James Stephenson

**Introduction**

In order to display images in the JLabel instead of numbers, I made the following changes to the *SliderDemo* class:

1. Five new classes are imported:
   1. *File* represents a file in the images directory.
   2. *MimetypesFileTypeMap* enables detection of the file’s MIME type.
   3. *IOException* is caught if it occurs while reading a file.
   4. *ImageIO* enables a file to be read into a *BufferedImage* array.
   5. *BufferedImage* represents a loaded image in the array.
2. The *FPS\_MAX* and *FPS\_INIT* static final variables are modified to reflect the DQ parameters.
3. The JLabel attribute is renamed as *imageLabel*. In addition, an array of *BufferedImages* is declared as an attribute.
4. An *addComponents* method is declared in order to refactor the constructor so it is smaller.
5. A *loadImages* method is introduced in order to load all the images inside the ‘images’ directory.
6. The update method is renamed *updateImage*, and now accesses the *BufferedImages* array to set the JLabel’s icon. All references to the old method name are corrected.
7. In the constructor, I changed the parameters to the *setMajorTickSpacing* and *setMinorTickSpacing* methods accordingly. The *loadImages* and *updateImage* methods are called.

The solution is intended to model a ‘Top 10’ chart in the UK. It depends on the ability to use the number selected on the *JSlider* as an index to the *BufferedImages* array. This only works in Java if you decrement that index by 1.