```java
// FroggerComponent.java
/*
  Program describtion: The main class for Frogger.
      Stores the state of the world, draws it,
      handles the tick() and key() logic.
*/

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.io.*;
import javax.imageio.*;

public class FroggerComponent extends JComponent  {
   // Size of the game grid
   public static final int WIDTH = 20;
   public static final int HEIGHT = 7;

   // Initial pixel size for each grid square
   public static final int PIXELS = 50;

   // Image filenames for car, lily, and frog
   public static final String[] IMAGES = new String[] {
                     "frog.png", "car.png", "lily.png"};

   // Colors for ROAD, WATER, and DIRT
   public static final Color[] COLORS = new Color[] {
                   Color.BLACK, Color.BLUE, Color.GRAY };

   // Codes to store what is in each square in the grid
   public static final int EMPTY = 0;
   public static final int CAR = 1;
   public static final int LILY = 2;

   private int [][] grid =new int [WIDTH][HEIGHT];
   Row [] rows =  new Row [7];
   private int x = 0;
   private int y = 0;
   private boolean dead;
```

```java
/*
  Provided utility method to read in an Image object.
  If the image cannot load,
  prints error output and returns null.
   Uses Java standard ImageIO.read() method.
*/
private Image readImage(String filename)
{
   Image image = null;
   try
   {
      image = ImageIO.read(new File(filename));
   }
   catch (IOException e)
   {
      System.out.println("Failed to load image '" +
                         filename + "'");
      e.printStackTrace();
   }
   return(image);
}


private void readRow(String file)
{
   try
   {
      FileReader fr = new FileReader(file);
      BufferedReader br = new BufferedReader(fr);
      String line = br.readLine();
      int count = 1;

      while ((line!=null) && (line!="\n") && (line!="\r"))
      {
         Row r = new Row(line);
         rows [count] = r;
         line = br.readLine();
         count++;
      }
   }
   catch (IOException ex)
   {
      System.out.println("File not found!");
   }
}
```

```
   public FroggerComponent(String filename)
   {
      setPreferredSize(new Dimension(WIDTH * PIXELS,
                                     HEIGHT * PIXELS) );
      readRow(filename);
      // readImage(...); ...
   }

   public void reset()
   {
      for (int x = 0; x < grid.length; x++)
         for (int y= 0; y < grid[x].length; y++)
            grid[x][y] = EMPTY;

      dead = false;
      x = 0;
      y = 6;

      repaint();
   }

   private void moveBy(int dx, int dy)
   {
      if (   (x + dx >= 0 && x + dx < WIDTH)
          && (y + dy >= 0 && y + dy < HEIGHT) )
      {
         x += dx;
         y += dy;
      }
   }

   public boolean isWin()
   {
      return (y == 0);
   }

   public void paintComponent(Graphics g)
   {   }

   public void key(int code)
                   throws ArrayIndexOutOfBoundsException
   {   }

   public void tick(int round)
   {   }
}
```