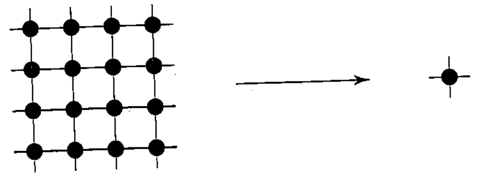

Figure 12.11 An example of a $b = 4$ cell used on the square lattice. The cell contains b^2 sites which are rescaled to a single supersite or cell after a renormalization group transformation.

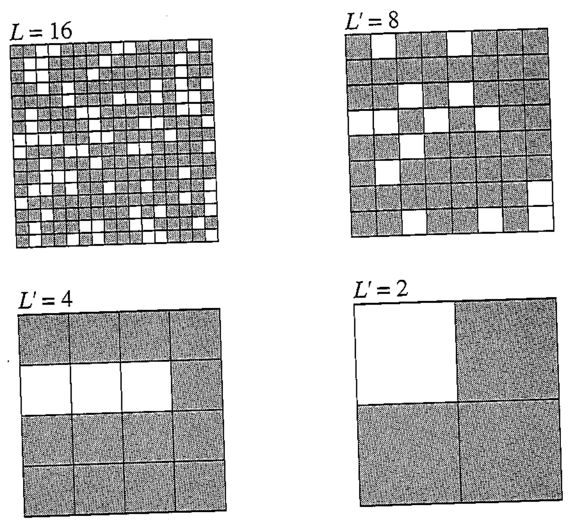

Figure 12.12 A percolation configuration generated at $p = 0.7$. The original configuration has been renormalized three times by transforming cells of four sites into one new supersite. What would be the effect of an additional transformation?

Because we want to preserve the main features of the original lattice and hence its connectedness (and its symmetry), we assume that a renormalized site is occupied if the original group of sites spans the cell. For simplicity, we adopt the vertical spanning criterion. The effect of performing a renormalization transformation on typical percolation configurations for p above and below p_c is illustrated in Figures 12.12 and 12.13, respectively. In both cases the effect of the successive transformations is to move the system away from p_c. We see that for $p = 0.7$, the effect of the transformations is to drive the system toward $p = 1$. For $p = 0.5$, the trend is to drive the system toward $p = 0$. Because we began with a finite lattice, we cannot continue the renormalization transformation indefinitely.

The class RGApp implements a visual interpretation of the renormalization group. This class creates four windows with the original lattice in the first window and three renormalized lattices in the other three windows.

Listing 12.4 The visual renormalization group.

```
package org.opensourcephysics.sip.ch12;
import org.opensourcephysics.controls.*;
import org.opensourcephysics.frames.*;
import java.awt.Color;
```

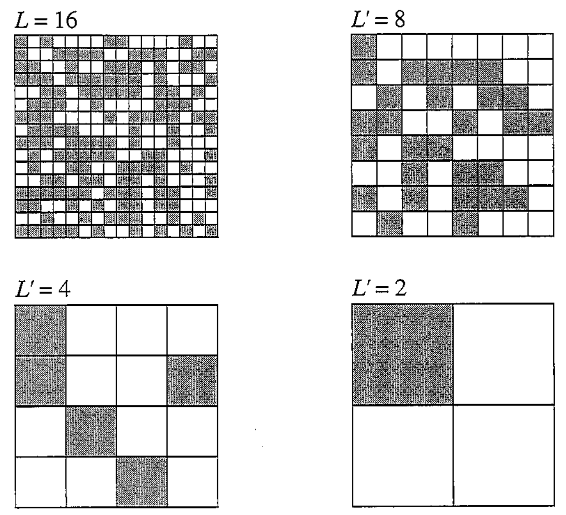

Figure 12.13 A percolation configuration generated at $p = 0.5$ (shaded cells are occupied). The original configuration has been renormalized three times by transforming blocks of four sites into one new site and rescaling all lengths by a factor of $b = 2$. What would be the effect of an additional transformation?

```
public class RGApp extends AbstractCalculation {
   LatticeFrame originalLattice = new LatticeFrame("Original Lattice");
   LatticeFrame block1 = new LatticeFrame("First Blocked Lattice");
   LatticeFrame block2 = new LatticeFrame("Second Blocked Lattice");
   LatticeFrame block3 = new LatticeFrame("Third Blocked Lattice");

   public RGApp() {
      setLatticeColors(originalLattice);
      setLatticeColors(block1);
      setLatticeColors(block2);
      setLatticeColors(block3);
   }

   public void calculate() {
      int L = control.getInt("L");
      double p = control.getDouble("p");
      newLattice(L, p, originalLattice);
      block(originalLattice, block1, L/2); // block original lattice
      block(block1, block2, L/4);          // next blocking
      block(block2, block3, L/8);          // final blocking
      originalLattice.setVisible(true);
      block1.setVisible(true);
      block2.setVisible(true);
      block3.setVisible(true);
   }

   public void reset() {
      control.setValue("L", 64);
      control.setValue("p", 0.6);
   }
```