Color Smalltalk
July 28, 1988

Tom Almy

This document explains the the color "extension" to 80386 Smalltalk.

1 Color and the EGA

The EGA display supports four bitplanes. We will call those planes plane 1, 2, 4, and 8. At any pixel location the planes combine to form a color number 0 through 15. Each one of these colors can be any of 64 display colors (4 intensities each of R, G, and B).

The meaning of the bits in the display colors are:
32 - Low intensity red
16 - Low intensity green
8 - Low intensity blue
4 - High intensity red
2 - High intensity green
1 - High intensity blue

Monochrome Smalltalk programmed the color map so that a zero would be white and a one would be black, while the cursor would be an opaque blue. Normally the color map has white as color 15, black as color 0, and various colors in color 1-14.

Color Smalltalk programs the values as follows:

Value	Color
0	White (63), normal background
1	Black (0), normal foreground
2	Red (44)
3	Violet (61)
4	Yellow (62)
5	Green (58)
6	Orange (38)
7	Brown (14)
8	Bright Blue (49), cursor
9	Bright Blue (49), cursor
10	Bright Blue
11	Bright Blue
12	Bright Blue
13	Bright Blue
14	Bright Blue
15	Bright Blue

The background is color 0, standard Forms use color 1, and the cursor uses colors 8 up. For general use of color, one might want to change color 1 to blue, and 8 up to black. This would give the best subtractive color effect.

2 Form Types

Color Smalltalk 386 has several new classes to handle color. These new classes are subclasses of Form. The copybits primitive is modified to handle these new classes.

The display bitmap is now either a ColorForm or a Form rather than a DisplayBitmap.

2.1 BiColorForm

BiColorForms are Forms with two additional instance variables, foreColor and backColor. The foreColor is the color represented when the pixel in the form is a "1", while the backColor is the color represented when the pixel is a "0".

Class methods returning BiColorForms of the grey halftone forms are provided. BiColorForms can be used anywhere a Form is used. The foreground color defaults to 15, and the background color to 0. These can be changed with new instance methods foreColor:, backColor:, and foreColor:backColor. The colors can be inspected with foreColor and backColor.

Instance methods for red, violet, yellow, green, orange, and brown work like the shade-of-grey instance methods, providing the color palate has not changed. Instance methods black and blue assume the palette has been changed so that palette value 1 is blue and 8 is black.

2.2 Form

In most cases, a Form is treated like a BiColorForm with a foreground color of 1 and a background color of 0. The reason that the foreground color is not 15 is to improve performance of applications that are not using color.

The new instance method foreColor always answers '1', while the new instance method backColor always answers '0'. The dummy instance methods foreColor: backColor: foreColor:backColor: have been added for completeness.

Several methods have been modified to use "self class" rather than "Form" so that method inheritance will work properly.

2.3 ColorForm

A ColorForm is a subclass of Form which has an array of four WordArrays as its bitmap rather than a single WordArray. The WordArrays are those for planes 1, 2, 4, and 8, in that order.

There are many methods that need to be written for ColorForm to get full functionality. Most of them are missing. For instance, you cannot save or read ColorForms from files.

2.4 DisplayScreen

DisplayScreen (whose only instance is Display) has surprisingly few changes. The display can be defined as either a monochrome or color display with the methods displayExtent: and colorDisplayExtent:. When colorDisplayExtent: is used, the bitmap becomes an array of four WordArrays (like a ColorForm) rather than a single instance of WordArray. As a monochrome display, the performance of color Smalltalk matches that of the original monochrome version.

The instance method color instantly changes the display from monochrome to color. It does this by replacing the bitmap with an array of four bitmaps such as the first bitmap is the original and the other three are new with all bits equal to zero. The instance method monochrome instantly changes the display from color to monochrome by replacing the array of bitmaps with the first bitmap. Thus color only need be used when required by the application. It is important when switching from color to monochrome that the second through fourth planes be cleared before the change, otherwise the image will remain on the screen in a ghostly fashion.

The instance method isColor answers true if the display is in color mode, otherwise it answers false. The instance method displayClass answers ColorForm if the display is in color mode, otherwise it answers Form. All methods that save portions of the display (such as pop-up windows) have been changed to use Display displayClass rather than Form to create the image save forms.

2.4.1 The Color Palette

An additional method has been added, setPalette:to:, which allows programming the palette registers. The current color palette is saved in the global variable ColorPalette. This allows a snapshot image to restore the color palette. The interpreter automatically restores the color palette after a DOS shell is executed.

2.5 ColorPen

A new class, ColorPen, has been added as a subclass of Pen. ColorPen differs from Pen in that the source form is a BiColorForm rather than a Form. ColorPen has the following instance methods:

defaultColorNib:foreColor:backColor:
to set the pen tip to one that is round and of certain colors.

defaultNib:
to set the pen tip to round with colors the same as that of class Pen.

color:
to set the (foreground) color of the pen to one of 16 colors.

filberts:side:
hilberts:
mandala:diameter:
spiral:angle:
Geometric Design examples. The mandala uses random colors.

2.6 DisplayBitmap

This class has been deleted.

2.7 Display cursor

The display cursor is automatically displayed on plane 8. The cursor will "flash" on any bitblt operations that affect plane 8 in the region of the cursor.

3 BitBLT copyBits

In all cases the halftone mask can be either a BiColorForm, a Form, or nil. Unless otherwise indicated, a Form behaves like a BiColorForm with foreColor=1 and backColor=0, and nil is equivalent to Form black.

3.1 Destination is BiColorForm

The foreground and background colors of a destination BiColorForm are left unchanged in all operations.

3.1.1 Source is BiColorForm or Nil

Behavior is identical to that of monochrome Smalltalk.

3.1.2 Source is ColorForm

The current implementation is the source is reduced to a BiColorForm with the bitmap of plane 1. This provides compatibility with monochrome Smalltalk.

The Digitalk rule is: the source form is reduced to a BiColorForm by the rule: if foreground of halftone equals the pixel color the resulting color is 1, otherwise it is zero. Only the Form over rule is supported. This rule is difficult to implement, although it does have merit.

3.2 Destination is Form

This works identically to destination being a BiColorForm

3.3 Destination is ColorForm

3.3.1 Source is BiColorForm

The source form is first "expanded" into a color form using the colors of the halftone form, but if the halftone form is nil or of class Form then the colors of the source form are used.

Note that the expansion into a colorform does not really occur. The foreground and background colors are compared for each bit plane and the following chart is used:

Foreground bit	Background bit	Operation
0	0	Bitblt with source considered to be 0's
1	0	Normal Bitblt
0	1	Complemented Source
1	1	Bitblt with source considered to be 1's

Note that in the first and last cases it is possible that the resulting bitblt operation performs no operation (example: first case with ORing "under" mode). When this happens, and the destination is the display, that particular display plane will not be updated, giving a performance boost. Careful use of or, and, and exclusive-or modes along with careful selection of BiColorForm halftone mask colors will enhance performance.

3.3.2 Source is Nil

Halftone form is expanded into color form (using its foreground and background colors), then bitblt is performed on each plane. The comments in the previous section apply.

3.3.3 Source is ColorForm

Operation is performed independently on each plane.

4 Hardcopy

The hardCopy message (which writes the form to the file) only works for Forms and BiColorForms.
The method Form printOnPrinter, which provides a hook for instant hardcopy, has been retained. The class ColorForm redefines printOnPrinter by creating a Form with a bitmap being the first bitmap of the color form, and then sending printOnPrinter to it. Thus, a single color printer driver need only implement Form printOnPrinter. A color printer driver should implement both printOnPrinter methods.

In addition, the class DisplayScreen implements printOnPrinter in such a way that ColorForm printOnPrinter is used if the display is in color mode and Form printOnPrinter is used if the display is in monochrome mode. This method first copies the display forms, and then creates a new process to do the printing in the background.