

FPP - by repeating the last line

Here follows a code example of FPP by repeating the last line of a char. The advantage of repeating the last line is the possibility of opening the sideborder, and having more time on each line. The drawback is less amount of usable graphic lines, only 8 lines per video-bank.

```

    *= $0900
    sei

    ldx #0 ; Put some crap data to make empty graphic banks visible
b_1
    lda #$f3
    sta $2000,x
    lda #$28
    sta $2100,x
    lda #$ad
    sta $2800,x
    lda #$99
    sta $2900,x
    lda #$71
    sta $3000,x
    lda #$ed
    sta $3100,x
    lda #$3b
    sta $3800,x
    lda #$cc
    sta $3900,x
    inx
    bne b_1

loop1
    bit $d011 ; Wait for new frame
    bpl *-3
    bit $d011
    bmi *-3

    lda #$1b ; Set y-scroll to normal position
    sta $d011
    lda #$17 ; Set normal d018
    sta $d018

    jsr SetupD018Table ; Nice FPS-effect

    lda #$51 ; Wait for position where we want FPD to start
    cmp $d012
    bne *-3

    ldy #10 ; Wait one more line..
    dey
```

```

    bne *-1
    nop
    nop
    cmp $d012 ; ..and make a bit more stabel raster
    bne *+5
    bit 0
    nop

    nop

loop2  ldx #0 ; Clear counter
      ldy #5 ; Wait some cycles
      dey
      bne *-1
      nop
      nop
      nop
      nop

      ldy D018Table,x
      lda $d012
      and #7
      ora #$18
      sta $d011 ; Do one line of linecrunch
      sty $d018 ; Set graphics bank of next line

      inx
      cpx #100
      beq loop1 ; Exit if end is reached
      jmp loop2 ; Otherwise loop

SetupD018Table
    lda #0 ; First clear the table
    ldy #0
    sta D018Table,y
    iny
    bne *-4

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

    lda #$10 ; Set starting position of stretch-effect
    sta YPos

    ldx #0
    ldy #0 ; This loop will calculate 8 different d018-values into the
table..
          ; Each value will be stretched down from the position of the

```

last value

; and thus simulate a stretcher effect (FPS).

SDT_1

```

lda AddVal
clc
adc #10
sta AddVal
bpl *+4
eor #$ff
lsr
lsr
lsr
lsr
sec
adc YPos
sta YPos
tya
asl

```

SDT_2

```

sta D018Table,x
inx
cpx YPos
bcc SDT_2

iny
cpy #8
bcc SDT_1
rts

```

```

YPos .byte 0
AddVal .byte 0

```

.align \$100 ; Align the table to a new page, this way lda D018Table,x always takes 4 cycles.

```

D018Table
.dsb 100,0 ; Reserve 100 bytes for the table

```

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

Permanent link: <https://codebase64.org/doku.php?id=base:fpp-last-line>

Last update: 2015-04-17 04:32

