

Trillium Color

Pat Hannaway

File Created: July 10, 1984

Last Edit: September 13, 1984

Hardware

In order to run the color package, you must have a D0 (or Dorado) with a color board, and a color monitor. Be aware that the color board will drive the display at a resolution of 640x480 pixels regardless of the size of the display. Therefore, smaller displays (~ 13 in.) will be much sharper than larger displays. Also, because of this constraint, the interfaces designed in color can have a window that is no bigger than 640x480 (this corresponds to a region slightly larger than 1/4 of the regular D0 display). If you try to display a larger interface window, only the bottom left corner will show up on the color screen.

Getting Started with Color

To run the color package, you must have the Birthday84 release of Trillium. Edit the user profile and change the value of Color.Available to 'Yes'. This will pop up a menu that asks if you want the color primitive itemtypes loaded if they haven't been already. If you don't have any color primitive itemtypes loaded then you won't see any items on the color screen, so it's a good idea to say 'yes'.

Now the Lisp color software will be loaded (be sure you have the most recent versions of the <Lisp>Library and LispUsers directories), and, if you wanted, the itemtypes are loaded. Then Trillium will try to determine if any of the interfaces previously loaded need to be converted for color. This basically consists of just modifying the Colors frame to give you a selection of colors. The Colors frame will be changed if there is only 1 item of type Color in the frame (the default case when you created the interface), or if there are no items in the frame. If the frame Colors does not exist, it will be created. If for some reason, the colors can not be added (i.e. there have been other items added to the Colors frame), a message will print out in the promptwindow telling you what to type if you really do want to update the frame. If the color frame has 8 color items, assumed to be the eight you need to run color in Trillium, it will not be changed

To Run an Interface

Simply choose Turn.Color.On.Or.Off in the Control Window to turn on the color display. A message will print in the prompt window when it is turned on (the next time you choose this item, the color display will turn off and print an appropriate message). Then open the window for your interface - even if the interface was developed in black and white, you will still see it on the color screen (if the interface window was already open, just redisplay the frame to see it on the color screen).

When you edit the items, you will see new parameters that allow you to specify colors for the items.

Several new items have been added to the Manipulate.Interface menu. These include:

Set.Background.Color -- sets the background color on the color screen.

Switch.Color.Maps -- this allows you to choose among the default color maps.

Save.Color.Map -- this allows you to save a color map in order to transfer it to another interface or store it as insurance before you edit the current one.

Retrieve.Color.Map -- installs the saved color map on the current interface, and asks if you want to save the one already on the interface.

Edit.Color.Map -- this will allow you to edit the current Color Map and store it on your interface - see <Lisp>Library>Color.Txt for an explanation of how the editor works. If you choose to put a test pattern on the color screen while editing, the 8 colors in the top row represent the eight colors found in the color frame. The 8 colors in the bottom row correspond to the inverses of those colors. The first color in the top row will invert to the last color in the bottom row, the 2nd color in the top row will correspond to the 2nd-to-last color in the bottom row etc.

One other point about editing the color map - changing the color map does not change the names of the colors - i.e. if you change the color red to chartreuse, then in order to get an item to display in chartreuse, you must either choose 'red' in the menu of color names, or go to the color frame and change the name parameter of the Color item from red to chartreuse. Also note that each item in the Colors frame has a parameter called Representative.Gray. This is the shade in which the color associated with the item will be represented on the black & white screen.

Performance Considerations

Running the color display ties down a large block of memory, and uses around 30% of the processor time to refresh the color screen. Therefore, the less time spent with the color display on, the better. In order to minimize the waiting you'll have to do, try to always load the color software into a fresh sysout (after you've loaded your interface). This will give you reasonable performance (at least for a while), and have the hidden benefit of making sure you have a backup copy of your interface just in case.

While you don't want to turn the color display on and off constantly (every time the display is turned on the processor must find and tie down an amount of memory equal to the size of the color screen, and that gets harder each time you do it), if you are going to be doing quite a bit of work that will not require the color display, turn it off, and you'll get done what you want literally twice as fast.

One last recommendation - because of the virtual memory problems, you are much more susceptible to running out of memory, so if you do anything of importance, be sure to cleanup your files!

Please send me (Hannaway.wbst) any bugs and/or suggestions!