**Exercises: Simple Graphics**

**1.**Make a popup window that draws an empty tic-tac-toe

board. I.e. something that draws four lines and looks like the

picture on the left. Assume that the JPanel is exactly 300x300

(make the JFrame 320x340 so that about 300x300 is left after

accounting for the title bar and borders) Also, don’t worry

about making the lines more than 1 pixel thick. Setting an

exact size for the JPanel and making thick lines can both be

done in Java, of course, but we didn’t cover how.

**2.** Make another version of the TTT application, but this time the drawing code should*not* know the

panel size ahead of time, but instead use the panel’s getWidth and getHeight methods to compute

the size before drawing. Interactively stretch the frame to be different sizes, and be sure your draw-

ing changes accordingly. Feel free to assume that the width and height are the same as each other

(i.e., the panel is roughly square), but if you feel inspired, think about how to make the TTT board

look square even when the panel is not.

**3.** Make a red window that draws 50 small yellow circles at random locations within the window. The

drawing code should not know the panel size ahead of time. Interactively stretch the frame to be

different sizes, and be sure your drawing changes accordingly.

**4.** Copy the Bill Gates image from my simple-graphics-exercises project, or go to

http://images.google.com/, search for “Bill Gates”, and save one of the images to your project

folder. Make a popup window that draws the image. Remember that the image goes in a location

relative to the root of the*compiled* code. So, if you specified “Use project folder as root for sources

and class files” when you created the project, the folder containing the image will go in the top

level of the Eclipse project. If, on the other hand, you specified “Create separate folders for sources

and class files”, then the folder containing the image will go in the top level of the bin (not src)

folder.

Having the frame automatically fit the image size can be done by putting the image into an Image-

Icon, putting the ImageIcon into the JPanel, and calling frame.pack(), but none of those things were

covered in the lecture. So, just play with the frame width and height to make the image fit.

**5.** Draw a mustache or some other graffiti on top of Bill.