```
/**
 *Elliot Duncan
 *Horton 7th
 *5/11/24
 *@(#)SeedPanel.java
 * Creates a JPanel to display a single seed, without motion.
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Graphics;
import javax.swing.JPanel;
class SeedPanel extends JPanel {
  private Seed seed;
  private int preferredSize;
  private int PIXEL SIZE;
  /**
   * Takes a panel size, and a seed, and defines a new JPanel of the specified
  * size displaying the seed.
   * @param s the seed
   * @param size size of panel
   */
  public SeedPanel(Seed s, int size) {
    setBackground(new Color(20, 20, 20));
    seed = s;
    int seedSize = Math.max(seed.getSizeX(), seed.getSizeY());
    PIXEL_SIZE = Math.min(100, size / seedSize);
    preferredSize = (seedSize * PIXEL_SIZE) + 100;
   PIXEL_SIZE = Math.max(PIXEL_SIZE, 1);
  }
   * returns the seed that is on the panel
  * @return the seed
  public Seed getSeed() { return seed; }
   *Returns the dimension representing the preferred size of the window.
   *@return the preferred size
   */
  @Override
  public Dimension getPreferredSize() {
    return new Dimension(preferredSize, preferredSize);
  }
  /**
   *paints the seed to the window
   *@param g the current graphics context
```

```
*/
  @Override
  public void paintComponent(Graphics g) {
    super.paintComponent(g);
    g.setColor(new Color(200, 200, 200));
    g.drawString("Press ENTER to load this seed.\n", 250, 15);
    g.drawString(seed.getName(), 10, 15);
    g.drawString(String.format("Contains %d cells\n", seed.getNumOfCells()), 10,
    g.drawString(
        String.format("%d pixels wide, %d pixels tall\n, area of %d pixels",
                      seed.getSizeX(), seed.getSizeY(),
                      seed.getSizeX() * seed.getSizeY()),
        10, 30);
    drawSeed(g);
  private void drawSeed(Graphics g) {
    int[][] cells = seed.getCells();
    int seedSizeX = cells[0].length;
    int seedSizeY = cells.length;
    for (int i = 0; i < seedSizeY; i++) {</pre>
      for (int j = 0; j < seedSizeX; j++) {
        if (cells[i][j] > 0)
          g.fillRect(25 + PIXEL_SIZE * j, 55 + PIXEL_SIZE * i, PIXEL_SIZE - 0,
                     PIXEL_SIZE - 0);
     }
   }
 }
}
```