

Tektronix Color Smalltalk, standard software on Tek's 4310 Series workstations, brings full color support to the traditionally monochrome Smalltalk-80 environment.

Tektronix Color Smalltalk: An Object-Oriented Approach to Graphics

by Rebecca J. Wirfs-Brock
Software Productivity Technologies
Tektronix, Inc.
Beaverton, OR

Four years ago, Tektronix was the first company to usher Smalltalk-80™ out of the research laboratories and into the hands of real-world users. Now, with the 4310 Series workstations, Tek becomes the first workstation vendor to provide full color support for the Smalltalk environment.

The new Tektronix Color Smalltalk, part of the standard software supplied with the 4310 Series bit-mapped workstations, brings integrated color to the traditionally monochrome world of object-oriented programming.

Programming Power

The leading example of object-oriented languages, Smalltalk gives programmers power and economy not found with procedural languages such as C and Pascal. Smalltalk contains a variety of features that speed code development—features that experienced Smalltalk programmers say make them many times more productive than when programming in a procedural, nondynamic language.

In conventional computer languages, a program consists of a series of specific instructions that outline what to do, where to do it, and how to get it done. All required instructions are contained or referenced by the main program, and the programmer must understand the finest details of implementation in order to write the code.

Smalltalk, in contrast, is nonprocedural and not strongly typed. Smalltalk uses objects to represent data. Objects can represent queues, character strings, numbers, dictionaries, rectangles, file directories, program compilers, and other items.

A Smalltalk program is a series of instructions, which take the form of messages directed to specific objects. By sending a message to an object, the programmer is, in essence, requesting it to do something, but not telling it how to get the job done. The “how” is the business of the object, not the programmer. Thus, Smalltalk applications tend to be smaller and more manageable than applications written in procedural languages. (Imagine a Pascal program that doesn't have to define its procedures or declare variable types because the system already understands those definitions, and

you'll have a simplified image of Smalltalk's directness.)

Since Smalltalk is not a typed language, a single Smalltalk method (or procedure) can serve many kinds of objects. A sorting routine, for example, can sort numbers, strings, or points. In the same manner, array objects can contain elements of any type. For instance, to display a circle, a programmer sends the circle the message, “display.” To view the colors of a color palette, the programmer sends the palette the same message.

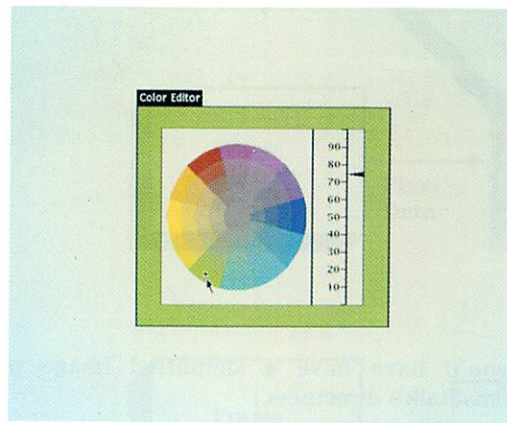
Smalltalk's hierarchical classes are another feature beneficial to programmers. Classes (or groups) of objects are arranged hierarchically, and subordinate classes can use the methods defined by any classes that appears above it in the hierarchy. Programmers define new classes that build on existing classes and their behaviors—a process known as programming by refinement, and one that offers a powerful technique for rapidly building viable applications by reusing as much code as possible.

In addition to the monochrome 4315 workstation, Tek Smalltalk adds gray scale support on the 4316 and color on the 4317.



The HLS color editor. Tek Color Smalltalk provides an editor for each of four supported color models: HLS, RGB, CMY, and gray-scale.

Pop-up menus make it easy to choose colors for screen background and windows.



Tektronix Color Smalltalk provides over 350 existing classes. The large number of classes gives the programmer an extensive set of components to use in constructing an application.

Interactive Programming

Adding to the power of the Smalltalk language is an interactive programming *environment*, which gives programmers an exceedingly rich set of development tools. Smalltalk reduces the conventional edit-compile-link-debug development cycle to a straightforward edit-debug process. In Smalltalk, there are no external editors, compilers, or debuggers. The language and the environment are one, and the programmer edits, runs, and debugs code within the context of that environment. Feedback is immediate: when code fails, a notifier informs the programmer, who has the option of continuing to run the code or debugging it. If the programmer opts to debug, the debugger highlights the portion of code that failed and lets the programmer look back through the complete stack of executing messages to pinpoint the problem. The programmer can then modify the offending code and continue code execution. The programmer can also sequence through the code a message at a time, examining and modifying the contents of any objects in the application.

Smalltalk includes in its windowed environment a number of built-in tools that enhance interactive code development. The browser

organizes access to all the Smalltalk code contained by the system. Through the browser, programmers can find specific portions of code, move quickly from one part of code to another, scroll through and read code, change code, and create code. And because of Smalltalk's interactive environment, all code changes take effect immediately.

The inspector is another interactive tool. During development, a programmer can run an application, then interrupt it and open an inspector, for a clear view of the objects the application uses and how they interact. Still another tool, the workspace, lets programmers experiment with code without actually incorporating the experimental code into the valid, running environment. A programmer can write, execute and debug code in a workspace, then pull it into the Smalltalk application when the new code is tested and operational.

Monochrome to Color

Along with its advanced facilities for code development, Smalltalk offers an excellent environment for interactive graphics applications (see sidebar). Only in this past year, however, has Smalltalk become available on color systems.

The integration of color into a monochrome environment was no simple matter. Color data is much more complex than monochrome data. To be effective, color must be fully integrated throughout Smalltalk, and not tacked on as a separate feature. Thus, Tektronix Color Smalltalk has been carefully designed so that the color models, dithering, graphics objects, and graphic editors work together in a cohesive system. At the same time, Tektronix Color Smalltalk remains compatible with Tek's monochrome Smalltalk.

In a Tek Smalltalk workstation, a color palette of 16 colors (or 16 shades of gray, on the 4316) is active for the display at any given time. Like a palette used by an artist, the display palette contains all the colors that are available for the display to use at one time. Again like the artist's colors, these 16 colors can be blended—dithered—to produce a great variety of shades.


Tektronix Color Smalltalk uses four well-known color models: hue-lightness-saturation (HLS), red-green-blue (RGB), cyan-magenta-yellow (CMY), and gray-scale. In the default color palette, eight colors are predefined: white, black, red, green, blue, cyan, yellow, and magenta, and the remaining eight palette entries can be defined interactively or with values specified in lines of Smalltalk code. A separate, interactive color editor is available for each color model, and any color can be converted from one color model to another with a single Smalltalk statement.

Tek Smalltalk also lets programmers define a different palette for every open window. When a window is selected for use (or, in Smalltalk terms, becomes "active"), its palette is installed as the display palette, and those color definitions define the colors that can be displayed.

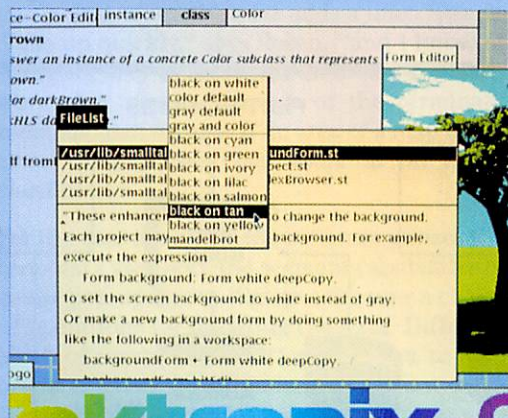
Enriching Applications

Smalltalk's speed and versatility make it a natural for graphics applications and allow it to provide a basis for applications ranging from integrated data analysis to natural language interfaces and complex simulations. Smalltalk is also widely used as a software prototyping tool, particularly for user interface prototyping.

Virtually any of these applications can be enriched by the addition of color. In user-interface simulations, for instance, color can signal that the user needs to perform a specific action: if an application consistently uses salmon-colored windows to request input from the user, then a salmon window that pops up will visually request the user's attention. Color can also point out error conditions (an application can literally raise red flags). On the hardware side, a Smalltalk-based circuit design tool can use color to represent different circuit board layers or to highlight sections of circuitry that have been selected for a specific function.

Whatever the target application, the integration of color into Tek's industrial-strength Smalltalk-80 is good news for object-oriented programmers and for the end-users of their applications. 

Object-Oriented Graphics



Tektronix Color Smalltalk fully integrates color into the Smalltalk programming environment. Programmers can choose a different palette for each open window.

Using the powerful graphics primitive called BitBlt (for bit block transfer), Smalltalk provides an excellent environment for creating graphics applications.

The simplest way to generate graphics in Smalltalk involves using a "pen" to draw lines, curves, arcs, circles and splines. Pens can draw images that are textured or multicolored, much like the "brushes" used in computer paint programs. Graphic images can be generated either on- or off-screen. Off-screen images, stored as bit-maps, can be displayed very quickly when needed. Bit-maps (or Forms as they are known in Smalltalk) can be combined with a clipping mask to produce non-rectangular graphic objects (or Opaque Forms). Shading and area-filling effects are possible by employing the appropriate halftone.

Smalltalk provides a bit editor for pixel-level editing of bit-maps. There's also a form editing tool for creating or editing full-size graphic images. The form editor and bit editor can be used to create a graphic form, refine a graphic image, or add color to existing monochrome Smalltalk graphics. Color is fully integrated into the bit- and form-editing tools. In addition, special color-editing tools provide direct access to the colors contained in the display palette. 