```
public void initialize() {
   initCells(control.getInt("grid size"));
}

private int calcNeighborsPeriodic(int row, int col) {
   // do not count self
   int neighbors = -latticeFrame.getValue(row, col);
   // add the size so that the mod operator works for row = 0 and
   // col = 0
   row += size;
   col += size;
   for(int i = -1;i<=1;i++) {
      for(int j = -1;j<=1;j++) {
         neighbors +=
               latticeFrame.getValue((row+i)%size, (col+j)%size);
      }
   }
   return neighbors;
}

public void doStep() {
   for(int i = 0;i<size;i++) {
      for(int j = 0;j<size;j++) {
         newCells[i][j] = 0;
      }
   }
   for(int i = 0;i<size;i++) {
      for(int j = 0;j<size;j++) {
         switch(calcNeighborsPeriodic(i, j)) {
         case 0:
         case 1:
            newCells[i][j] = 0; // dies
            break;
         case 2:
            // life goes on
            newCells[i][j] = (byte) latticeFrame.getValue(i, j);
            break;
         case 3:
            newCells[i][j] = 1; // birth
            break;
         default:
            newCells[i][j] = 0; // dies of overcrowding if > 3
         }
      }
   }
   latticeFrame.setAll(newCells);
}

public static void main(String[] args) {
   OSPControl control = SimulationControl.createApp(new LifeApp());
   control.addButton("clear", "Clear"); // optional custom action
}
}
```



**Problem 14.6 The Game of Life**

(a) `LifeApp` allows the user to determine the initial configuration interactively by clicking on a cell to change its value before hitting the Start button. Choose several initial configurations with a small number of live cells and determine the different types of patterns that emerge. Some suggested initial configurations are shown in Figure 14.2b. Does it matter whether you use fixed or periodic boundary conditions? Use a $16 \times 16$ lattice.

(b) Modify `LifeApp` so that each cell is initially alive with a 50% probability. Use a $32 \times 32$ lattice. What types of patterns typically result after a long time? What happens for 20% live cells? What happens for 70% live cells?

(c) Assume that each cell is initially alive with probability $p$. Given that the density of live cells at time $t$ is $\rho(t)$, what is $\rho(t+1)$, the expected density at time $t+1$? Do the simulation and plot $\rho(t+1)$ versus $\rho(t)$. If $p = 0.5$, what is the steady-state density of live cells?

*(d) `LifeApp` has not been optimized for the Game of Life and is written so that other rules can be implemented easily. Rewrite `LifeApp` so that it uses bit manipulation (see Section 14.6). ■

The Game of Life is an example of a universal computing machine. That is, we can choose an initial configuration of live cells to represent any possible program and any set of input data, run the Game of Life, and the output data will appear in some region of the lattice. The proof of this result (see Berlekamp et al.) involves showing how various configurations of cells represent the components of a computer, including wires, storage, and the fundamental components of a CPU—the digital logic gates that perform *and*, *or*, and other logical and arithmetic operations. Other cellular automata can also be shown to be universal computing machines.

## 14.2 ■ SELF-ORGANIZED CRITICAL PHENOMENA

Very large events such as a magnitude eight earthquake, an avalanche on a snow covered mountain, the sudden collapse of an empire (for example, the Soviet Union), or the crash of the stock market are rare. When such events occur, are they due to some special set of circumstances or are they part of a more general pattern of events that would occur without any specific external intervention? The idea of *self-organized criticality* is that in many cases the occurrence of a very large event does not depend on special conditions or external forces and is due to the intrinsic dynamics of the system.

If $s$ represents the magnitude of an event, such as the energy released in an earthquake or the amount of snow in an avalanche, then a system is said to be *critical* if the number of events $N(s)$ follows a power law:

$$N(s) \sim s^{-\alpha} \quad \text{(no characteristic scale).} \tag{14.1}$$

If $\alpha \approx 1$, the form (14.1) implies that there would be one large event of size 1000 for every 1000 events of size one. One implication of the power law form (14.1) is that there is no