Chapter 4 Exercise Hints and Solutions

Agent-based and Individual-Based Modeling: A Practical Introduction, 2nd Edition

Exercise 1

Students should predict that when *q* is zero, the butterflies disperse in random directions instead of aggregating on a hilltop; and when *q* is one, they all follow the same paths directly to the nearest hilltop. These predictions should be correct.

Exercise 2

In the setup procedure, change this:

```
; Create butterflies
crt 500
[
  set size 2
  ; Set initial location to random patch
  setxy random-pxcor random-pycor
  pen-down
]
```

to something like this:

```
; Create butterflies
crt 500
[
  set size 2
  ; Set initial location to a patch near the center,
  ; which is at 74, 74
  setxy (70 + random 9) (70 + random 9)
  pen-down
]
```

Other potential solutions are:

```
; Create butterflies
crt 500
[
  set size 2
  ; Set initial location to a patch near the center,
  ; which is at 74, 74
  move-to one-of patches with [(distance patch 74 74) <= 10]
  pen-down
]
```

and:

```
; Create butterflies
crt 500
[
  set size 2
```

```
    ; Set initial location to a patch near the center,
    ; which is at 74, 74
    move-to one-of patches with
    [
      pxcor >= 64 and pxcor <= 83 and
      pycor >= 64 and pycor <= 83
    ]
    pen-down
  ]
```

Exercise 3

It should be clear that butterflies tend to move in directions that are multiples of 45 degrees. To understand why, it helps to see the exact elevations of the patches.

You can see patch elevations by setting their label to their elevation, via this statement (which can be added to the `setup` procedure, or just entered in the Command Center):

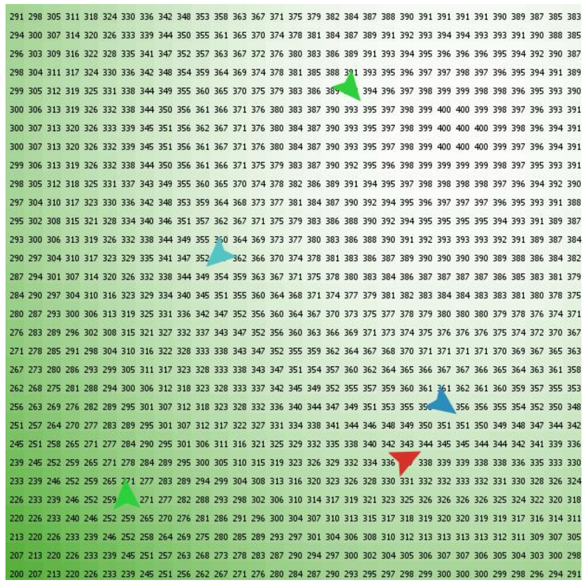
```
  ask patches [set plabel elevation]
```

But if you do this, the View becomes indecipherable because the patch labels occupy all the space. It helps to eliminate the decimal places in the elevation, and turn the labels black. Use this in `setup`:

```
  ; Assign an elevation to patches and color them by it
  ask patches
  [
    ; Elevation is a sine function of X, Y coordinates
    ; with maximum elevation of 400 when sin is 1.0
    set elevation 200 + (100 * (sin (pxcor * 3.8) + sin (pycor * 3.8)))

    set pcolor scale-color green elevation 0 400
    set plabel precision elevation 0
    set plabel-color black
  ] ; end of "ask patches"
```

It also helps to make the world small so it includes only one (or part of one) hill. For example, set the world so that `max-pxcor` and `max-pycor` are 30 and the patch size is about 20. Now the View should look like this:



If you look at any patch, you see that the highest neighbor elevation is always either (a) at a 45-degree diagonal, or (b) directly left or right, or up or down. This regularity simply results from the geometry of the sinusoidal hill, which is symmetric.

Exercise 4

Adding random variation to the elevation breaks up the regularity identified in the solution to Exercise 3: now the highest neighbor is not always at a 45- or 90-degree heading. However, this effect is not clear unless the amount of randomness is relatively large compared to the differences in elevation among neighboring cells (as illustrated in the above figure), e.g., by adding a random number between zero and 50 to each patch's elevation.

A NetLogo code with this "noise" in the landscape is available with the instructor materials; it is called "ButterflyModel-NoiseInElevation-2ndEd.nlogo".