HMONSTER - a round-trip ticket from Windows BMP's to Apple II HGR Files

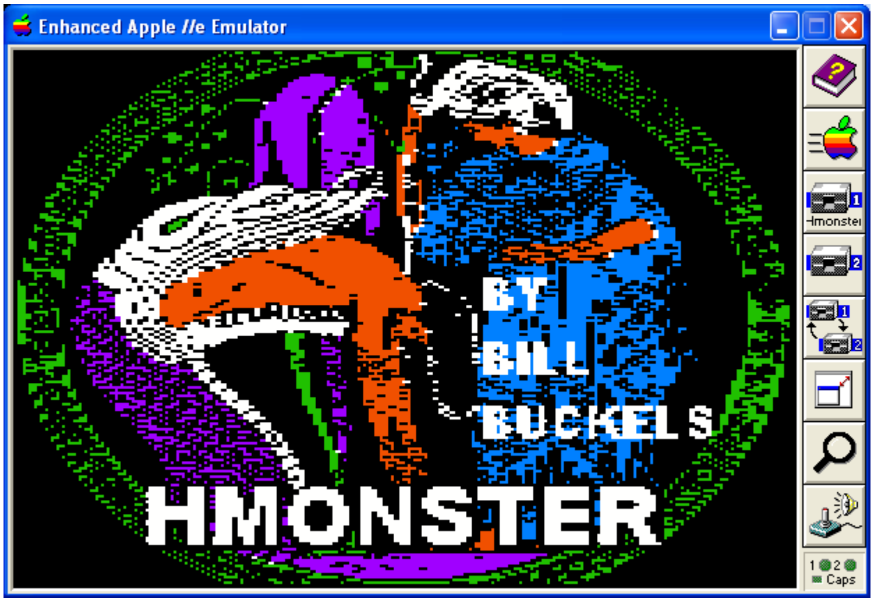
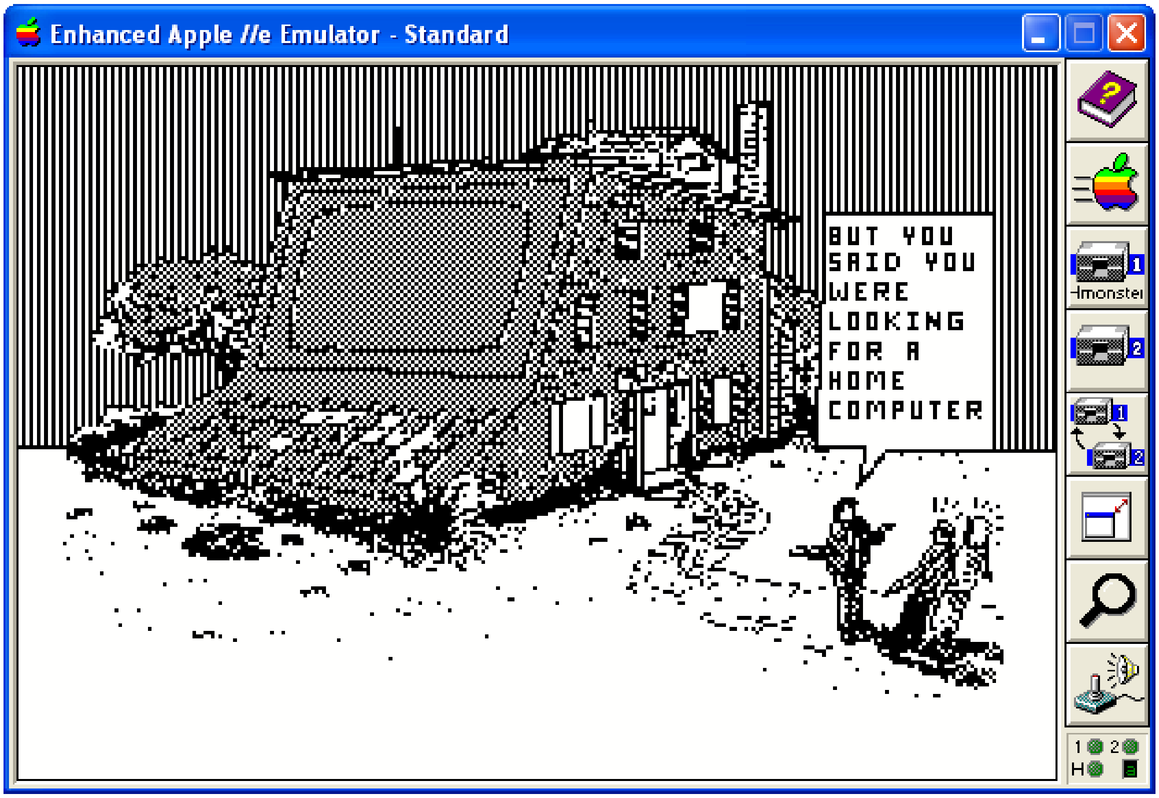


Table of Contents

Introduction



HMONSTER (hmon.exe) is a graphics file conversion utility program. It converts back and forth between Windows/MS-DOS graphics files, and Apple II Hi-Res (HGR) graphics screen files. It is an MS-DOS command-line application (written in Microsoft C), therefore it will run on ancient PC's and in MS-DOS emulators like DOSBox, as well as in a Windows XP cmd Window… in short it can be used on any computer that can support an MS-DOS session (which is just about any desktop and laptop machine on the planet).

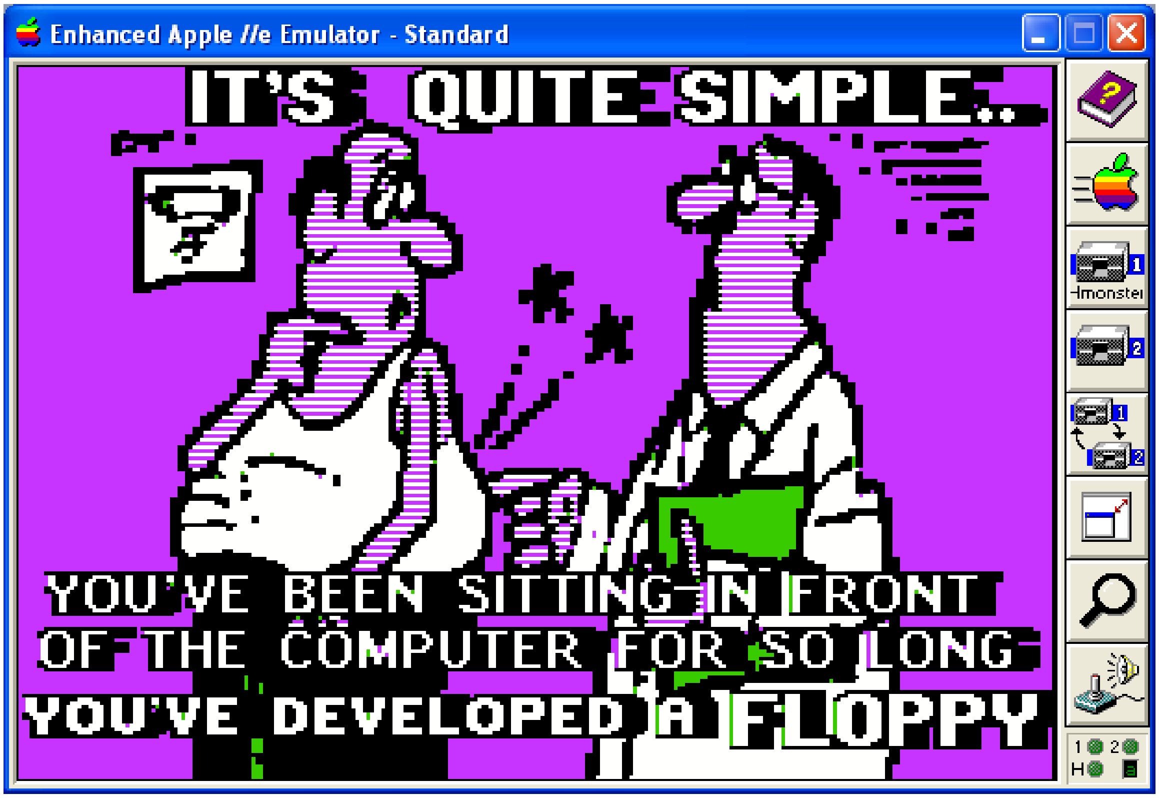
It serves 2 main functions:

1. Conversion of Monochrome Graphics Files (2 Color, Black and White Images) back and forth between Apple II HGR files and Windows bmp files.
2. Conversion of Color Graphics Files back and forth between Apple II HGR files and Windows bmp files.

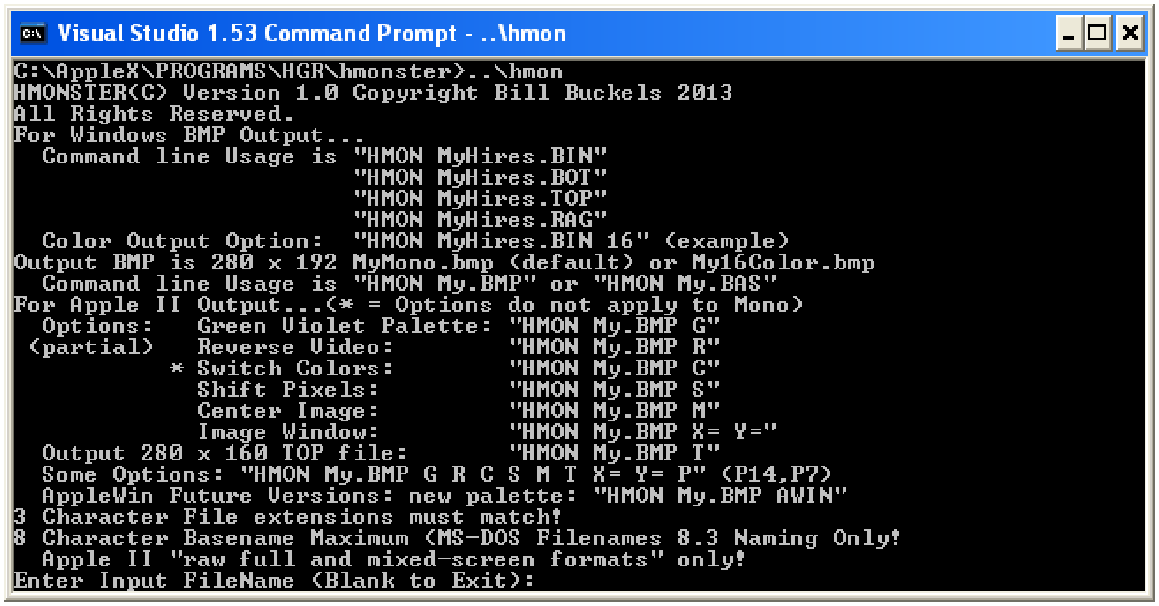
It also serves 2 secondary functions:

1. One-Way conversion from IBM-PC legacy 320 x 200 x 4 color GWBASIC BSaved CGA Screens to Apple II HGR files.
2. "Tweaking" of Windows/MS-DOS formats during conversion to Apple II HGR files.

HMONSTER Synopsis


Enhanced Apple //e Emulator - Standard
IT'S QUITE SIMPLE..
YOU'VE BEEN SITTING IN FRONT
OF THE COMPUTER FOR SO LONG
YOU'VE DEVELOPED A FLOPPY
Hmonster
1
2
H

The screen below displays HMONSTER's interactive prompt. But to enter options you must use HMONSTER from the command line.



```
C:\AppleX\PROGRAMS\HGR\hmonster>..\hmon
HMONSTER(C) Version 1.0 Copyright Bill Buckels 2013
All Rights Reserved.
For Windows BMP Output...
  Command line Usage is "HMON MyHires.BIN"
                        "HMON MyHires.BOT"
                        "HMON MyHires.TOP"
                        "HMON MyHires.RAG"
  Color Output Option:  "HMON MyHires.BIN 16" (example)
Output BMP is 280 x 192 MyMono.bmp (default) or My16Color.bmp
  Command line Usage is "HMON My.BMP" or "HMON My.BAS"
For Apple II Output...(* = Options do not apply to Mono)
  Options:   Green Violet Palette: "HMON My.BMP G"
 (partial)   Reverse Video:        "HMON My.BMP R"
           * Switch Colors:        "HMON My.BMP C"
             Shift Pixels:         "HMON My.BMP S"
             Center Image:         "HMON My.BMP M"
             Image Window:         "HMON My.BMP X= Y="
  Output 280 x 160 TOP file:       "HMON My.BMP T"
  Some Options: "HMON My.BMP G R C S M T X= Y= P" (P14,P7)
  AppleWin Future Versions: new palette: "HMON My.BMP AWIN"
3 Character File extensions must match!
8 Character Basename Maximum (MS-DOS Filenames 8.3 Naming Only!
  Apple II "raw full and mixed-screen formats" only!
Enter Input FileName (Blank to Exit):
```

Windows BMP and MS-DOS GWBASIC Input Formats

HMONSTER converts from the following IBM-PC (MS-DOS and Windows) Graphics file formats to Apple II HGR files:

- 2 Color (Monochrome) BMP Files in 280 x 192 or 320 x 200 resolution
- 16 Color (VGA) BMP Files in 280 x 192 or 320 x 200 resolution
- 256 Color BMP Files in 280 x 192 or 560 x 384 resolution
- 24 bit BMP Files in 280 x 192 or 560 x 384 resolution
- 320 x 200 x 4-Color GWBASIC BSaved Images (now "extinct")

Apple II HGR Output Formats from Windows/MS-DOS Files

The Apple II HGR files it creates are limited to 280 x 192 ("full-screen" Graphics-Only) and 280 x 160 (Mixed Text and Graphics size) Images in the following formats:

- BIN - 280 x 192 Color Images in BSaved (BIN) format
- BOT - 280 x 192 Color Images in Raster (RAG) format
- TOP - 280 x 160 Color Images in Raster (RAG) format

- BIN - 280 x 192 Monochrome Images in BSaved (BIN) format
- BOT - 280 x 192 Monochrome Images in Raster (RAG) format
- TOP - 280 x 160 Monochrome Images in Raster (RAG) format

Several Commandline Options are available.

Windows BMP Output Formats from Apple II HGR Files

HMONSTER Converts from the following Apple II Graphics file formats to Windows 280 x 192 x 16 Color BMP files:

- BIN - 280 x 192 Color Images in BSaved (BIN) format
- BOT - 280 x 192 Color Images in Raster (RAG) format
- TOP - 280 x 160 Color Images in Raster (RAG) format

HMONSTER Converts from the following Apple II Graphics file formats to Windows 280 x 192 x 2 Color (Monochrome) BMP files:

- BIN - 280 x 192 Monochrome Images in BSaved (BIN) format
- BOT - 280 x 192 Monochrome Images in Raster (RAG) format
- TOP - 280 x 160 Monochrome Images in Raster (RAG) format

Licence Agreement

You have a royalty-free right to use, modify, reproduce and distribute HMONSTER, its source code, and related demos and the other stuff it comes with; you may use this program for whatever you wish, provided that you agree that I have no warranty obligations or liability resulting from said distribution in any way whatsoever.

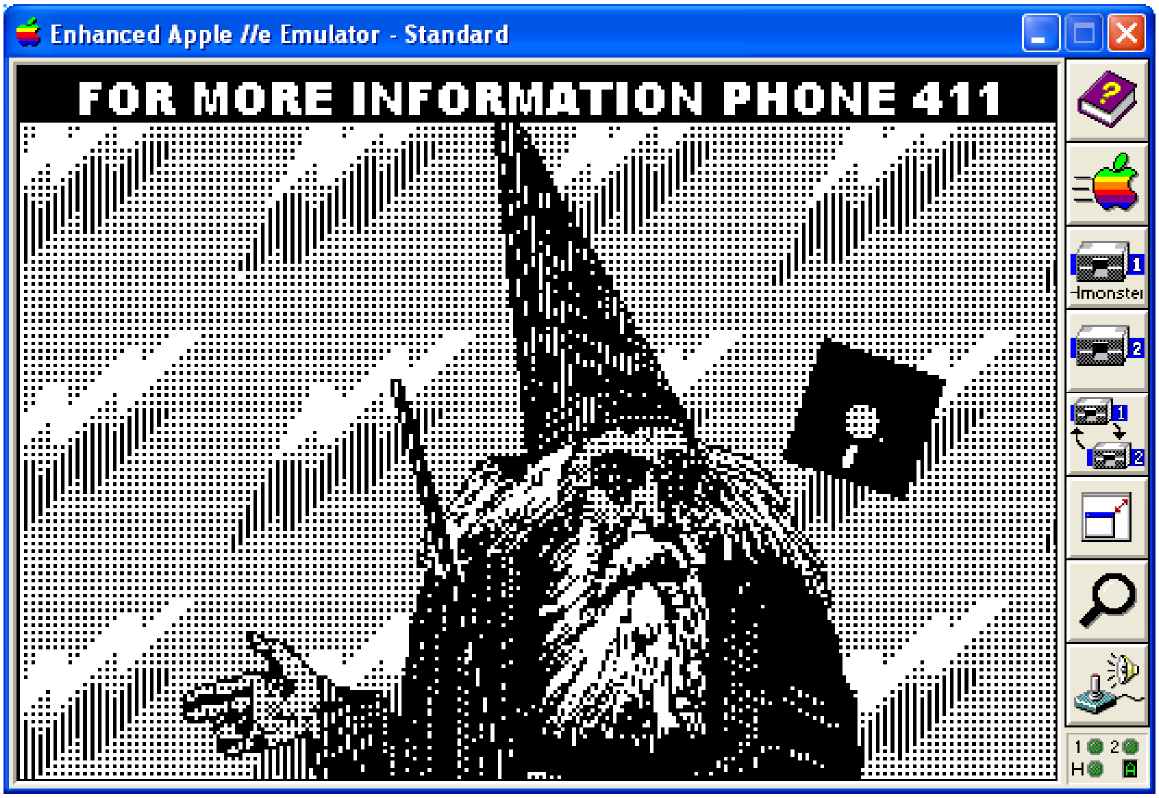
This seems fair enough since the rest of the planet can do exactly the same thing, and I hope they will.

Just one more thing before we get into the technical bits… the HGR images that you see in this document were converted using HMONSTER.

Historical Information and Developing HMONSTER

How HGR colors on the Apple II work is summarized in **Appendix A**. If you are not familiar with how HGR color work look at Appendix A before reading further.

Monochrome Backgrounder


Enhanced Apple //e Emulator - Standard
FOR MORE INFORMATION PHONE 411
Hmonster
1
2
H

HMONSTER (hmon.exe) started life innocently enough… in the beginning it was called hmono.exe and only worked with 2 color bmp files. Because the Apple II HGR display does not know the difference between monochrome graphics and color graphics, a converted Windows monochrome file will generate color on the Apple II on an Apple II color display. But whether a 280 x 192 image is a Monochrome Windows BMP or a 280 x 192 Apple II HGR image displayed on an Apple II monochrome display, monochrome images are equivalent wherever you hoe your row…. The pixel bit is either on or off.

Monochrome on the Apple II and Early PC's

In the 1980's and even in the early 1990's, many IBM-PC's only had a simple color display called a CGA (Color Graphics Adapter) if they had color at all. The CGA display worked similarly to the Apple II display when it came to displaying its standard 4 color image in monochrome mode (CGA mode 4 is 4 color and CGA mode 6 is monochrome). The "bit patterns" of the colors translated to "shades" or "dithers" of color bit pairs in monochrome, whether on an Apple II HGR display or on an IBM-PC display.

The difference between the displays themselves is the limiting factor when converting from the CGA to the less capable Apple II HGR display. I should note that the Apple II has a more "capable" monochrome display than HGR. DHGR (Double Hi-Res) is as capable as CGA monochrome but with a slightly lower resolution (560 x 192 as opposed to CGA's 640 x 200), and with exactly the same BSaved file-size (16384 bytes).

However, for practical purposes DHGR on the Apple II is slow to read and display when compared to HGR (which has a monochrome resolution of 280 x 192, less than half the monochrome resolution of the CGA). The plot thickens!

The file-size of the HGR Bsaved Image is 8192 bytes (half that of DHGR and CGA). The disk size of an Apple II is 140 K each side = 280 K.

The disk size of the "equivalent" IBM-PC of the day had evolved from 160 K single-sided, through 320 K double-sided, to 360 K as a "standard" by 1984, when IBM introduced 1.2 MB disks with their AT model. By the end of the 80's, 3.5 inch disks had taken-over even on early PC's, due to lower cost of PC peripherals and other factors. Disk access and processor speed on the IBM-PC was generally faster too.

For practical purposes, the DHGR BSaved Image Size only made sense if details were more important than load speed and disk space. It is for these overwhelming reasons that HGR was as preferable then as today, even though every second pixel of a 640 x 200 CGA monochrome image must be "thrown away" when converting from CGA.

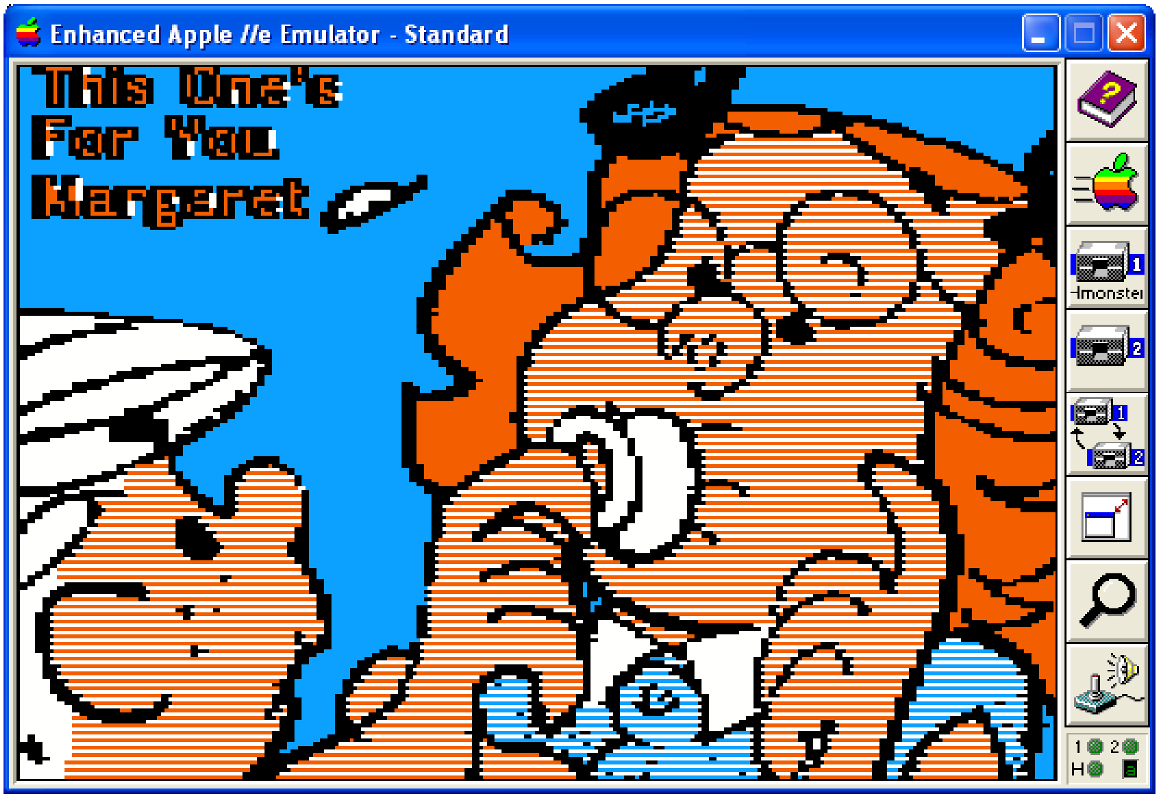
CGA 4 Color Backgrounder

After adding monochrome support to HMONSTER, I decided to add the ability to set the "palette bit" for the entire image in my Apple II output, so that a monochrome image with a dithered pattern could be displayed on an Apple II in 4 predetermined colors, allowing for either a Green-Violet palette or an Orange-Blue palette.

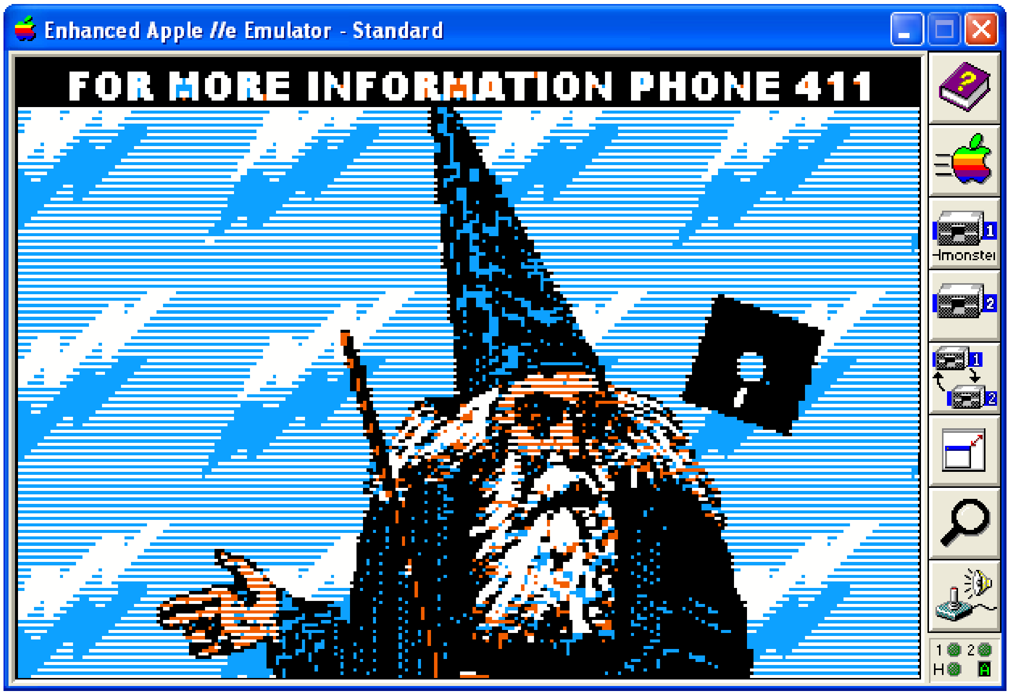
Doing so prompted me add conversion of old 320 x 200 x 4 color IBM-PC GWBASIC 320 x 200 x 4 color BSAVED images (now-obsolete) to HGR files. Even though every second pixel of a CGA 4 color image must also be “thrown away” when converting to HGR, color integrity can be preserved, especially if the CGA image contains pixel-pairs (i.e. colored lines and text 2 pixels wide). GIGO!

By setting the palette bit to a fixed 4 colors, the conversion results to HGR are eventually predictable from a 4 color IBM file, whether it is a CGA image or a BMP file.

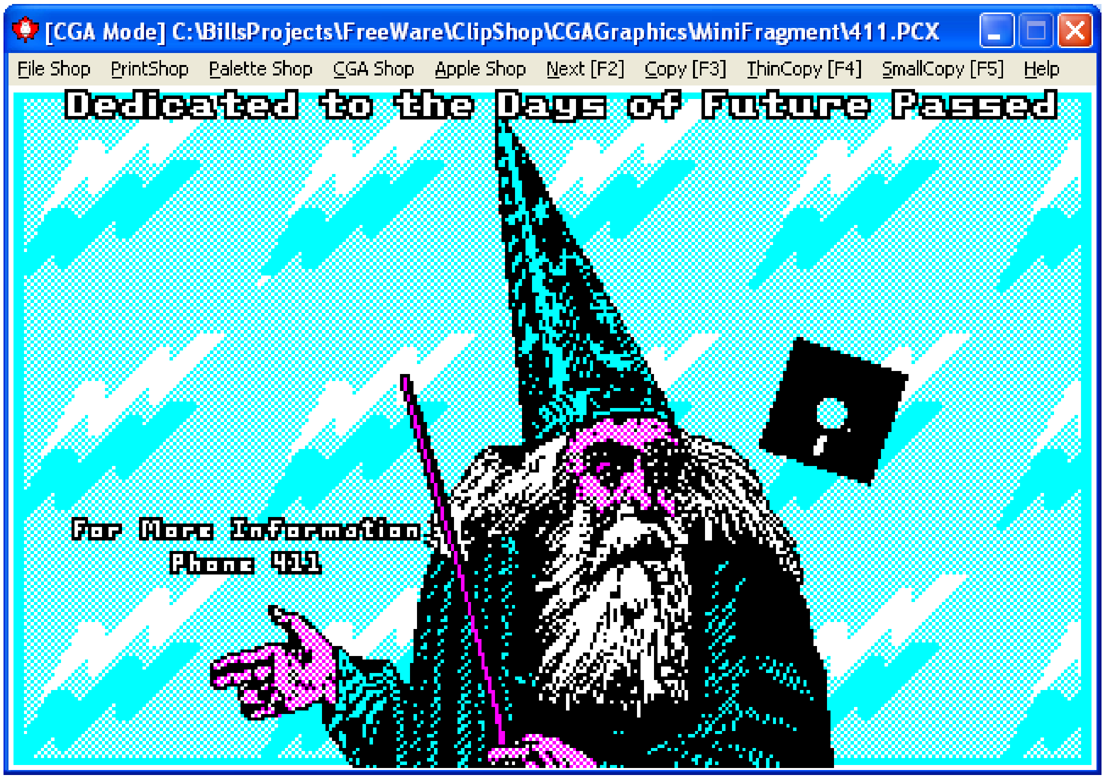
But back to what I said earlier about monochrome images being of the same general appearance whether IBM or Apple II… the wrinkle here is that colored images do not directly map from CGA 2 bit colors to Apple II 2 bit colors.



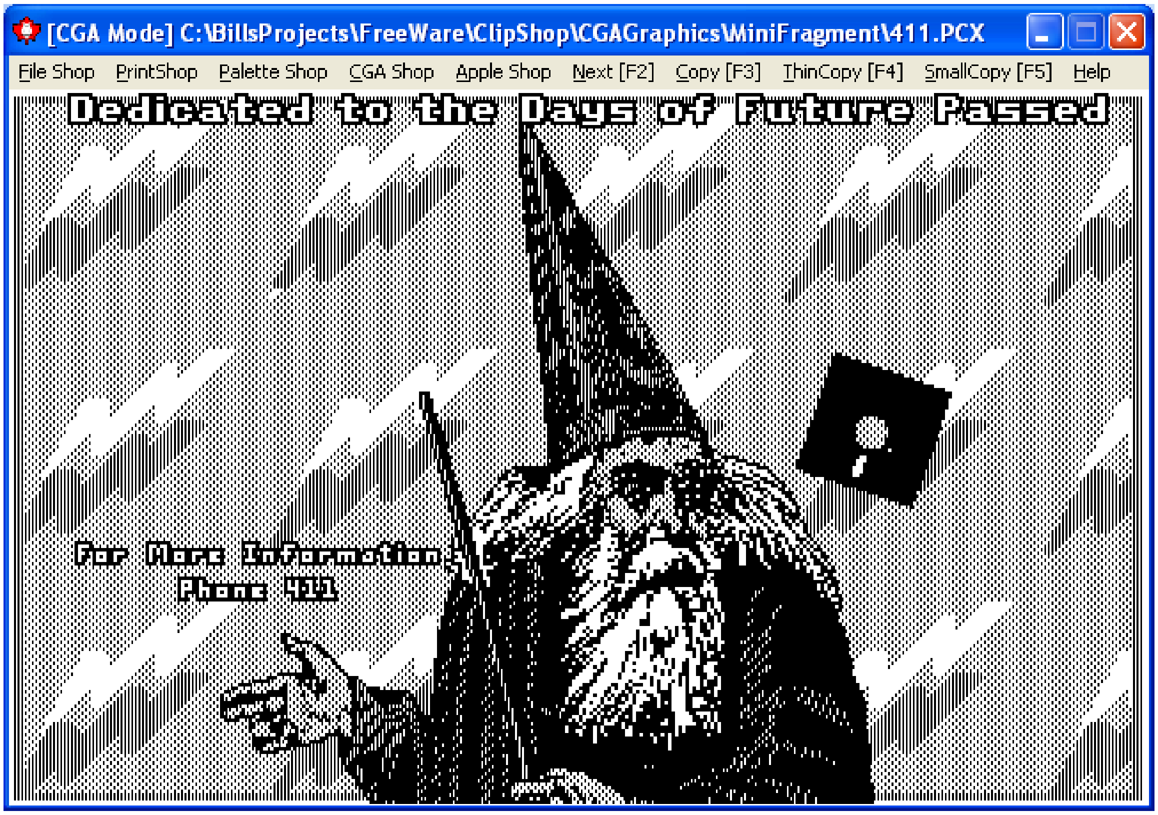
BOTTOM OF PAGE LEFT INTENTIONALLY BLANK
PROVIDE YOUR OWN GAP ANALYSIS HERE


Enhanced Apple //e Emulator - Standard
FOR MORE INFORMATION PHONE 411
monster
1
2
H
A

The image of the wizard above is exactly the same image as the one shown in the previous section on Monochrome. It started life as a 320 x 200 x 4 color CGA screen many years back during my port of Minipix to the PC. Original CGA screen below:



To prepare this file for conversion, I used **ClipShop** (another of my utilities) to display the file in simulated monochrome mode (shown below) and then made a ClipShop “Small Copy” to the Windows Clipboard. That reduced the Monochrome paste to 320 x 200. I pasted it into Windows Paint into a monochrome bitmap file, and removed the small text. It was illegible anyway. And I added some bigger text at the top. I made sure to preserve the color patterns that the monochrome image had “translated” during my cleanup. I then ran DMONSTER to convert it using a pixel shift of 1 pixel to the left so the colors mapped similarly.



If I had not shifted the monochrome image to the left, (or right), orange and blue would have been reversed and my colors would have looked terrible. Cyan on the CGA directly maps to Orange on the Apple HGR. Both have a left to right binary value of “01” (1), which translates directly to the pixel pattern in a monochrome bitmap. Magenta on the CGA directly maps to Blue on the Apple HGR. Both have a binary value of “10” (2). By offsetting into the monochrome BMP file by 1 pixel using HMONSTER, I reversed the color bitmapping and “presto”! I was indeed well-pleased…

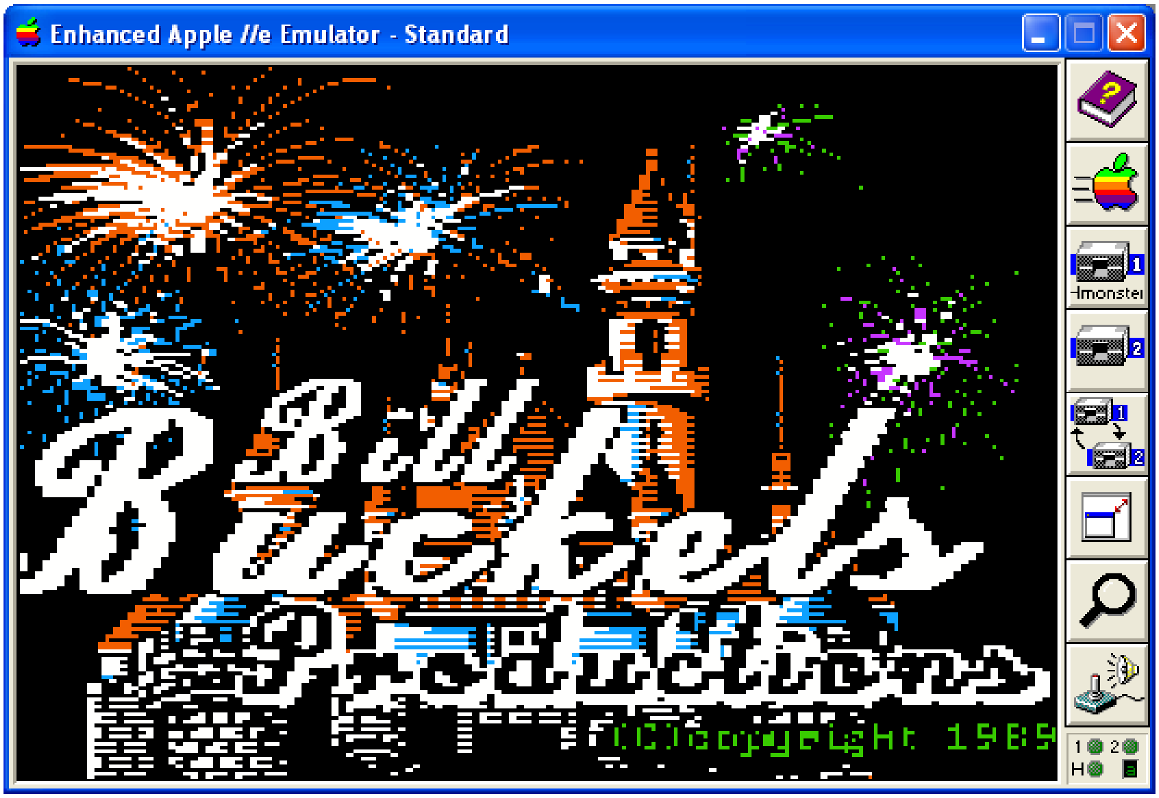
The monochrome representation of the Apple II image was not affected because the shift did not affect the appearance of the 1 bit pixel patterning. This long-winded example begins to explain how color works in 2-bit color, which the 4 color CGA and the Apple II HGR both use. It also explains the concept of shifting, which is a command line option that DMONSTER supports for images with a horizontal resolution of 320.

But the method I noted above is anything but "user friendly", unless you have been doing this as long as I have. The CGA converter for the old GWBASIC images isn't much good for most people either. Behind the scenes, HMonster does almost exactly what I explained above with a CGA image automagically, but alas, nobody has CGA images anymore, or editors for them (although ClipShop is bundled with several) .Windows Paint does not support a 2 bit color format. And on the IBM-PC, CGA mode only allowed 4 colors instead of the Apple II HGR's 6 colors.

Obviously HMONSTER needed more colors. But I did not wring my hands and weep and wail… Instead I added support for Windows 16 color BMP's as my next step.

Before leaving discussion of CGA and Monochrome altogether, I should note that I could also have corrected the image above somewhat, by switching to a Green and Violet palette. Cyan on the CGA maps directly to Green on the Apple II HGR display, and Magenta maps directly to Violet. I could have done that using HMONSTER's Green Palette command line option. The default palette in DMONSTER is Orange and Blue.

Windows 16 Color BMP's and Reducing to 6 HGR Colors



Colors in a 16 color bmp file are locked in Windows Paint. This provides you with the stable opportunity to create a picture in colors that are close enough to the Apple II HGR display that you can see what you will get color-wise on the Apple II.

Choose Red and Blue and Green and Magenta and Black and White in a 16 color BMP in Windows Paint, and they will end-up as Orange and Blue and Green and Violet and Black and White on the Apple II, after you convert with HMONSTER (using the P14 or P7 pixel grouping command-line option). More, or less… To begin with, on the HGR display, Orange and Blue cannot exist in the same 7 pixels as Green and Violet.

So you either have the option of sticking with 4 colors, like in CGA mode, or laying-out your image so it separates these conflicting colors grouping in 7 or 14 pixel blocks from left to right (using the P7 or P14 command line option).

The HGR pictures in this document that display all 6 HGR colors were laid-out in Windows Paint keeping this color pixel block grouping concept in mind. ClipShop also proved useful for remapping color palettes.

256 Color (SVGA) "Applewin" BMP Support and 24-Bit BMP Support

At this point in the development of HMONSTER, I began to look at using the screens from the **AppleWin** emulator as a source for graphics. Completely ignoring the fact that AppleWin itself saves its own screens, I used the Windows ClipBoard to capture an AppleWin Screen and pasted it into a 24 bit BMP and half scaled the screen to 280 x 192. as a 24 bit bmp, then saved it as a 16 Color bmp. This worked fine but was too much work. So I added support for single and double scaled 24-bit BMP's, again ignoring the fact that AppleWin saves its own screens to a 256 color bmp.

At this point in HMONSTER's development I grew less concerned about the support for the 320 x 200 IBM-PC formats, and instead did as AppleWin does, and supported the larger 560 x 384 size, as well as the miniscule 280 x 192 size. The higher resolution of the larger size is more "eye-friendly" in Windows on today's large screens.

But when editing for converting these 256 color and 24-bit bmps's, you must stick with the AppleWin colors, or with the same basic colors that are in the 16 color bmps.

Conclusion of the HMONSTER Story

I have said quite enough on the development of HMONSTER. And you can read the source code and use HMONSTER to learn more about how it converts to Apple II HGR.

I wish to note however, that in the dim past when I built Graphics File Converters for IBM to Apple HGR Graphics, I focused on trying to port all the detail of an IBM image to Apple II HGR with all sorts of fancy algorithms.

I still do this with my other contender, a utility called BMP2RAG. In the other corner, HMONSTER is all about byte-paired color and monochrome bit masks and managing colors and accepting the Apple II display for what it is.

HMONSTER (hmon.exe) was supposed to be a simple utility, and BMP2RAG was (and still is) a little more advanced and harder to use for most, since it manipulates color through algorithmic patterns that are a little more complex and must be correctly specified. Essentially a higher learning curve is required than for HMONSTER.

Pinging HGR files and BMP files – Round-Trip Editing

HMONSTER is also about “round-trip” conversion. A 280 x 192 x 16 color BMP file with color pairs of 2 bytes created in HMONSTER can be converted right back to the same 16 color BMP file from the destination HGR image since there are no enhancement algorithms (strictly speaking) that affect bit position. This featureless feature allows you to continue to edit the BMP in Windows Paint and ping it back from a converted Apple HGR image until you get what you want.

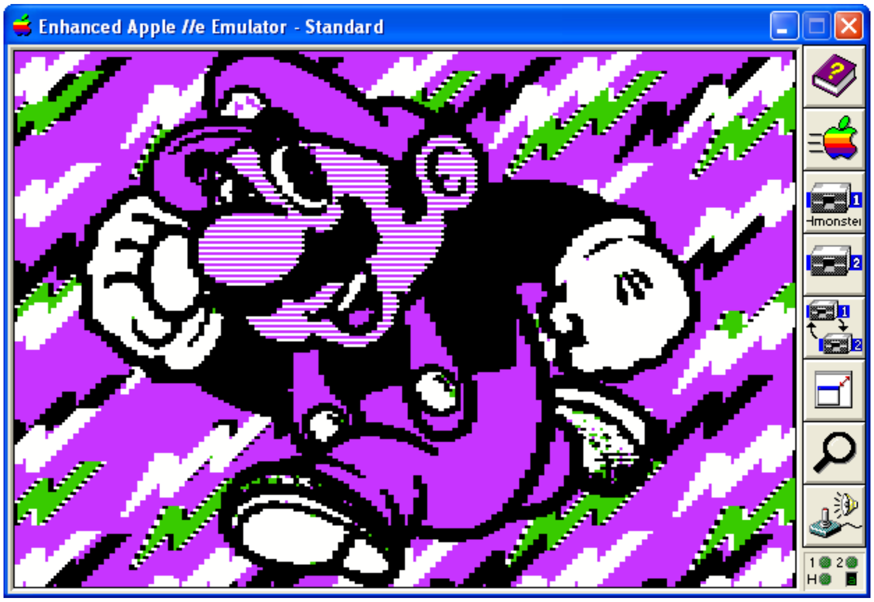
Monochrome Files are even easier because they port directly between the Apple and Windows Paint without change. WYSIWG. Only the palette bits are not preserved on the ride back from Apple HGR to Monochrome bmp.

Closing Remarks

Like anything, you simply need to understand the basics and build on them with your own experience. Or you could also take my approach and pretend to understand everything and be forced to write short novels like this one, and create monsters. A vicious circle if the truth were all that glitters.

‘Nuff said! Have Fun!

Bill Buckels
bbuckels@mts.net



Appendix A - Apple II HGR Byte Pair Diagrams

Note - In Figures 1 and 2, the Palette Select Bit could have been "Don't Care", since Black or White are available in either palette by using the same Pixel Pair Pattern.

Figure 1 - Black - Palette 0

```
All BLACK : byte value = '\x00'______ Palette Select Bit _____
                              :                               :
  0   1   2   3   4   5   6   7   0   1   2   3   4   5   6   7
-----------------------------------------------------------------
: 0 : 0 : 0 : 0 : 0 : 0 : 0 : 0 : 0 : 0 : 0 : 0 : 0 : 0 : 0 : 0 :
-----------------------------------------------------------------
: :   :   :   :   :   :   :     : :   :   :   :   :   :   :     :
: 0   1   2   3   4   5   6     : 7   8   9   10  11  12  13    :
: pixels                        :                               :
:___________ Byte ZERO _________:___________ Byte ONE __________:
```

Figure 2 - White - Palette 1

```
All WHITE : byte value = '\xff'______ Palette Select Bit _____
                              :                               :
  0   1   2   3   4   5   6   7   0   1   2   3   4   5   6   7
-----------------------------------------------------------------
: 1 : 1 : 1 : 1 : 1 : 1 : 1 : 1 : 1 : 1 : 1 : 1 : 1 : 1 : 1 : 1 :
-----------------------------------------------------------------
: :   :   :   :   :   :   :     : :   :   :   :   :   :   :     :
: 0   1   2   3   4   5   6     : 7   8   9   10  11  12  13    :
: pixels                        :                               :
:___________ Byte ZERO _________:___________ Byte ONE __________:
```

Figure 3 - Green - Palette 0

```
Palette 0 - Bit 7 is LOW - Violet EVEN pixel or Green ODD pixel

All Green (Turn Violet Off)    ______ Palette Select Bit _____
                              :                               :
  0   1   2   3   4   5   6   7   0   1   2   3   4   5   6   7
-----------------------------------------------------------------
: 0 : 1 : 0 : 1 : 0 : 1 : 0 : 0 : 1 : 0 : 1 : 0 : 1 : 0 : 1 : 0 :
-----------------------------------------------------------------
: :   :   :   :   :   :   :     : :   :   :   :   :   :   :     :
: 0   1   2   3   4   5   6     : 7   8   9   10  11  12  13    :
: V   G   V   G   V   G   V     : G   V   G   V   G   V   G     :
: pixel colors                  :                               :
:___________ Byte ZERO _________:___________ Byte ONE __________:
```

Figure 4 - Violet - Palette 0

```
Palette 0 -  Bit 7 is LOW - Violet EVEN pixel or Green ODD pixel

All Violet (Turn Green Off)    ______ Palette Select Bit _____
                              :                               :
  0   1   2   3   4   5   6   7   0   1   2   3   4   5   6   7
-----------------------------------------------------------------
: 1 : 0 : 1 : 0 : 1 : 0 : 1 : 0 : 0 : 1 : 0 : 1 : 0 : 1 : 0 : 0 :
-----------------------------------------------------------------
: :   :   :   :   :   :   :     : :   :   :   :   :   :   :     :
: 0   1   2   3   4   5   6     : 7   8   9   10  11  12  13    :
: V   G   V   G   V   G   V     : G   V   G   V   G   V   G     :
: pixel colors                  :                               :
:___________ Byte ZERO _________:___________ Byte ONE __________:
```

Figure 5 - Orange - Palette 1

```
Palette 1 -  Bit 7 is HIGH - Blue EVEN pixel or Orange ODD pixel

All Orange(Turn Blue Off)      ______ Palette Select Bit _____
                              :                               :
  0   1   2   3   4   5   6   7   0   1   2   3   4   5   6   7
-----------------------------------------------------------------
: 0 : 1 : 0 : 1 : 0 : 1 : 0 : 1 : 1 : 0 : 1 : 0 : 1 : 0 : 1 : 1 :
-----------------------------------------------------------------
: :   :   :   :   :   :   :     : :   :   :   :   :   :   :     :
: 0   1   2   3   4   5   6     : 7   8   9   10  11  12  13    :
: B   O   B   O   B   O   B     : O   B   O   B   O   B   O     :
: pixel colors                  :                               :
:___________ Byte ZERO _________:___________ Byte ONE __________:
```

Figure 6 - Blue - Palette 1

```
Palette 1 -  Bit 7 is HIGH- Blue EVEN pixel or Orange ODD pixel

All Blue (Turn Orange Off)     ______ Palette Select Bit _____
                              :                               :
  0   1   2   3   4   5   6   7   0   1   2   3   4   5   6   7
-----------------------------------------------------------------
: 1 : 0 : 1 : 0 : 1 : 0 : 1 : 1 : 0 : 1 : 0 : 1 : 0 : 1 : 0 : 1 :
-----------------------------------------------------------------
: :   :   :   :   :   :   :     : :   :   :   :   :   :   :     :
: 0   1   2   3   4   5   6     : 7   8   9   10  11  12  13    :
: B   O   B   O   B   O   B     : O   B   O   B   O   B   O     :
: pixel colors                  :                               :
:___________ Byte ZERO _________:___________ Byte ONE __________:
```