# HW 1 Problems

*Note: These exercises presuppose that you have the Game of Life Model, as shown in Wilensky and Rand Chapter 2, pgs. 45-68. If you have not programmed the model yourself, you should download a copy from canvas.*

## Basic

### 1

Write a new setup procedure called make-glider that clears the world and makes a glider with its top-left corner at (0,0) which moves down and to the right (see pg. 64).

Here is a stub for you to work with:

to make-glider

ca

[\*\* Your code here. \*\*]

reset-ticks

end

**Answer:**

to make-glider

ca

ask patches [

set pcolor blue - 3]

ask patch 0 0 [set pcolor green]

ask patch 1 -1 [set pcolor green]

ask patch 2 -1 [set pcolor green]

ask patch 0 -2 [set pcolor green]

ask patch 1 -2 [set pcolor green]

]

reset-ticks

end

### 2

Find a still-life configuration with five live cells and write a new procedure called make-still which clears the world and creates the still configuration at (0, 0).

Here is a stub:

to make-still

ca

[\*\* Your code here. \*\*]

reset-ticks

end

**Answer:**

to make-still

ca

ask patches [

set pcolor blue - 3]

ask patch 0 0 [set pcolor green]

ask patch 0 -1 [set pcolor green]

ask patch 1 0 [set pcolor green]

ask patch 2 -1 [set pcolor green]

ask patch 1 -2 [set pcolor green]

**]**

### 3

Find a period two oscillator different from the blinker and write a new procedure call make-oscillator which creates it at the origin.

Here is a stub:

to make-oscillator

ca

[\*\* Your code here. \*\*]

reset-ticks

end

**Answer:**

to make-oscillator

ca ask patches [

set pcolor blue - 3

ask patch 0 0 [set pcolor green]

ask patch 1 0 [set pcolor green]

ask patch 2 0 [set pcolor green]

ask patch 1 -1 [set pcolor green]

ask patch 2 -1 [set pcolor green]

ask patch 3 -1 [set pcolor green]

]

reset-ticks

end

### 5

Make a new slider on the interface with range from 0 to 100 in increments of 1 which is called spawn-percentage. Change the setup method so that, on average, spawn-percent percent of the patches begin alive at setup.

## Beyond the Basics

### 1

Write a function which creates the glider gun shown on page 65 with top-left corner at (-25, 25).

### 2

Create four sliders, min-spawn, max-spawn, min-die, and max-die, which scale between 0 and 8. Write a new go function on the model of the basic one, except replacing the hard-coded numbers with slider values.

### 3

Write a function which implements an altered set of transition rules. If a patch is set to die in the old rules, instead of changing its pcolor to blue, change it to yellow. All yellow patches should die at the beginning of each tick.

## Discussion Questions

### 1

Consider the discussion of the Game of Life on pages 65-68, in which Wilensky and Rand discuss real-world applications of the game. What other kinds of practical applications can you imagine the game of life could be applied to? Can you think of slight modifications of the rules which would make it more closely conform to specific real-world scenarios?

**Possible Answer:**

### 2

**A.** Consider the three types of objects in pages 60-63. Such objects, often call *emergent phenomena,* appear in the model even though they are not hard coded in. So, even though the model only expresses a grid of cells which can be either alive or dead, we can observe objects moving around, rather than merely cells blinking green and blue. Hypothesize what it is about the rules of this model that allow for such cohesive-appearing “objects” to emerge.

**Possible Answer:**

**B.** Observe some runs of the Game of Life model and discuss whether you can think of any other types of objects or emergent phenomena. Try to define any new types as clearly as possible.

**Possible Answer:**