

Drazlace displayer

Written by HCL.

Start the prog with sys4096.

You also need a drazlace file named "tehGfx.prg", or you can haXX the code and insert gfx of your choice.

```
; Ok, Drazlace is from my point of view the most
; wasted gfx mode someone could ever have invented.
; Yet a lot of people seem to like it very much.
;
; So why is DrazLace totally st0pid? Because..
; The format it just big enough not to be squeezed
; into one video bank. Two bitmaps plus one colormap
; needs $4400 aligned bytes, where as one video bank
; is only $4000 bytes. Still the format does the
; least possible out of those two video banks.
; The one color map has to be duplicated into both
; video banks, where it would have been no effort
; to use two individual colormaps and thus widen
; the graphical possibilities a little.
;
; I hate DrazLace, but i hate more the fact that
; other people seem to like it :EEE.
; Here is a viewer for n00bs to this sucky phormat.

z1   = $02
z2   = $04

d021   = $7f40
d800map = $5800
colormap1 = $5c00
colormap2 = $dc00
bitmap1  = $8000
bitmap2  = $e000

      *= $5800
      .binclude "tehGfx.prg"

      *= $1000
      sei
      lda #$35
      sta 1

; First transfer some data from the original format,
; since it's not possible to show the picture the
; way it is stored when saved from editor.
```

```
; Copy bitmap to 1337 place in memory.
```

```
ldy #0
sty z1
sty z2
lda #>bitmap1
sta z1+1
lda #>bitmap2
sta z2+1
lda (z1),y
sta (z2),y
iny
bne *-5
inc z1+1
inc z2+1
bne *-11
```

```
; Copy colormap.. this is so st0pid!
```

```
dec 1
lda #>colormap1
sta z1+1
lda #>colormap2
sta z2+1
ldx #4
lda (z1),y
sta (z2),y
iny
bne *-5
inc z1+1
inc z2+1
dex
bne *-12
inc 1
```

```
; Also copy d800 colors. Yummie.
```

```
lda #>d800map
sta z1+1
lda #>d8
sta z2+1
ldx #4
lda (z1),y
sta (z2),y
iny
bne *-5
inc z1+1
inc z2+1
dex
bne *-12
```

```
; Now everything is done to start teh c0de pr0n.  
; Screen is going to flixx0r like hell :).  
  
; Since bitmap and colormap are placed on the same  
; relative place in the two video banks, only  
; dd00 (video bank) and d016 (finescroll x-position)  
; have to be canged every frame.  
  
    lda d021    ;RTFM to learn about VIC registers.  
    sta $d021  
    lda #$3b  
    sta $d011  
    lda #$78  
    sta $d018  
loop  
    bit $d011  
    bpl *-3  
    bit $d011  
    bmi *-3  
  
    lda #0  
    eor #1  
    sta *-3  
    ora #$d8  
    sta $d016  
    eor #$d9  
    asl  
    sta $dd00  
    jmp loop
```

From:
<https://codebase64.org/> - **Codebase 64 wiki**

Permanent link:
https://codebase64.org/doku.php?id=base:drazlace_displayer

Last update: **2015-04-17 04:31**

