Changes to 6502 Development Environment
— 19 October 1988

NEW HANDYCRAFT FEATURES

Handycraft now has two Free-Memory keys:

- 'I' frees the emulation sprites memory
- 'F' frees the edit sprites memory

Handycraft and Handebug talk the same function-key language now. When the Handycraft window is the active one, you can enter the Handebug function keys and get Handebug functionality.

Handycraft now allows you to upload partial sprite lists and to set the colors. In fact, now you *must* set the colors, else you get junk.

SPRITES AND THE DISPLAY

The SPRITES macro is now a macro that:

- Causes your sprites to be rendered
- Causes your offscreen buffer to become the display buffer

This is the macro you should use when you've got the sprite display list that you want rendered and then displayed. In code running on Handy hardware, this macro will start up the sprite engine and then hit the display register. In code running in Handy emulation mode, this code uploads your sprite list and then a DISPLAY command to Handycraft.

There is a new sprite macro, PSPRITE, which macro causes the sprites in the list to be rendered but it doesn't cause the offscreen buffer to become the new display. This is akin to invoking the sprite engine to render some sprites offscreen without displaying the partially-built buffer.

The macro DISPLAY will cause your offscreen buffer to be displayed.

In order to make the offscreen buffer become the new display, you could :

- Make zero or more calls to PSPRITE to build up parts of the display
- Call SPRITE with the last sprites of the display, or simply call DISPLAY

COLORS

To change the colors of your display, we now have some color-modification macros.

The macro RGB16 allows you to set up all 16 colors. You call this macro with the address of a 32-byte table where the color data is laid out in exactly the order that the hardware expects it: 16 bytes of green, 16 bytes of blue-red. Note that the macro presumes that your 32-byte table isn't in zero-page. On real Handy hardware, this macro will stuff your values directly into the color registers. In Handy emulation mode, your values are stashed to be uploaded the next time an instruction is sent that will cause the display to be rendered anew.

The next macro, RGB_AXY, expects that you have Red in A, Green in X, and Blue in Y, and that the argument to the macro is the handy hardware address for the green value of the pen you want to change. For instance, if you want to change pen 2 to white, do the following:

```
        LDA     #$0F
```

```
LDX         #$0F
LDY         #$0F
RGB_AXY     GREEN2
```

Note that when a macro causes the Handycraft display emulator to swap to a new display (either SPRITES or DISPLAY), the currently-defined colors are uploaded and displayed at the same time.