HRGL: A Simple Programmatic Graphics Language

Chris Willmore, Lee Ballard, and Brett Daniel

Most vector graphics languages in use today fall in three distinct groups. The first contains low-level file formats such as Adobe Illustrator's binary file format, Macromedia's various Flash-based formats, the ISO standard Computer Graphics Metafile (CGM) used in technical illustration, and Windows Metafile. These powerful vector formats are almost always written by machine and are invisible to the average user.

The second group contains higher-level, human-readable languages used to specify graphics. These languages can be compiled down to a binary raster or vector format or interpreted to create drawings in a viewer. Examples include Adobe PostScript, the W3C's Scalar Vector Graphics markup language (SVG), and uncompiled CGM. Postscript is notable because it contains conditional and looping structures, making it a Turing-complete language. In contrast, most of the languages in this second group declare the parts of a drawing without control flow.

The final group contains those external libraries with which general-purpose languages draw pictures. Java's Shape classes are a prime example of such a library. These are not languages in themselves but rather additions to an existing language. Similarly, any general purpose language can output one of the aforementioned vector graphics formats.

The language we have created— tenuously dubbed Human Readable Graphics Language (HRGL)— fills the niche between the latter two groups. It contains the capabilities of a declarative, human-readable graphics language: it can draw lines, shapes, paths, strokes, and fills as well as perform transformations on collections of drawings. It also contains control structures, variables, and functions reminiscent of a general-purpose language. The difference lies in that the language contains specialized operations and structures tailored to drawing pictures.