OOP Class
DrawSquare

Purpose: In this exercise you will use the PostScript single-operator functions to draw in a view. You will become familiar with the **frame** and **bounds** concepts of the View class and the **drawSelf::** method.

1. First, to get an idea of the kind of application you will build, run **Line.app** in the **Examples** directory. Study the way it works and study the code in the LineView class definition. (The files were derived from an example at the NeXT Developer Camp in the past.)

2. Start a new project called **DrawSquare** in the **DrawSquare** directory.

3. Create the **SquareView** class. To make things easy, copy *LineView.h* and *LineView.m* into the new project directory. Change their names to *SquareView.(hm)*. (Both the file names and the class names in the files.) Now add them to the project by selecting **Classes** in the PB file window, double-clicking the suitcase, and choosing *SquareView.m*. Save the project.

4. Modify the **drawSelf::** method such that a square is drawn in the middle of the view, and the size of the square varies with the position of the slider on the interface. The size of the square should vary from a small dot to the full size of the view. Use the **bounds** instance variable to compute where to draw. Draw with a line width of 5.0.

5. Open *DrawSquare.nib* in IB. Put a slider along the bottom of the window and a CustomView filling the rest of the window so that it looks the same as it did in **Line.app**.

6. Now **Parse** the SquareView class so that IB knows about it. Select the CustomView and make it a SquareView by choosing SquareView in the CustomView inspector.

7. Connect the slider to the SquareView.

8. Save the nib file. Return to PB, build and run the application.

Extension: The technique you used above was to draw the square directly using the bounds instance variable. Another way is to translate and scale the view so that you draw in a more natural set of coordinates. Modify your program to use View's **setDrawOrigin::** and **setDrawSize::** methods so that the view now has its (0,0) point at the center, and the width and height of the view are both scaled to 1.0. Since the view is not really square on the screen, you will distort the drawing and the square will become a rectangle!