

Sprite Stretching

Sprite stretching uses the technique of setting the bits of \$d017 to 1, and then back to 0 on the next rasterline. This will fool the VIC not to increase the internal sprite-gfx-pointer and display the same line of sprite-graphics again. By repeating this trick every rasterline, you can decide how many times each line of sprite-graphics will be shown.

Code example follows. Note that \$d017 is set from the table on the first rasterline. On the second rasterline it is set back to 0 and then set again to a new value from the table. Repeating on the third line and on.

```
*= $0900

sei
bit $d011 ; Wait for new frame
bpl *-3
bit $d011
bmi *-3

lda #$ff ; Enable sprites
sta $d015

ldx #14 ; Set some x-positions
clc
lda #$f0
sta $d000,x
sbc #$18
dex
dex
bpl *-7

ldx #14 ; Set some y-positions
lda #$40
sta $d001,x
dex
dex
bpl *-5

lda #$24 ; Set sprite pointers to display this code :).
ldx #7
sta $07f8,x
dex
bpl *-4

lda #$bd ; Set idle-pattern
sta $3fff
loop1
jsr StretchCalc ; Make beautiful stretching.
```

```
    lda #$40 ; Wait for sprite y-position
    cmp $d012
    bne *-3

    ldx #4 ; Wait a few cycles to make the d017-stretch work
    dex
    bne *-1

    ldx #0
loop2
    lda StretchTab,x ; $ff will stretch, 0 will step one line of graphics in
the sprite
    sta $d017

    sec
    lda $d011
    sbc #7
    ora #$18
    sta $d011 ; Step d011 each line to avoid badlines

    bit $ea ; Make the whole loop 44 cycles = one raster line when using 8
sprites
    nop
    nop
    nop

    lda #0 ; Set back for the next line
    sta $d017

    inx
    cpx #100
    bne loop2 ; Loop 100 times

    lda #$1b ; Set back char-screen mode
    sta $d011
    jmp loop1

StretchCalc ; Setup the stretch table
    ldy #0
    sty YPos
    lda #$ff ; First clear the table
    sta StretchTab,y
    iny
    bne *-4

    lda #0 ; Increase the starting value
    inc *-1
    asl
    sta AddVal

    ldy #0 ; This loop will insert 16 0:s into the table..
```

```
        ; At those positions the sprites will not stretch
```

```
SFT_1
    lda AddVal
    clc
    adc #10
    sta AddVal
    bpl *+4
    eor #$ff
    lsr
    lsr
    lsr
    lsr
    sec
    adc YPos
    sta YPos
    tax
    lda #0
    sta StretchTab,x
    iny
    cpy #20
    bcc SFT_1
    rts
```

```
YPos    .byte 0
AddVal  .byte 0
```

```
    .align $100 ; Align the table to a new page, this way lda StretchTab,x
always takes 4 cycles.
```

```
StretchTab
    .dsb 256 ; Reserve 256 bytes for the table
```

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

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

Last update: **2015-04-17 04:34**

