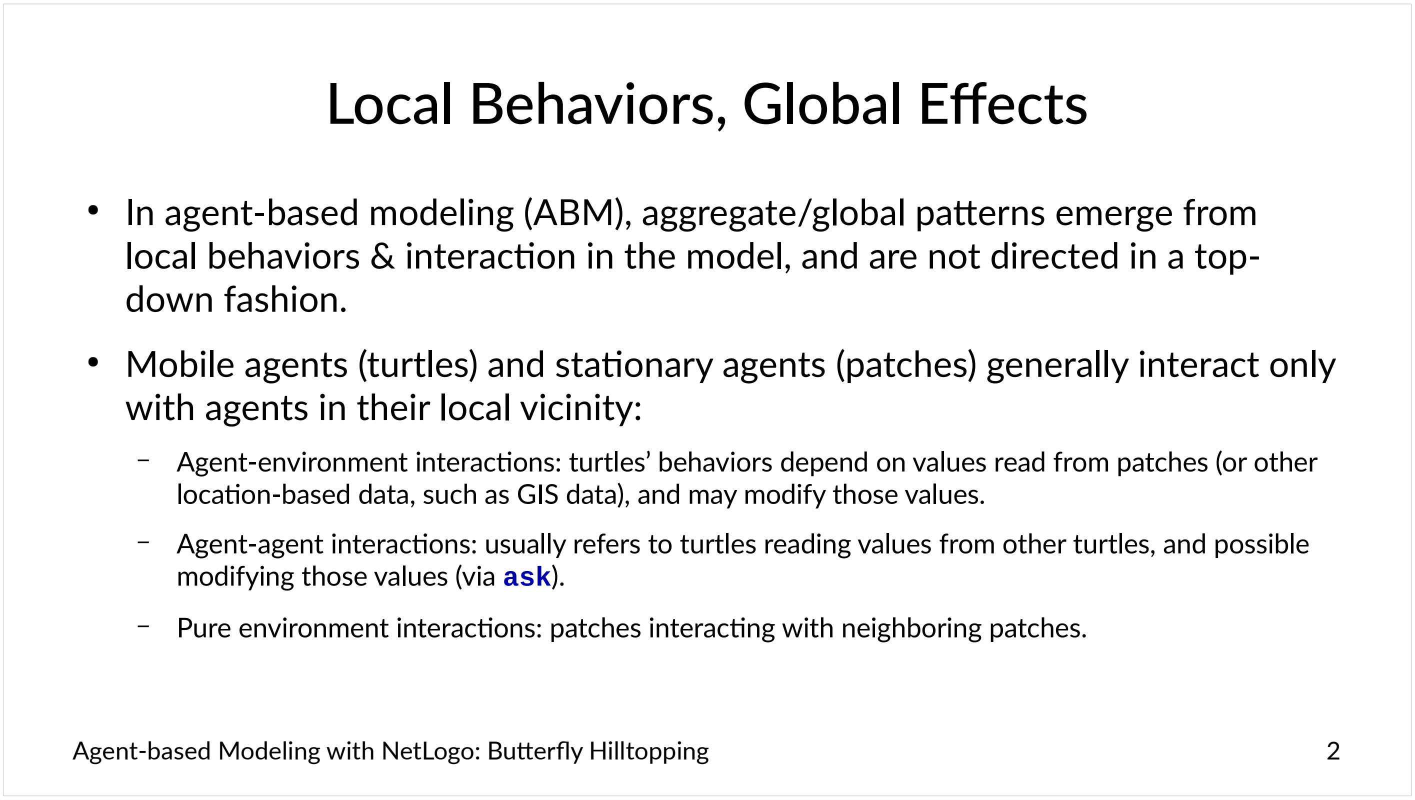
Agent-based Modeling with NetLogo: Butterfly Hilltopping

New Mexico Supercomputing Challenge
Nick Bennett & Janet Penevolpe



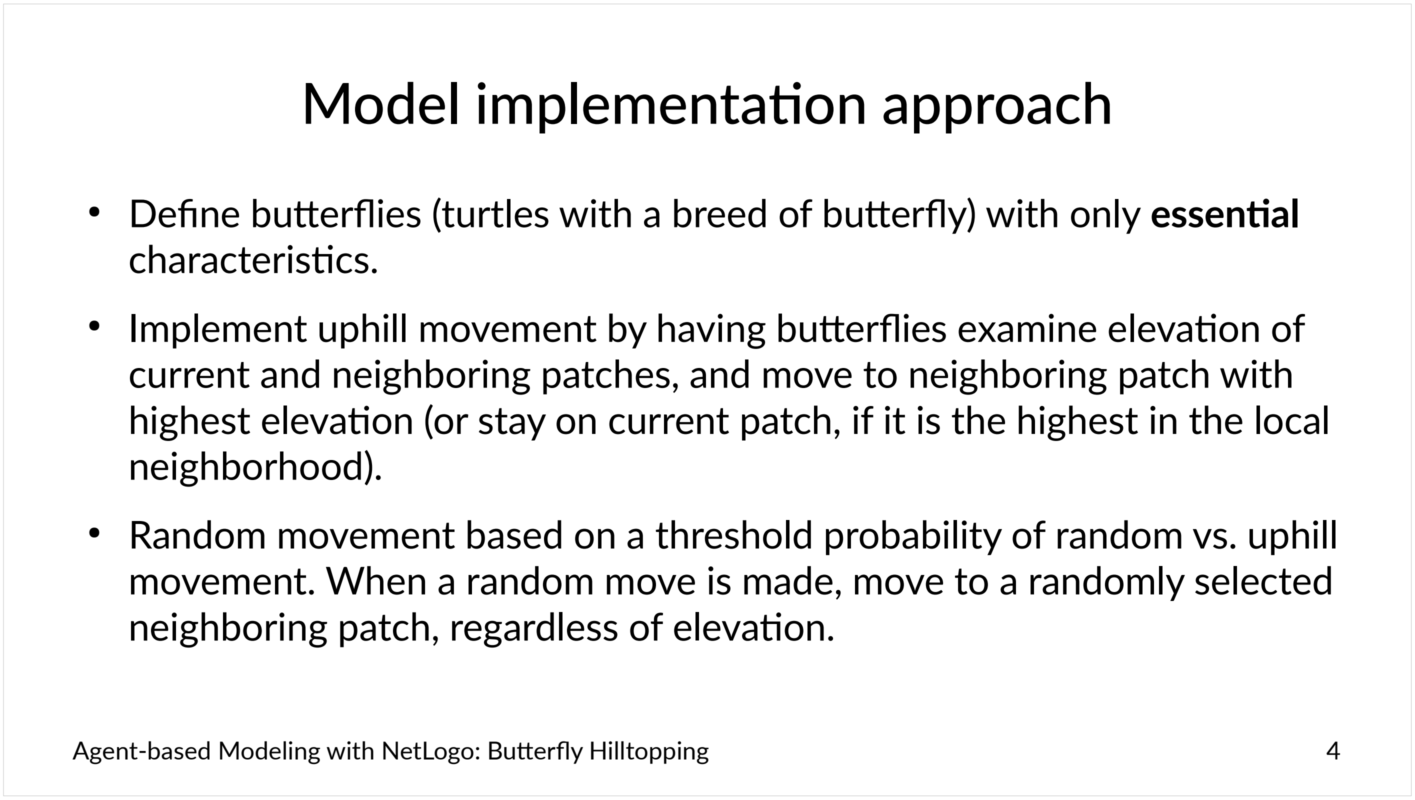
Local Behaviors, Global Effects

- In agent-based modeling (ABM), aggregate/global patterns emerge from local behaviors & interaction in the model, and are not directed in a top-down fashion.
- Mobile agents (turtles) and stationary agents (patches) generally interact only with agents in their local vicinity:
  - Agent-environment interactions: turtles' behaviors depend on values read from patches (or other location-based data, such as GIS data), and may modify those values.
  - Agent-agent interactions: usually refers to turtles reading values from other turtles, and possible modifying those values (via **ask**).
  - Pure environment interactions: patches interacting with neighboring patches.

- Note that "agents" in NetLogo doesn't just mean turtles; patches and links are also agents. However, in common usage, when we say "agent-environment interactions", we usually mean turtles interacting with patches; similarly, "agent-agent interactions" are usually turtles interacting with turtles.
- Patches interact with other patches when something "flows" across the terrain, and for whatever reason, we don't or can't model that flow with turtles. For example, we might model the spread of fire across a terrain this way, rather than using "fire turtles".

Butterfly mating behavior

- Butterflies observed to move uphill in general—but with some apparent randomness.
- Congregating (for example, at hilltops) provides mating opportunities.
- Research questions:
  - Does the apparent randomness sometimes exhibited by butterflies help them to gather at the tops of hills?
  - If “yes”, then would different kinds of terrain result in different mixes of random vs. uphill movement evolving in different species/populations of butterflies?



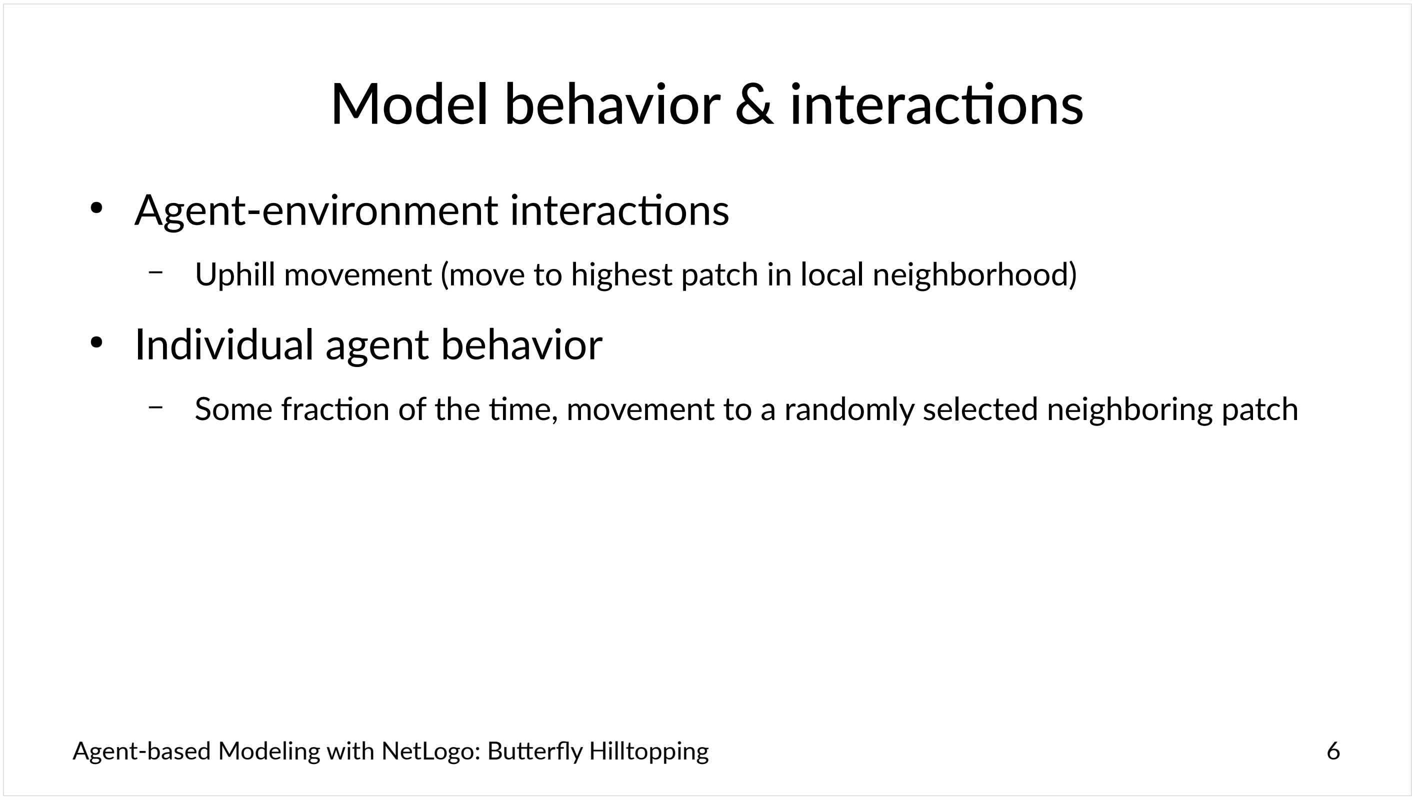
Model implementation approach

- Define butterflies (turtles with a breed of butterfly) with only **essential** characteristics.
- Implement uphill movement by having butterflies examine elevation of current and neighboring patches, and move to neighboring patch with highest elevation (or stay on current patch, if it is the highest in the local neighborhood).
- Random movement based on a threshold probability of random vs. uphill movement. When a random move is made, move to a randomly selected neighboring patch, regardless of elevation.

- What does “essential” mean?
  - Depends on aims & economics.
  - In any event, when modeling any non-trivial system, it’s practically impossible to capture all characteristics.
- This form up uphill movement is very crude, but often appropriate as a first-order approximation. What other algorithms for uphill movement might we consider?
- As described, the random portion of movement is a 2-dimensional random walk. What other stochastic movement regimes have we implemented in other models? (“Wiggling”, etc.)

Defining agents & characteristics

- Turtle breed for butterflies
  - Built-in characteristics (who #, location, orientation, etc.)
- Patches (fixed agents in the environment)
  - Built-in characteristics (location)
  - Elevation



Model behavior & interactions

- Agent-environment interactions
  - Uphill movement (move to highest patch in local neighborhood)
- Individual agent behavior
  - Some fraction of the time, movement to a randomly selected neighboring patch

- How can we generate some event randomly, with probability *p*?
  - Such a random event is a Bernoulli trial: It occurs with probability *p*, or does not occur with probability (1 - *p*). In other words, a Bernoulli random variable is one taking the values 1 or 0, success or failure, on or off, yes or no—or (in this case) uphill or random.
  - We can sample from a Bernoulli distribution by sampling from a continuous uniform distribution on the interval [0, 1). If the sampled value *X* < *p*, then the value of the Bernoulli variable is 1; otherwise, it’s 0. Essentially, we’re flipping a weighted coin.
  - Conveniently, `random-float 1` samples from the continuous uniform distribution on [0, 1).

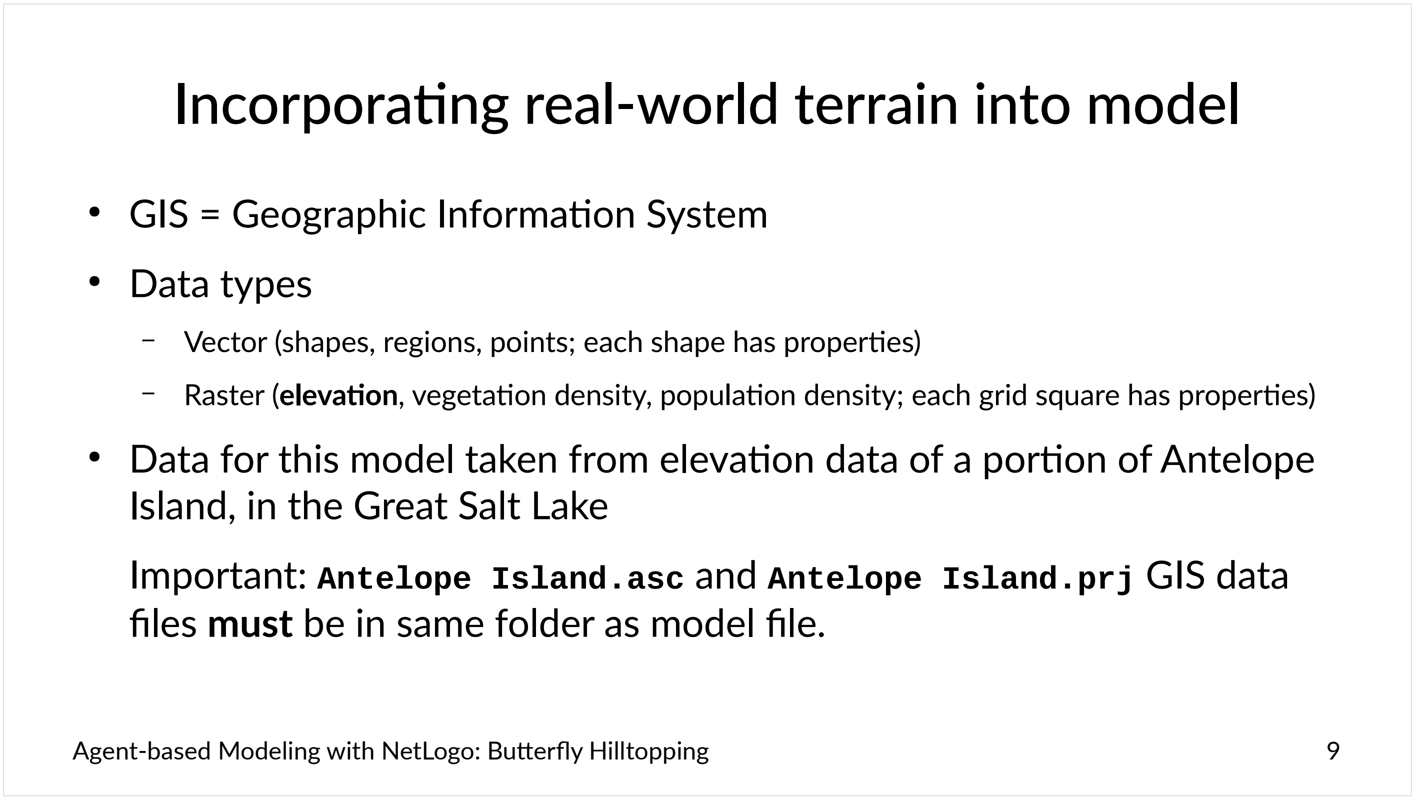
Measuring aggregate agent behaviors

- Grouping behavior (mating possibilities)
  - Average local relative density = (Average number of butterflies in a small circular neighborhood around each butterfly) / (area of that neighborhood) /(total population)

Artificial terrain/patches

Before using real world data, a simple, artificially generated terrain will be used.

- Simple slopes, leading up hill to a single peak.
- Initially smooth; some randomness added.



Incorporating real-world terrain into model

- GIS = Geographic Information System
- Data types
  - Vector (shapes, regions, points; each shape has properties)
  - Raster (**elevation**, vegetation density, population density; each grid square has properties)
- Data for this model taken from elevation data of a portion of Antelope Island, in the Great Salt Lake

  Important: **`Antelope Island.asc`** and **`Antelope Island.prj`** GIS data files **must** be in same folder as model file.

- In fact, the GIS files can be located in another directory (though all related GIS files must be in a single directory); however, the code to load the GIS data must then specify the location, including directory, relative to the NetLogo model file location.

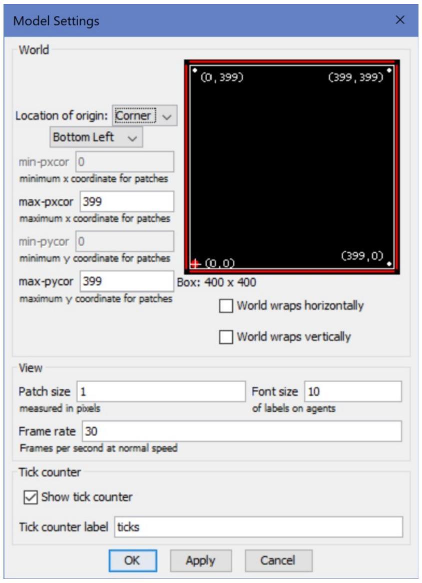
Let’s write the code!

- All code in this document is shown on a light-gray background.
- The code you’ll need to write at each step is shown in boldfaced type, colored as it would be in the NetLogo editor. When a step is adding code to a procedure you’ve already written, the previously written code is shown in gray.
- Good code has comments that make clear how the code works! These are shown in medium gray italic type in the slides.

  For today, you do not need to include these comments in your model; **however**, for your project models, include informative comments. (Remember, most Supercomputing Challenge judges don’t know NetLogo—and even those that do know NetLogo generally won’t have the time to figure out the details of your code.)

Configure the model: World settings

- Location of origin: Corner
- max-pxcor: 399
- max-pycor: 399
- World wraps horizontally: off
- World wraps vertically: off
- Patch size: 1
- Font size: 10
- Frame rate: 30
- Show tick counter: on
- Tick counter label: ticks



Model Settings

World

Location of origin: Corner
Bottom Left

min-pxcor 0
minimum x coordinate for patches

max-pxcor 399
maximum x coordinate for patches

min-pycor 0
minimum y coordinate for patches

max-pycor 399
maximum y coordinate for patches

Box: 400 x 400

□ World wraps horizontally

□ World wraps vertically

View

Patch size 1
measured in pixels

Font size 10
of labels on agents

Frame rate 30
Frames per second at normal speed

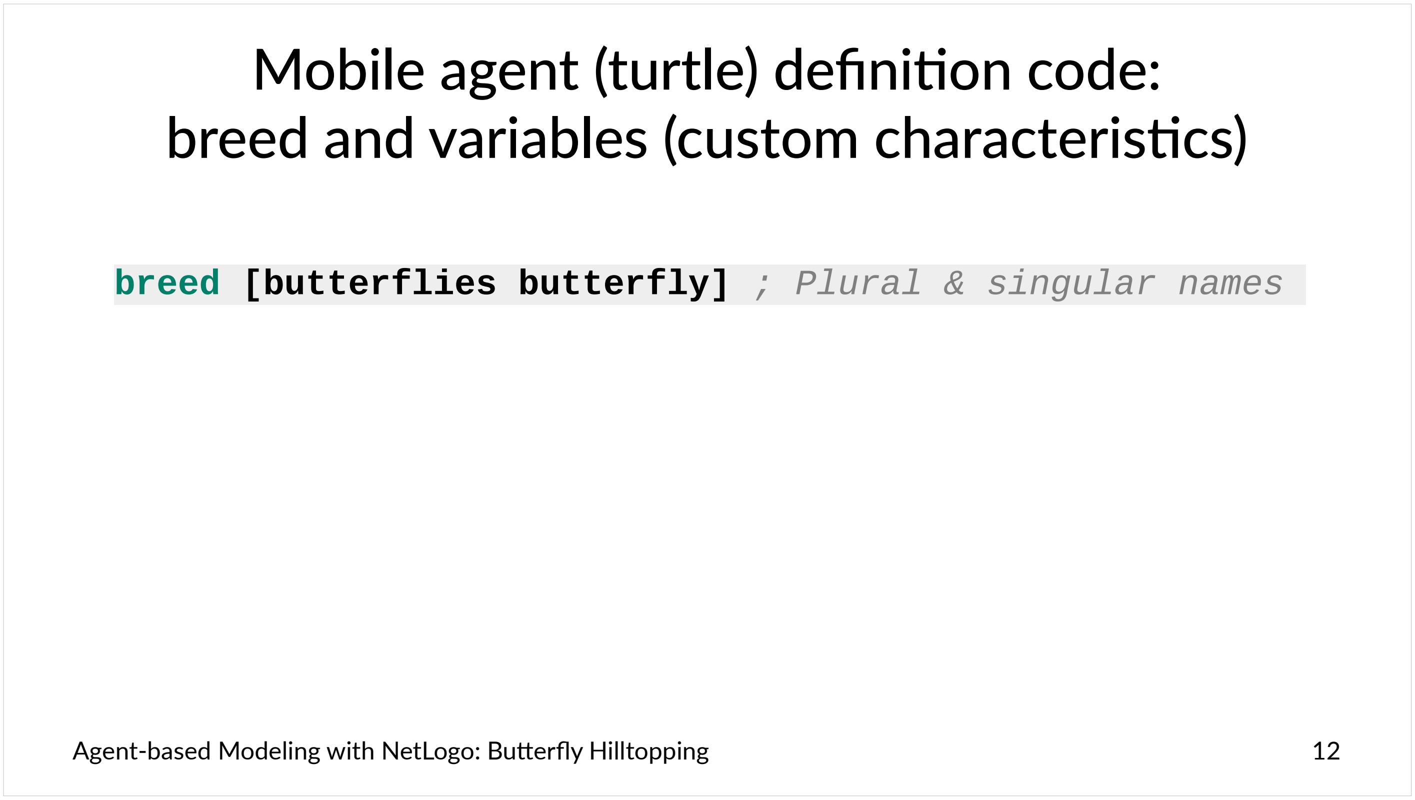
Tick counter

☑ Show tick counter

Tick counter label ticks

- Model settings can be accessed by clicking the **Settings...** button near the top of the NetLogo window.

Model: **`Butterfly Hilltopping (Phase 0).nlogo`**



Mobile agent (turtle) definition code: breed and variables (custom characteristics)

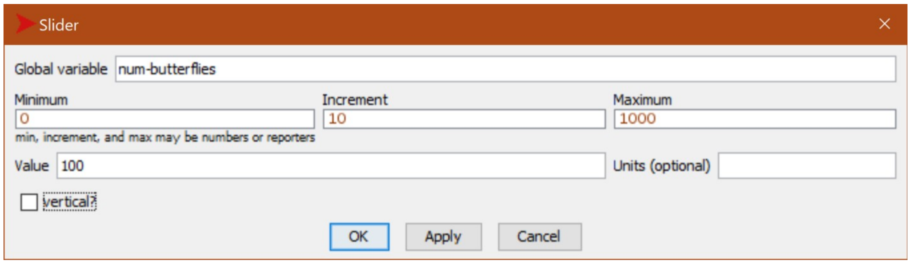
```
breed [butterflies butterfly] ; Plural & singular names
```

- Plural and singular names are both required. They must be different, and must appear in the order shown (i.e. plural first).

Environment definition: Patch variables (custom characteristics)

```
patches-own [
  elevation
]
```

Controlling number of butterflies: `num-butterflies` slider



Global variable num-butterflies

Minimum 0

Increment 10

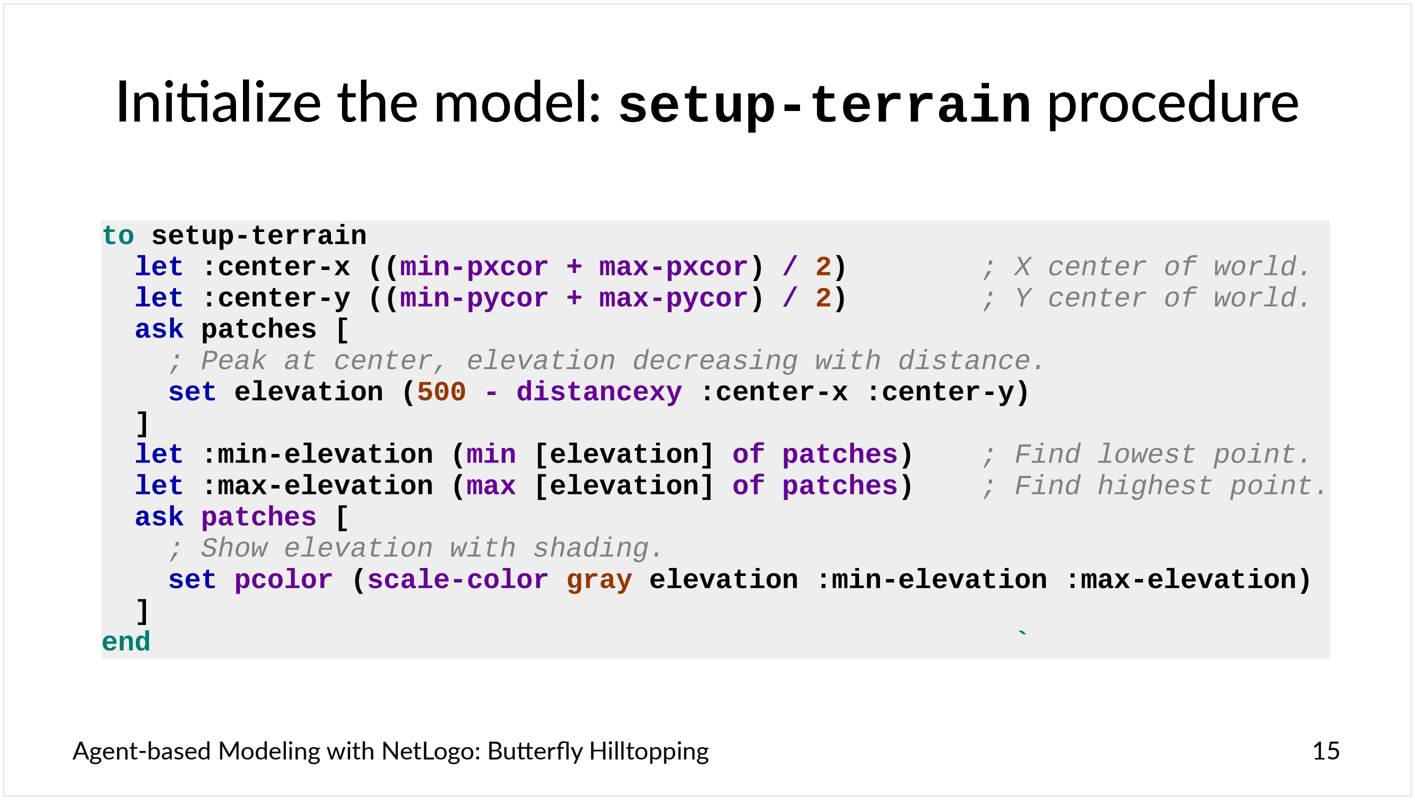
Maximum 1000

min, increment, and max may be numbers or reporters

Value 100

Units (optional)

□ vertical?



Initialize the model: **`setup-terrain`** procedure

```
to setup-terrain
  let :center-x ((min-pxcor + max-pxcor) / 2)        ; X center of world.
  let :center-y ((min-pycor + max-pycor) / 2)        ; Y center of world.
  ask patches [
    ; Peak at center, elevation decreasing with distance.
    set elevation (500 - distancexy :center-x :center-y)
  ]
  let :min-elevation (min [elevation] of patches)    ; Find lowest point.
  let :max-elevation (max [elevation] of patches)    ; Find highest point.
  ask patches [
    ; Show elevation with shading.
    set pcolor (scale-color gray elevation :min-elevation :max-elevation)
  ]
end                                                     `
```

- Note: All procedures must be located below the **`breed`** and **`patches-own`** blocks.

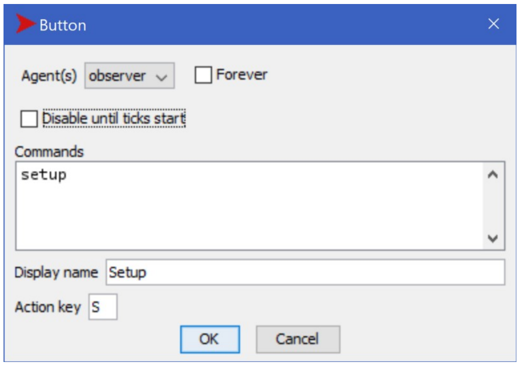
Initialize the model: `setup-butterflies` procedure

```
to setup-butterflies
  set-default-shape butterflies "butterfly" ; Breed shape.
  create-butterflies num-butterflies [
    setxy random-pxcor random-pycor   ; Random position.
    set size 10                       ; Make shape visible.
  ]
end
```

Initialize the model: **`setup`** procedure

```
to setup
  clear-all           ; Clear agents, plots, etc.
  setup-terrain       ; Set patch elevations.
  setup-butterflies ; Create & place butterflies.
  reset-ticks         ; Start simulation clock.
end
```

Start model from UI: **Setup** button



Agent(s) observer □ Forever

□ Disable until ticks start

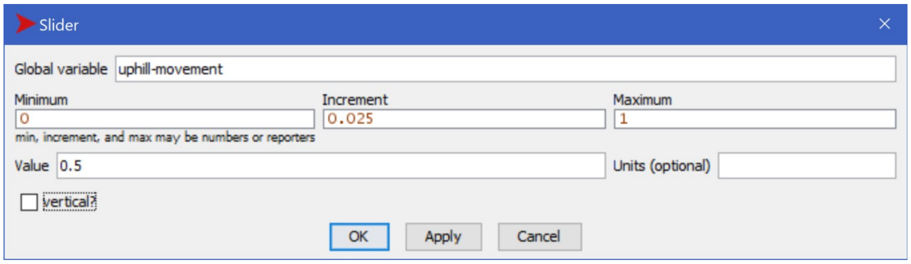
Commands

```
setup
```

Display name Setup

Action key S

Controlling probability of uphill movement: `uphill-movement` slider



Global variable uphill-movement

Minimum 0

Increment 0.025

Maximum 1

min, increment, and max may be numbers or reporters

Value 0.5

Units (optional)

☐ vertical?

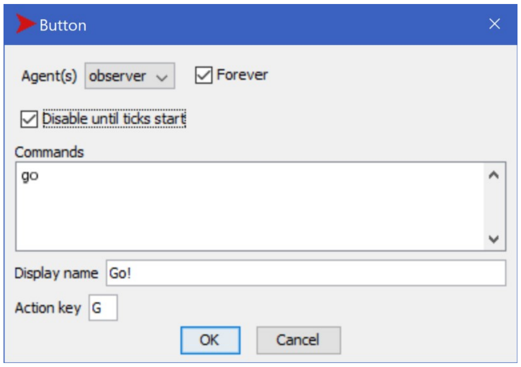
Agent behavior: **move** procedure

```
to move
  ; Flip a weighted coin to choose uphill/random move.
  ifelse ((random-float 1) < uphill-movement) [
    uphill elevation           ; Move to highest neighbor.
  ] [
    move-to (one-of neighbors) ; Move to random neighbor.
  ]
end
```

Model execution: **go** procedure

```
to go
  ask butterflies [
    move
  ]
  tick          ; Advance model clock.
end
```

Run model from UI: **Go** button



Agent(s) observer ☑ Forever

☑ Disable until ticks start

Commands

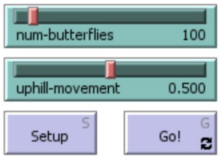
```
go
```

Display name Go!

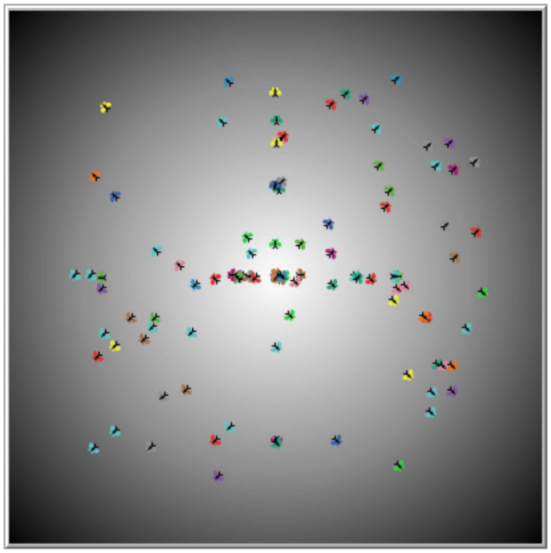
Action key G

Model: **`Butterfly Hilltopping (Phase 1).nlogo`**

Run the model!


num-butterflies 100

uphill-movement 0.500

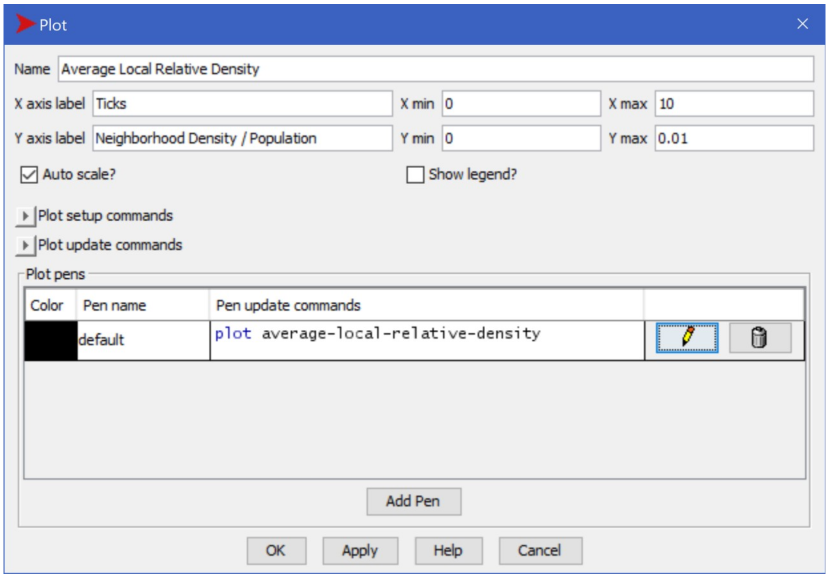


- Click **Setup** to initialize the model.
- Click **Go** button to run the model.
- Experiment with the slider values, initializing and running the model multiple times. What do you notice?

Measure butterfly grouping: **`average-local-relative-density`** procedure

```
to-report average-local-relative-density
  let :radius 5
  ; Avg. # local butterflies, divided by area & population.
  report
    ((mean [count butterflies in-radius :radius] of butterflies)
    / (count butterflies * :radius * :radius * pi))
end
```

Plot butterfly grouping measure in UI



Name: Average Local Relative Density

X axis label: Ticks — X min: 0 — X max: 10

Y axis label: Neighborhood Density / Population — Y min: 0 — Y max: 0.01

☑ Auto scale? ☐ Show legend?

▸ Plot setup commands

▸ Plot update commands

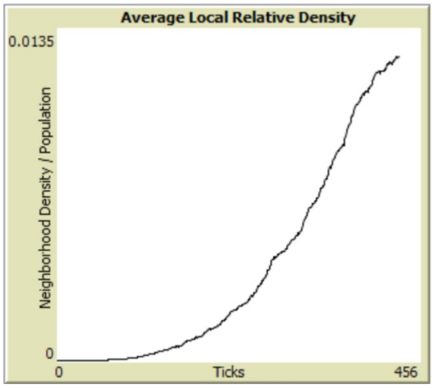
Plot pens

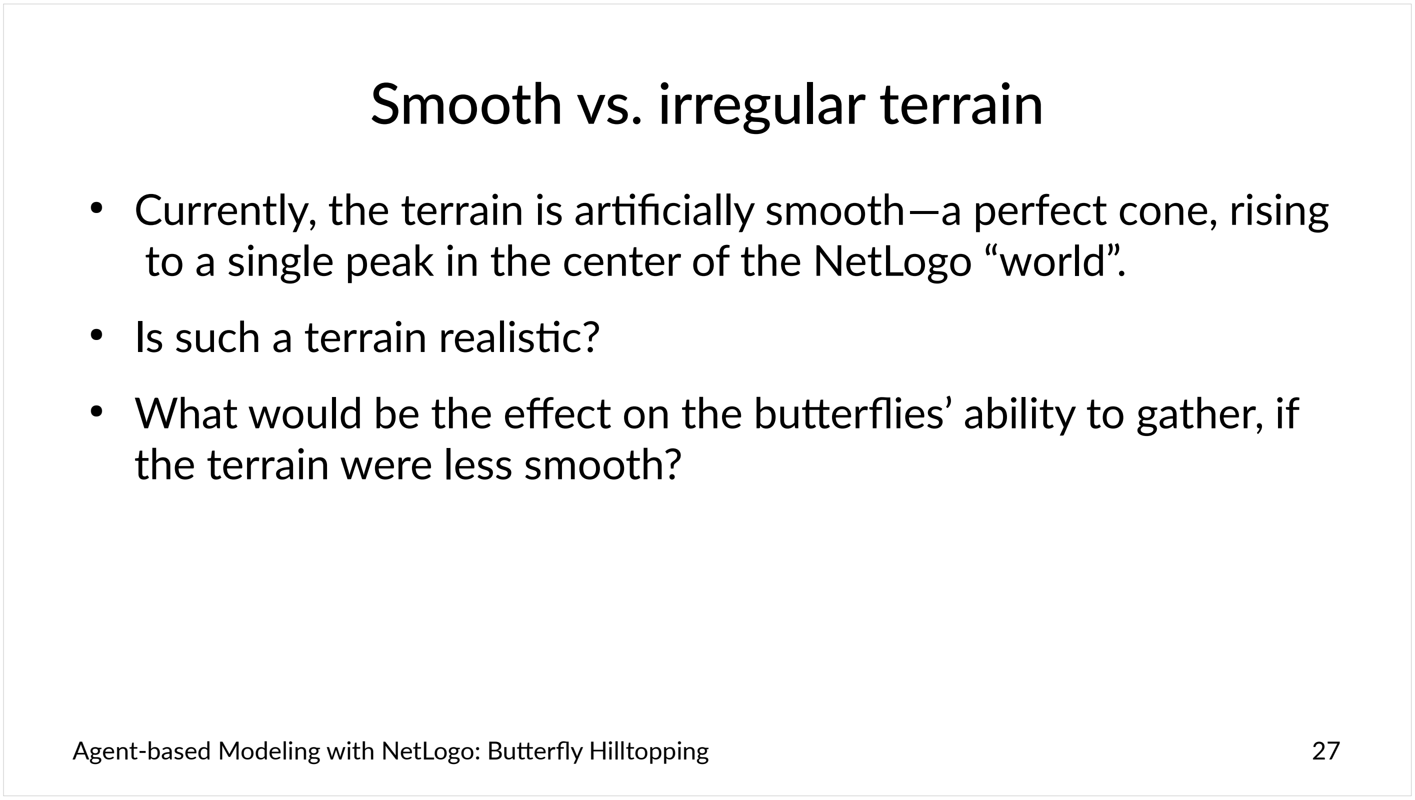
Color	Pen name	Pen update commands
■	default	`plot average-local-relative-density`

Model: **`Butterfly Hilltopping (Phase 2).nlogo`**

Run the model again!

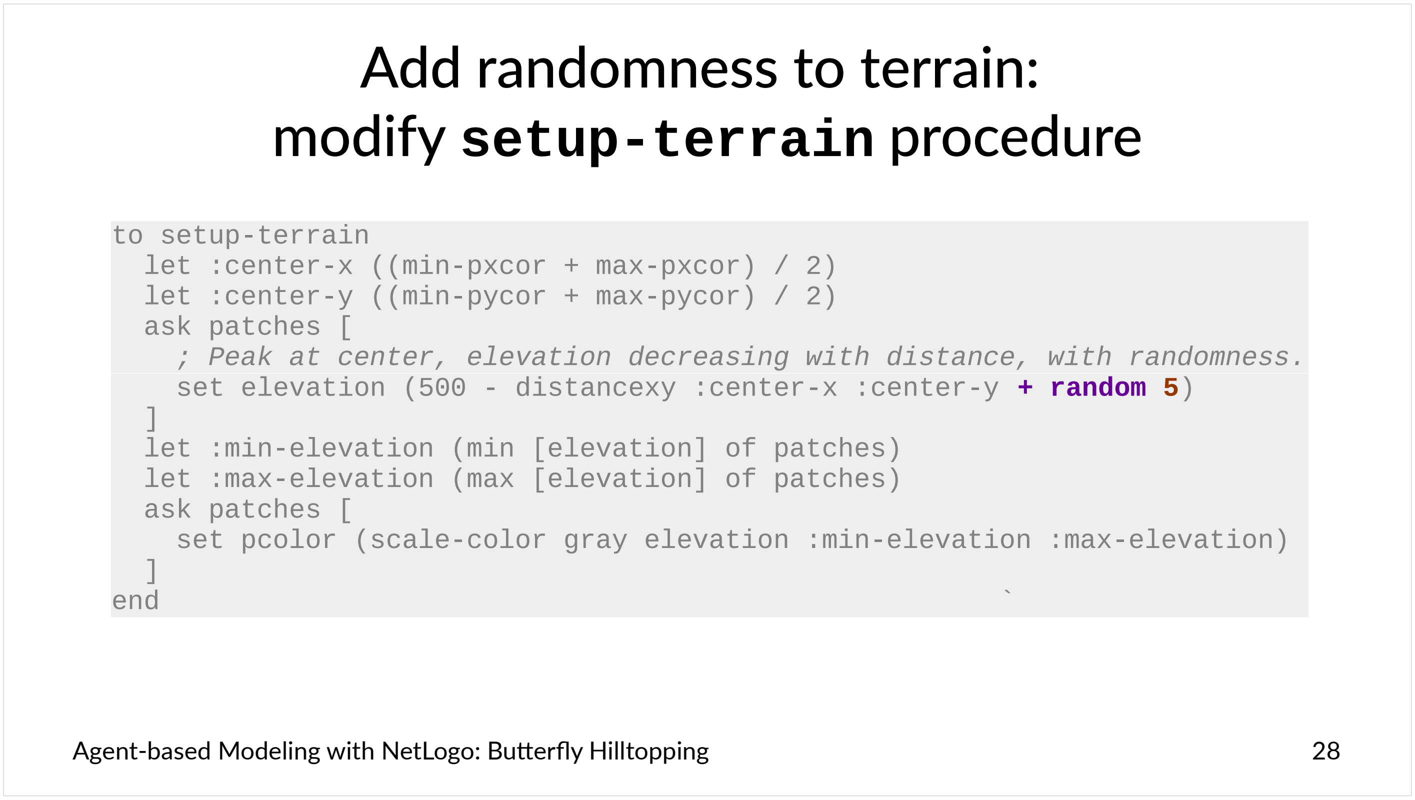
- Initialize and run the model multiple times with different values of **`uphill-movement`**. What do you observe in the **Average Local Relative Density plot**?





Smooth vs. irregular terrain

- Currently, the terrain is artificially smooth—a perfect cone, rising to a single peak in the center of the NetLogo “world”.
- Is such a terrain realistic?
- What would be the effect on the butterflies’ ability to gather, if the terrain were less smooth?

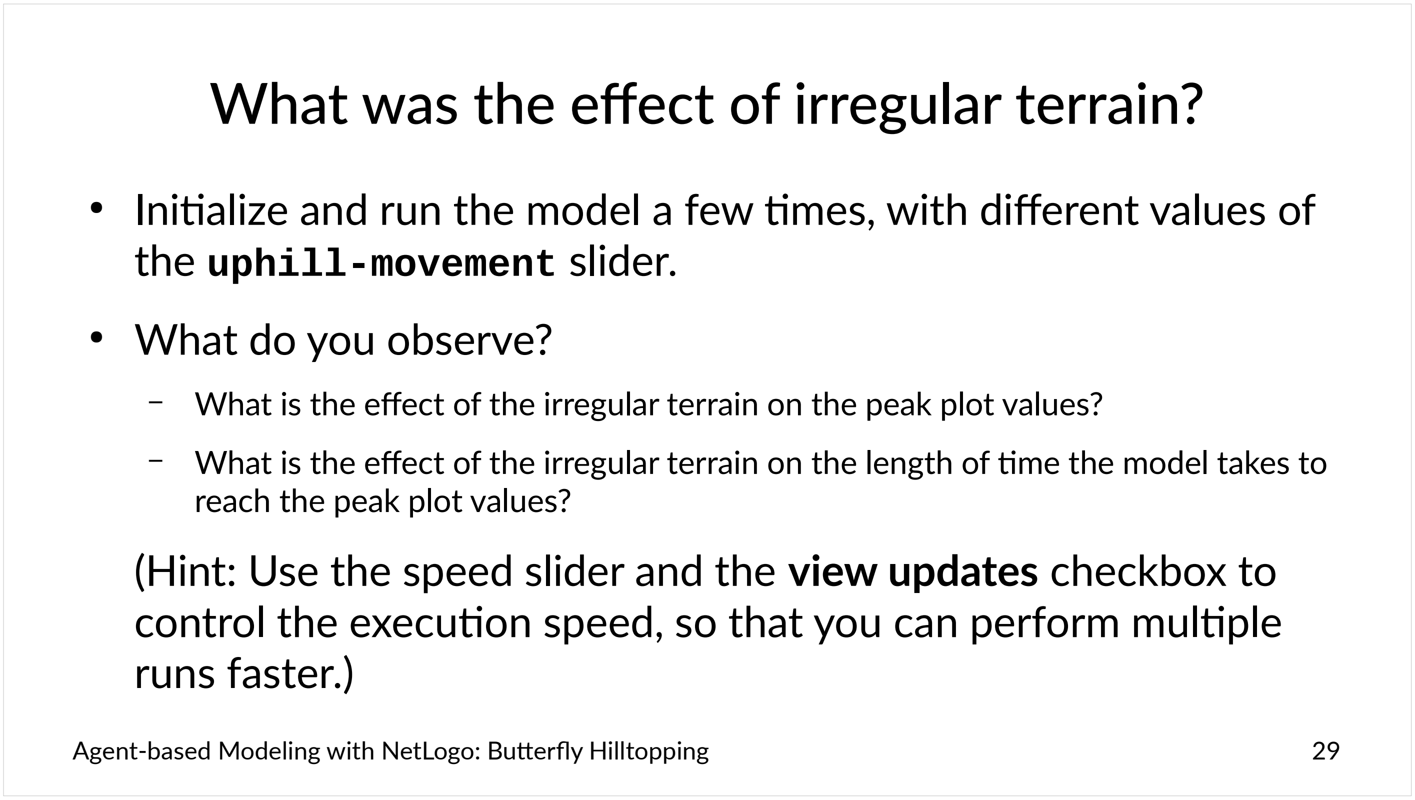


Add randomness to terrain: modify `setup-terrain` procedure

```
to setup-terrain
  let :center-x ((min-pxcor + max-pxcor) / 2)
  let :center-y ((min-pycor + max-pycor) / 2)
  ask patches [
    ; Peak at center, elevation decreasing with distance, with randomness.
    set elevation (500 - distancexy :center-x :center-y + random 5)
  ]
  let :min-elevation (min [elevation] of patches)
  let :max-elevation (max [elevation] of patches)
  ask patches [
    set pcolor (scale-color gray elevation :min-elevation :max-elevation)
  ]
end                                                    `
```

- Code shown includes code already written, in gray, normal weight typeface.
- New code is shown in bold, colored as in the NetLogo editor.

Model: **`Butterfly Hilltopping (Phase 3).nlogo`**

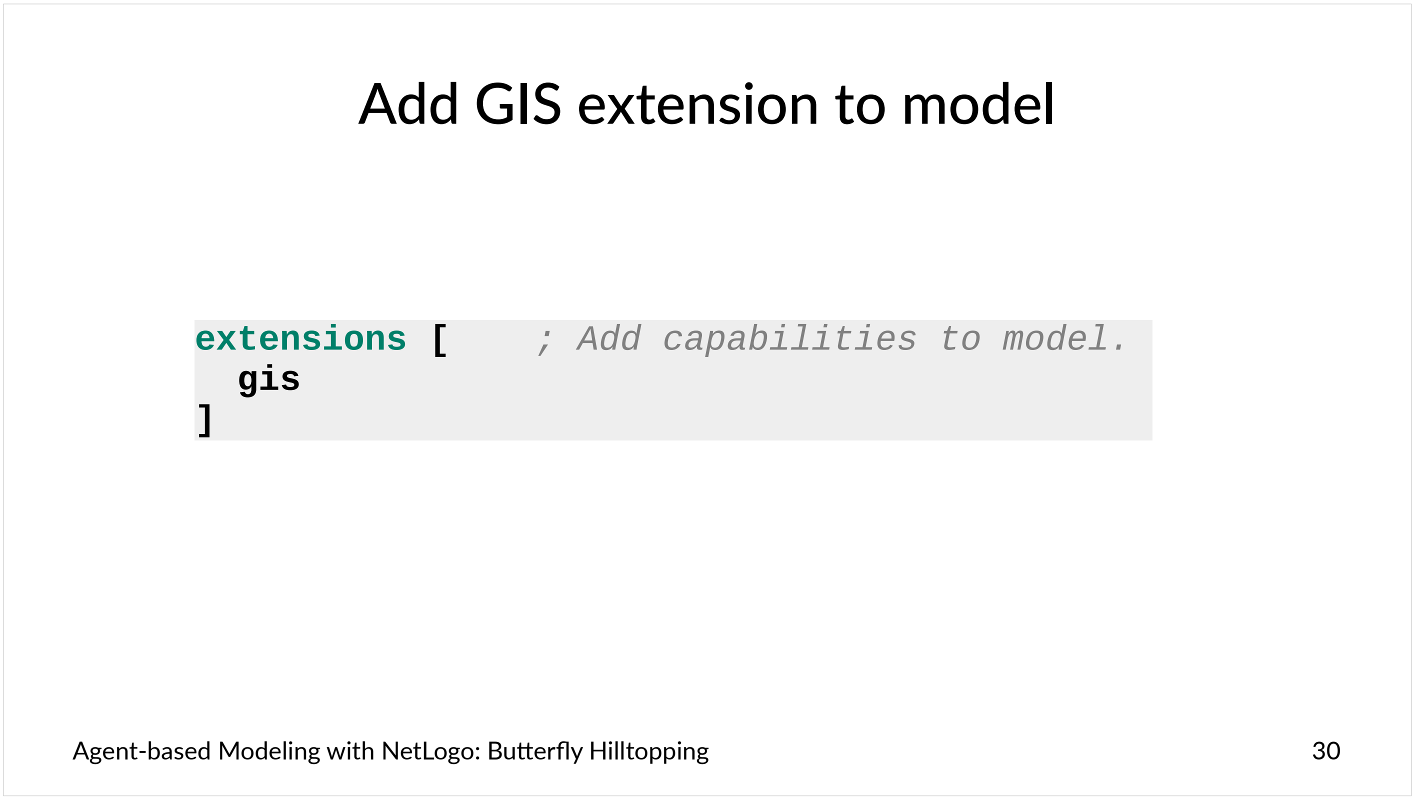


What was the effect of irregular terrain?

- Initialize and run the model a few times, with different values of the **`uphill-movement`** slider.
- What do you observe?
  - What is the effect of the irregular terrain on the peak plot values?
  - What is the effect of the irregular terrain on the length of time the model takes to reach the peak plot values?

(Hint: Use the speed slider and the **view updates** checkbox to control the execution speed, so that you can perform multiple runs faster.)

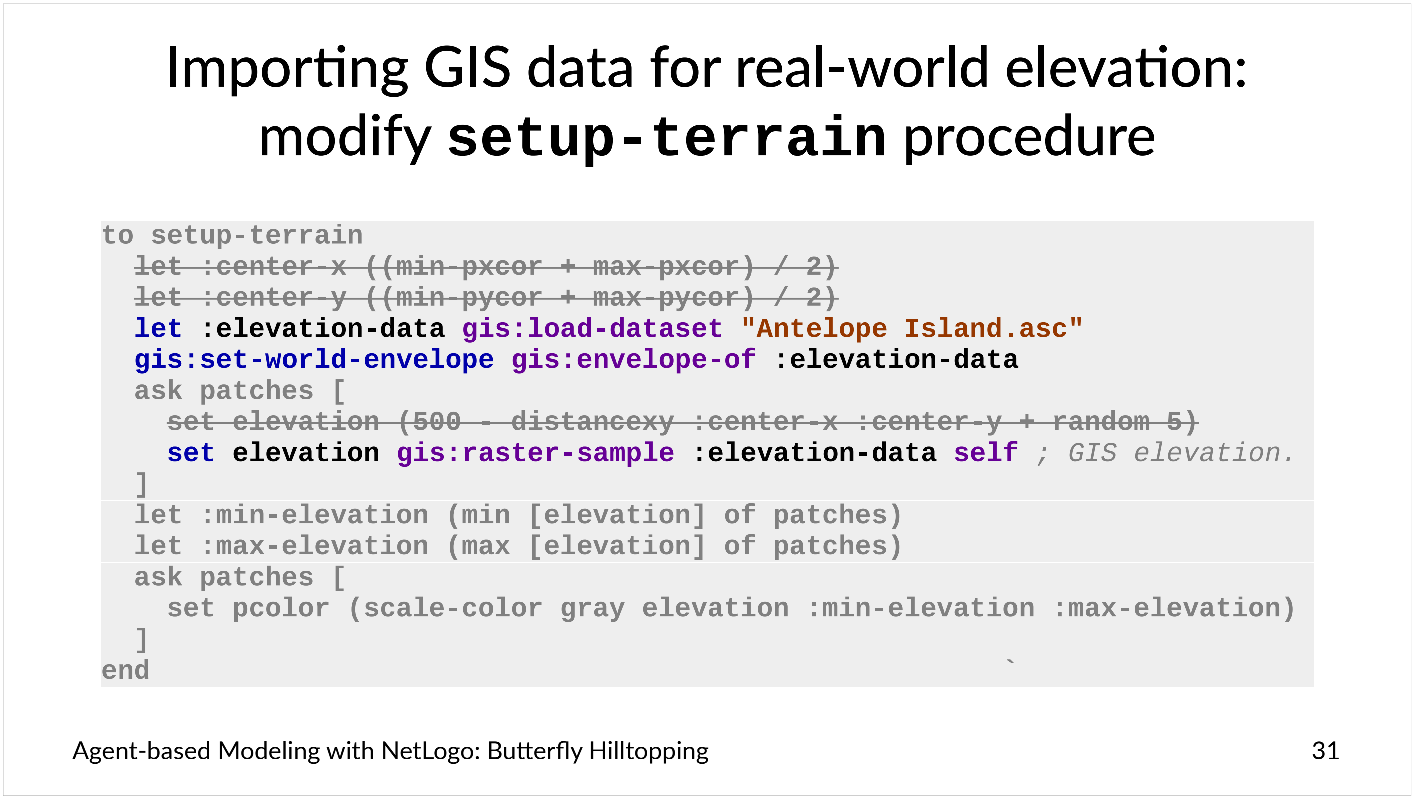
- 100% uphill movement (**`uphill-movement`** = 1.0) should result in butterflies climbing to the top of whatever local hill they happen to be closest to, and staying there. If the terrain is more than just a little bumpy, small groups of butterflies will be spread on little molehills all over.
- On the other hand, 100% random movement (**`uphill-movement`** = 0.0) will result in the butterflies never even attempting to climb hills.



Add GIS extension to model

```
extensions [    ; Add capabilities to model.
  gis
]
```

- Important: The `extensions` block must be at the very start of the model code (before `breed`, `globals`, procedures, etc.).

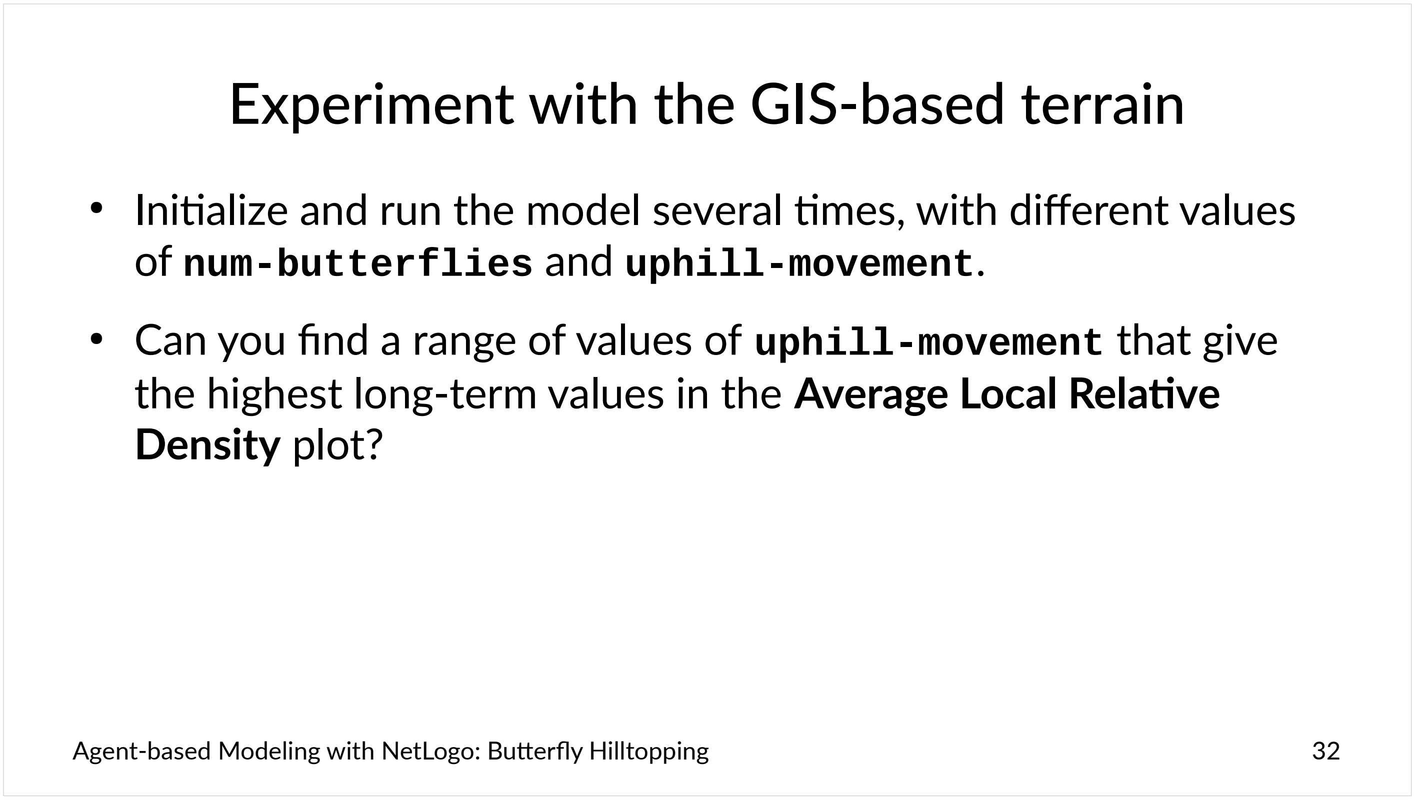


Importing GIS data for real-world elevation: modify **`setup-terrain`** procedure

```
to setup-terrain
  ~~let :center-x ((min-pxcor + max-pxcor) / 2)~~
  ~~let :center-y ((min-pycor + max-pycor) / 2)~~
  let :elevation-data gis:load-dataset "Antelope Island.asc"
  gis:set-world-envelope gis:envelope-of :elevation-data
  ask patches [
    ~~set elevation (500 - distancexy :center-x :center-y + random 5)~~
    set elevation gis:raster-sample :elevation-data self ; GIS elevation.
  ]
  let :min-elevation (min [elevation] of patches)
  let :max-elevation (max [elevation] of patches)
  ask patches [
    set pcolor (scale-color gray elevation :min-elevation :max-elevation)
  ]
end
```

- **`gis:load-dataset`** requires specification of a file location relative to the directory location of the NetLogo model file. If the GIS data files are in the same directory, this is simple; otherwise, the necessary directory components must be included in the input to **`gis:load-dataset`**.

Model: **`Butterfly Hilltopping (Phase 4).nlogo`**



Experiment with the GIS-based terrain

- Initialize and run the model several times, with different values of **`num-butterflies`** and **`uphill-movement`**.
- Can you find a range of values of **`uphill-movement`** that give the highest long-term values in the **Average Local Relative Density** plot?

- What tools/techniques would be useful for answering the question?
  - BehaviorSpace
  - BehaviorSearch
  - …
- What does “long-term” mean?
  - Forever
  - Some maximum tick value
    - How long is a tick (i.e. what real-world time interval does a model tick correspond to)? Does this affect the answer?