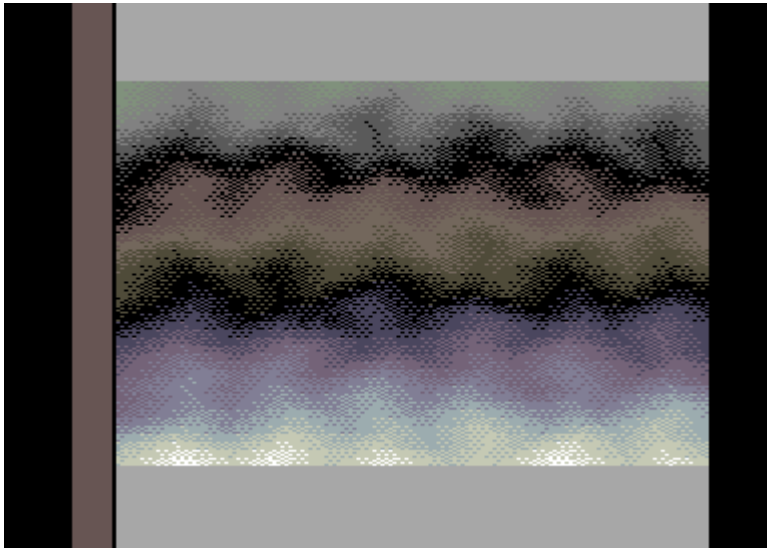


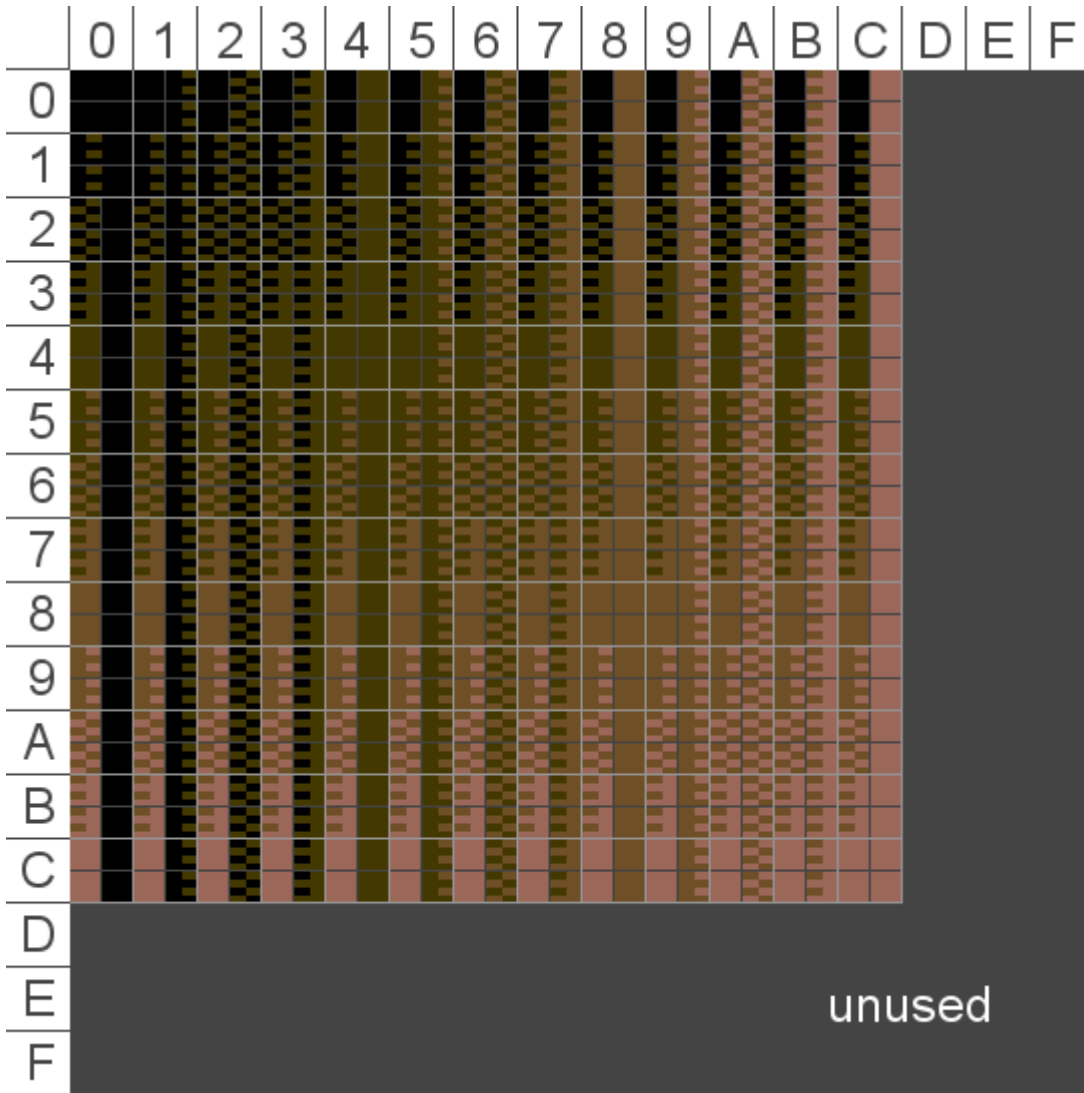
# 4x4 charset fire with lots of colors



## 4x4

For this effect a standard (that works with new and old CIA) 4x4 routine is used, but instead of bitmap a charset is displayed as half chars, thus we have 16 possible symbols that we can display per cell, the symbols only need to be arranged correctly as a charset so that the lownybble makes the right cell to change and the highnybble makes the left cell of a char to change.

If we want to make symbols that do a gradient from multicolor 1 to multicolor 4 we can best do that in 13 steps. This means that there are a few symbols left. So the first symbol is \$00 and the last is \$cc



As you can see, there are still some chars unused. Ideal to place a logo or gfx in there that can be combined at a granularity of 4x4 with the effect.

Also, you might imagine a bunch of other effects that can be done with that kind of mode, just think of the sphere-mapper in Coma Light 13.

## Data propagation in grid/matrix

For flames we need to do three things, they move upward, have a movement in x direction at the base and they fade when ascending. So what we need to do within the whole screen is to move the halfchars from a line to the line above and by that way also decrement the value of the symbols. So if we have a doublecell containing \$42 we need to propagate \$31 to the halfchar above.

Propagation within one char that is resembled by two halfchar lines.

```

|   |   |
|   |   |
| 3 | 1 |
| ^ | ^ |   force badline

```

```

| | | | |
| | | | |
| 4 | 2 |
|____|____|___ normal DMA happens and next chars are fetched

```

For derementing the halfchars one could make use of the sbx command, but problems occur when one of the nybbles underrun. The easiest way to get a decremented value is a lookup table. Whenever the values underrun you can either limit them to \$00 and make the fire thus cease at that point or wrap around to make it fill the whole screen repeatedly.

So for each halfchar you basically do:

```

ldx curr_line,y
lda lut,x
sta prev_line,y
...

```

## Flames/Input

Now if we'd do only that, we somewhen would end up in a matrix filled with all zeros or just statically scrolling upwards. Not very fireish indeed. So we need to feed in new high values at the bottom line that can then again be propagated. This we pick a spot per flame and let it move from left to right with some kind of sine or manual table. Also we build a descending gradient for each of those spots and in case two gradients of two spots meet, limit them to a value of one of the spots.

Example of two spots with gradients placed at the bottom line:

```

|00|01|23|45|43|23|45|67|89|87|65|43|00|00|00|...

```

## Additions

Now one can enrich things by adding some rastersplits / colorchanges in colram to have more colors displayed on the whole screen. Also a fadein and fadeout and a spritebar to cover the FLI-bug are a nice thing to have.

## Sourcecode

Yet to be enriched with comments.

```

!cpu 6510

      *= $6000

```

```
f_offs    = $30    ;6 bytes flame positions
count     = $36
colidx    = $37
maxval    = $38
tmp2      = $39
tmp1      = $3a
dir       = $3b
src       = $40
dst       = $42
screen    = $4000
charset   = $4800
raster    = $5800
nibbles   = $0340
waves     = $03a0

startline = $2d

ra        = $02
rx        = $03
ry        = $04

ra_       = $60
rx_       = $61
ry_       = $62

fadecnt   = $50
barcnt    = $51
reg_a     = $52

start_irq
    jsr do_setup
    sei

    ;vsync
    bit $d011
    bpl *-3
    bit $d011
    bmi *-3

    ;get a stable raster with half-variance approach
    ldx $d012
    inx
    cpx $d012
    bne *-3
    ldy #$0a
    dey
    bne *-1
    inx
    cpx $d012
```

```
    nop
    beq +
    nop
    bit $ea
+
    ldy #$09
    dey
    bne *-1
    nop
    nop
    inx
    cpx $d012
    nop
    beq +
    bit $ea
+
    ldy #$0a
    dey
    bne *-1
    inx
    cpx $d012
    bne +
+
    nop
    nop
    nop
    nop
    nop
    ;setup stuff for 4x4 mode irq
    lda #$3e
    sta $dc04
    sty $dc05
    lda #$11
    sta $dc0e
    lda #$7f
    sta $dc0d
    sta $dd0d
    lda #$01
    sta $d01a
    sta $d019
    lda #startline
    sta $d012
    lda #$35
    sta $01
    lda #<irq4
    sta $fffe
    lda #>irq4
    sta $ffff
    lda $dc0d
    lda $dd0d
    asl $d019
```

```
    lda #<nmi
    sta $fffa
    lda #>nmi
    sta $ffffb
    lda #$f7
    sta $dd04
    lda #$01
    sta $dd05
    cli
    jmp start

irq4
    sta ra
    asl $d019
    nop
    lda $dc04
    and #$07
    sta t4+1
    lda #$07
    sec

t4
    sbc #$04
    sta t5+1

t5
    bpl *+2
    cmp #$c9
    cmp #$c9
    cmp #$c9
    cmp #$24
    nop
    nop
    nop
    nop
    nop

    lda #<normirq
    sta $fffe
    lda #>normirq
    sta $ffff

    bit $ea
    lda #$11
    sta $dd0e
    lda #$81
    sta $dd0d
    lda $dd0d
val
    lda #$34
    sta $d012
    lda #$32
    cmp $d012
    bne *-3
    nop
    nop
```

```

        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
border3  lda #$0c
        sta $d021
        lda ra
        rti
normirq
        sta ra

        ;setup next irq
        lda #<stopnmi
        sta $fffe
        lda #>stopnmi
        sta $ffff
        lda #$f6
        sta $d012

        ;switch raster/fire
rush2   lda #$62
        sta p2+1
rush1   lda #$72
        sta p1+1

        ;decrease irq-position if further possible (this does the fade in
effect)
        lda val+1
        sec
        sbc #$04
        cmp #$30
        beq +
        sta val+1
+

```

```
        lda ra
        asl $d019
        rti
stopnmi
        sta ra
        stx rx
        sty ry
        nop
        nop
        nop
        nop
border  lda #$02
        sta $d021
        lda #<irq4
        sta $fffe
        lda #>irq4
        sta $ffff
        lda #$02
        sta $d018
        lda #startline
        sta $d012
        lda #$00
        sta $dd0e
        lda $d011
        eor #$08
        sta $d011
        lda #<d021tab
        sta d1+1
        lda #<d022tab
        sta d2+1
        lda #<d023tab
        sta d3+1
        lda d023tab
        sta .uff+1

        lda #$fd
        cmp $d012
        bne *-3
        lda $d011
        eor #$08
        sta $d011
        jsr irq_code
start_fo
        lda #$00
        beq +
        lda #>fo
        pha
        lda #<fo
        pha
        lda #$00
```



```
        pha
+
        lda ra
        sta ra_
        ldx rx
        stx rx_
        ldy ry
        sty ry_
        asl $d019
        rti

nmi
        ;will be manipulated to fit our needs depending on cia-type
code1   sta code2+1
        lda $dd04
        eor #$e7
        sta t6+1
t6      bpl *
        cmp #$c9
        cmp #$c9
        cmp #$24
        nop

p1      lda #$72
        sta $d018
d011_1  lda #$7f
        sta $d011

d1      lda d021tab
        sta $d021
d2      lda d022tab
        sta $d022
.uff    lda #$00
        sta $d023

p2      lda #$62
        sta $d018
d011_2  lda #$7b
        sta $d011

        inc d1+1
        inc d2+1
        inc d3+1
d3      lda d023tab
        sta .uff+1

code2   lda #$00
        jmp $dd0c

detect_cia
        lda $d011
        and #$0f
```

```
    sta $d011
    lda $d011
    bpl *-3
    lda $d011
    bmi *-3

    sei
    lda $01
    sta tmp2
    lda #$35
    sta $01
    lda #<detect
    sta $fffa
    lda #>detect
    sta $fffb

    lda #$81
    sta $dd0d

    lda #$04
    sta $dd04
    lda #$00
    sta $dd05

    sta tmp1

    lda #%10011001
    sta $dd0e

    lda $dd0d
    lda $dd0d
    inc tmp1
    jmp *
```

detect

```
    lda $dd0d
    pla
    pla
    pla
    lda tmp1
    beq is_new
    lda tmp2
    sta $01
    cli
    rts
```

is\_new

```
    ;new CIA is used, manipulate code
    lda #$85 ;sta $xx
    sta code1+0
```

```
    lda #reg_a
    sta code1+1
    sta code2+1

    lda #$ea ;nop
    sta code1+2

    lda #$a5 ;lda $xx
    sta code2+0
    lda tmp2
    sta $01
    cli
    rts
```

#### do\_setup

```
    jsr detect_cia
    lda #$40
    sta $dd0c

    lda #$0b
    sta $d011
    jsr setup
    jsr gen_raster
    lda #$1f
    sta d011_1+1
    lda #$1b
    sta d011_2+1

    ;cover sprite
    ldx #$3e
-
    lda #%11111110
    sta $5000,x
    dex
    lda #%11111111
    sta $5000,x
    dex
    lda #%10111111
    sta $5000,x
    dex
    bpl -

    lda #$40
    ldx #$07
-
    sta screen + $3f8,x
    sta screen + $7f8,x
    sta raster + $3f8,x
    sta raster + $7f8,x
    dex
    bpl -
```

```
lda #$02
sta $d00
lda #$00
sta $7fff
lda #$18
sta $d016
rts
```

#### setup\_sprites

```
lda #$fe
sta $d015
sta $d017
sta $d01c
lda #$00
sta $d01b
sta $d01d
sta $d028
sta $d029
sta $d02a
sta $d02b
sta $d02c
sta $d02d
sta $d02e
sta $d025
sta $d026

lda #$18
sta $d002
sta $d004
sta $d006
sta $d008
sta $d00a
sta $d00c
sta $d00e
lda #$00
sta $d003
sta $d010
clc
adc #$2a
sta $d005
adc #$2a
sta $d007
adc #$2a
sta $d009
adc #$2a
sta $d00b
adc #$2a
sta $d00d
adc #$2a
```

```

        sta $d00f
        rts
start
        jsr setup_sprites
        ;do $44 dry runs without displaying to populate whole matrix
-
        jsr go
cnt1    lda #$44
        beq +
        dec cnt1+1
        jmp -
+
        inc irq_code+1
-
        ;wait until fade out is triggered
        lda fo+1
        cmp #$40
        bne +
        jmp *
+
.cnt    lda #$00
        jsr go
        inc .cnt+1
        bne -
        inc start_fo+1
        jmp -

setup
sll     ldx #$04
        lda tabset,x
        sta f_offs,x
        dex
        bpl sll
        inx
        stx maxval
        stx count
        stx colidx
        stx dir
        stx fadeCnt
        stx barcnt

        ;setter for colram used for fading
        jsr gen_colram

        ;clear screens
        ldx #$00
        lda #$44
-
        sta $4000,x
        sta $4100,x

```

```
    sta $4200,x
    sta $42e8,x
    sta $4400,x
    sta $4500,x
    sta $4600,x
    sta $46e8,x
    dex
    bne -

    ;generate charset
    jmp mkset

endl
    lda #$15
    sta $d018
    lda #$03
    sta $dd00
    sei
    lda #$48
    sta $fffe
    lda #$ff
    sta $ffff
    lda #$37
    sta $01
    cli
    rts

irq_code
    lda #$00
    bne +
    jsr fade
    jmp ++

+
    cmp #$01
    bne ++
    inc irq_code+1
    lda val+1
    cmp #$34
    bne ++
    lda dir
    eor #$01
    sta dir
    lda #$f4
    sta val+1

++
    lda dir
    bne +

buff1  lda #$02
        sta p2+1

buff2  lda #$12
```

```

    sta p1+1
    lda #$62
    sta rush2+1
    lda #$72
    sta rush1+1
    jmp over2
+
    lda #$62
    sta p2+1
    lda #$72
    sta p1+1
    lda #$02
    sta rush2+1
    lda #$12
    sta rush1+1
over2
    rts

tabset
    !byte $00,$08,$10,$05,$15

!align 255,0

;macro for setting wave/flame

!macro set_wave .form_l, .form_r, .pos {
    ldy .form_l
    lda maxval
    sec
    sbc tab3,y
    tay
    lda #.pos
    ldx .form_r
    jsr setfl2
}

;update matrix and wave/flame inputs
go
    ldx #$4f
    lda #$00
-
    sta waves,x
    dex
    bpl -

    lda maxval
    cmp #$1f
    beq *+4
    inc maxval

+set_wave f_offs+2, f_offs+0, $00
+set_wave f_offs+3, f_offs+1, $0c

```

```
+set_wave f_offs+0, f_offs+2, $18
+set_wave f_offs+1, f_offs+3, $24
+set_wave f_offs+2, f_offs+4, $30
+set_wave f_offs+4, f_offs+0, $3c

jsr incfl
jmp setit

incfl
    ldx #$04
-
    ldy f_offs,x
    iny
    cpy #$18
    bne +
    ldy #$00
+
    sty f_offs,x
    dex
    bpl -
    rts

    ;sets flame with selected gradients to left and right and at
desired position
setfl2
    sty tmp2

    sec
    sbc tab2,x
    clc
    adc #$14
    sta tmp1
    tax

    tya
    bmi end_sfl
-
    cmp waves-4,x
    bcc ++
    sta waves-4,x
+
    sbc #$01
    bmi ++
    dex
    bpl -
++
    ldx tmp1
    lda tmp2
-
    cmp waves-4,x
```



```

        bcc ++
        sta waves-4,x
+
        sbc #$01
        bmi ++
        inx
        cpx #$28
        bne -
++
end_sfl
        rts
tab2
        !byte $00,$01,$01,$02,$02,$03
        !byte $02,$02,$01,$00,$01,$02
        !byte $01,$01,$00,$01,$02,$03
        !byte $03,$02,$02,$01,$01,$00
tab3
        !byte $00,$00,$01,$03,$05,$06
        !byte $06,$06,$05,$03,$01,$00
        !byte $00,$00,$01,$02,$02,$03
        !byte $03,$03,$02,$02,$01,$00

        ;now propagate new values in matrix
!macro propagate .buffer {
        ldy #$27
        sec
-
!set .scr = .buffer + 1 * 40
!for .block, 24 {

        ;load line
        ldx .scr,y
        lda sbctab,x
        ;store result in next line
        sta .scr + $400 - 1 * $28,y

        ldx .scr + $400,y
        lda sbctab,x
        sta .scr,y
        !set .scr = .scr + $28
}

        sty tmp1
        tya
        asl
        tay
        ;calc new value for first line
        ldx waves+1,y
        lda nibbles,x
        ldx waves+0,y
        ora nibbles+$30,x

```

```
    ldy tmp1
    sta .buffer + 24 * $28,y

    dey
    cpy #$02
    beq +
    jmp -
+
}
```

setit

```
+propagate screen
rts
```

```
;-----
;GENERATE CHARSET
;-----
```

mkset

```
    lda #>charset
    sta dst+1
    lda #<charset
    sta dst
```

```
    ldx #$10
```

```
--
    ldy #$7f
```

```
-
    lda setb,y
    sta (dst),y
    dey
    bpl -
    lda dst
    eor #$80
    sta dst
    bne *+4
    inc dst+1
    dex
    bne --
```

```
    lda #>charset
    sta dst+1
```

```
    lda #<setb
    sta src
    lda #>setb
    sta src+1
```

mk4

```
    ldx #$10
```

mk3

```
ldy #$07
mk5
lda (src),y
asl
asl
asl
asl
ora (dst),y
sta (dst),y
dey
bpl mk5
lda dst
clc
adc #$08
sta dst
bcc *+4
inc dst+1
dex
bne mk3

lda dst+1
cmp #(>charset)+$08
beq mkdone
lda src
clc
adc #$08
sta src
bcc *+4
inc src+1
jmp mk4
mkdone
ldx #$00
ldy #$00
mkd2
tya
sta nibbles,x
sta nibbles+1,x
asl
asl
asl
asl
sta nibbles+$30,x
sta nibbles+$31,x
cpy #$0f
beq *+3
iny
inx
inx
cpx #$30
bne mkd2
rts
```

### gen\_colram

```
        ldx #$00
--
        ldy #$27
        lda d800tab,x
-
cr1     sta $d800,y
        dey
        bpl -
        lda cr1+1
        clc
        adc #$28
        sta cr1+1
        bcc +
        inc cr1+2
+
        inx
        cpx #$19
        bne --
        rts
```

*;fill screens with respective chars to make those huge bars appear after color fade in*

### gen\_raster

```
        ldx #$00
--
        ldy #$27
-
        lda rastab,x
ras1    sta raster+$000,y
        lda rastab+1,x
ras2    sta raster+$400,y
        dey
        bpl -
        lda ras1+1
        clc
        adc #$28
        sta ras1+1
        sta ras2+1
        bcc +
        inc ras1+2
        inc ras2+2
+
        inx
        inx
        cpx #$32
        bne --
        rts
```

```
!align 255,0
;subtraction table for propagation
sbctab
    !byte
$ff,$f0,$f1,$f2,$f3,$f4,$f5,$f6,$f7,$f8,$f9,$fa,$fb,$fc,$fd,$fe
    !byte
$0f,$00,$01,$02,$03,$04,$05,$06,$07,$08,$09,$0a,$0b,$0c,$0d,$0e
    !byte
$1f,$10,$11,$12,$13,$14,$15,$16,$17,$18,$19,$1a,$1b,$1c,$1d,$1e
    !byte
$2f,$20,$21,$22,$23,$24,$25,$26,$27,$28,$29,$2a,$2b,$2c,$2d,$2e
    !byte
$3f,$30,$31,$32,$33,$34,$35,$36,$37,$38,$39,$3a,$3b,$3c,$3d,$3e
    !byte
$4f,$40,$41,$42,$43,$44,$45,$46,$47,$48,$49,$4a,$4b,$4c,$4d,$4e
    !byte
$5f,$50,$51,$52,$53,$54,$55,$56,$57,$58,$59,$5a,$5b,$5c,$5d,$5e
    !byte
$6f,$60,$61,$62,$63,$64,$65,$66,$67,$68,$69,$6a,$6b,$6c,$6d,$6e
    !byte
$7f,$70,$71,$72,$73,$74,$75,$76,$77,$78,$79,$7a,$7b,$7c,$7d,$7e
    !byte
$8f,$80,$81,$82,$83,$84,$85,$86,$87,$88,$89,$8a,$8b,$8c,$8d,$8e
    !byte
$9f,$90,$91,$92,$93,$94,$95,$96,$97,$98,$99,$9a,$9b,$9c,$9d,$9e
    !byte
$af,$a0,$a1,$a2,$a3,$a4,$a5,$a6,$a7,$a8,$a9,$aa,$ab,$ac,$ad,$ae
    !byte
$bf,$b0,$b1,$b2,$b3,$b4,$b5,$b6,$b7,$b8,$b9,$ba,$bb,$bc,$bd,$be
    !byte
$cf,$c0,$c1,$c2,$c3,$c4,$c5,$c6,$c7,$c8,$c9,$ca,$cb,$cc,$cd,$ce
    !byte
$df,$d0,$d1,$d2,$d3,$d4,$d5,$d6,$d7,$d8,$d9,$da,$db,$dc,$dd,$de
    !byte
$ef,$e0,$e1,$e2,$e3,$e4,$e5,$e6,$e7,$e8,$e9,$ea,$eb,$ec,$ed,$ee

;charset to be generated (each line describes one symbol by bitmasks
setb
    !byte $00,$00,$00,$00
    !byte $00,$00,$00,$00
    !byte $08,$00,$02,$00
    !byte $08,$00,$02,$00
    !byte $02,$08,$02,$08
    !byte $02,$08,$02,$08
    !byte $0a,$08,$0a,$02
    !byte $0a,$08,$0a,$02
    !byte $0a,$0a,$0a,$0a
    !byte $0a,$0a,$0a,$0a
    !byte $06,$0a,$09,$0a
    !byte $06,$0a,$09,$0a
    !byte $09,$06,$09,$06
```

```
!byte $09,$06,$09,$06
!byte $05,$06,$05,$09
!byte $05,$06,$05,$09
!byte $05,$05,$05,$05
!byte $05,$05,$05,$05
!byte $0d,$05,$07,$05
!byte $0d,$05,$07,$05
!byte $07,$0d,$07,$0d
!byte $07,$0d,$07,$0d
!byte $0f,$0d,$0f,$07
!byte $0f,$0d,$0f,$07
!byte $0f,$0f,$0f,$0f
!byte $0f,$0f,$0f,$0f
!byte $03,$0f,$0c,$0f
!byte $03,$0f,$0c,$0f
!byte $0c,$03,$0c,$03
!byte $0c,$03,$0c,$03
!byte $03,$00,$0c,$00
!byte $03,$00,$0c,$00
```

```
!align 255,0
```

```
d021tab
```

```
!byte $0c,$0c,$0c,$0c
!byte $0c,$0c,$08,$08
!byte $08,$08,$08,$08
!byte $08,$08,$04,$04
!byte $04,$04,$04,$04
!byte $04,$04,$01,$01
!byte $0f
```

```
d022tab
```

```
!byte $0f,$0f,$00,$00
!byte $00,$00,$00,$00
!byte $00,$00,$00,$00
!byte $00,$00,$00,$00
!byte $00,$00,$03,$03
!byte $03,$03,$03,$03
!byte $0f
```

```
d023tab
```

```
!byte $0b,$0b,$0b,$0b
!byte $0b,$0b,$0b,$0b
!byte $09,$09,$09,$09
!byte $09,$09,$09,$09
!byte $0e,$0e,$0e,$0e
!byte $0e,$0e,$0e
```

```
border2
```

```
!byte $02,$02
```

```
d800tab
```

```
!fill 25,$08
```

```
d800tab_fade
```

```
!byte $0d,$0d,$0d,$0d,$0d
```

```

!byte $0a, $0a, $0a, $0a, $0a, $0a, $0a, $0a
!byte $0e, $0e, $0e, $0e, $0e, $0e, $0e, $0e
!byte $0f, $0f, $0f, $0f

```

rastab

```

!fill 50,$cc

```

rastab\_fade

```

!byte $88, $cc, $cc, $cc, $00, $00, $00, $00
!byte $44, $44, $44, $44, $88, $88, $88, $88
!byte $cc, $cc, $cc, $cc, $00, $00, $00, $44
!byte $44, $44, $44, $88, $88, $88, $88, $cc
!byte $cc, $cc, $cc, $00, $00, $00, $44, $44
!byte $44, $44, $88, $88, $88, $88, $cc, $cc
!byte $cc, $44, $44

```

bcol

```

!byte $02,$02,$02,$02
!byte $0b,$0b,$0b,$0b
!byte $0c,$0c,$0c,$0c
!byte $0f,$0f,$0f,$0f
!byte $01,$01,$01,$01
!byte $0f,$0f,$0f,$0f
!byte $0f,$0f,$0f,$0f
!byte $0f,$0f,$0f,$0f
!byte $0f,$0f,$0f,$0f
!byte $0f,$0f,$0f,$0f

```

col20

```

!byte $02,$02,$02,$02
!byte $08,$08,$08,$08
!byte $0a,$0a,$0a,$0a
!byte $0f,$0f,$0f,$0f
!byte $07,$07,$07,$07
!byte $0f,$0f,$0f,$0f
!byte $0a,$0a,$0a,$0a
!byte $08,$08,$08,$08
!byte $02,$02,$02,$02
!byte $00,$00,$00,$00

```

spcol

```

!byte $02,$02,$02,$02
!byte $02,$02,$02,$02
!byte $08,$08,$08,$08
!byte $0a,$0a,$0a,$0a
!byte $0f,$0f,$0f,$0f
!byte $07,$07,$07,$07
!byte $0f,$0f,$0f,$0f
!byte $0a,$0a,$0a,$0a
!byte $08,$08,$08,$08
!byte $02,$02,$02,$02

```

fade

```

ldy #$00
cpy #$28

```

```
    beq +
    jsr set
    lda bcol,y
    sta border+1
    sta border2+0
    sta border2+1
    lda spcol,y
    sta $d026
    lda #$30
    cmp $d012
    bne *-3
    lda col20,y
    sta $d020
    inc fade+1
+
    rts

set
!for i, 25 {
    !set x = i - 1
    lda rastab_fade + x * 2 + 0
    sta raster + x * 40,y
    lda rastab_fade + x * 2 + 1
    sta raster + x * 40 + $400,y
    lda d800tab_fade + x
    sta $d800 + x * 40,y
}
    rts

fo
    lda #$00
    cmp #$40
    bne +
    jmp fo_end
+
    inc fo+1
    lsr
    lsr
    tay

    lda tab05d,y
    sta c1+1
    lda tab02d,y
    sta c2+1
    lda tab06d,y
    sta c3+1
    lda tab07d,y
    sta c4+1

    ldx #$27-3
```



```
-  
c1      lda #$00  
        sta $d800+3,x  
        sta $d828+3,x  
        sta $d850+3,x  
        sta $d878+3,x  
        sta $d8a0+3,x  
c2      lda #$00  
        sta $d8c8+3,x  
        sta $d8f0+3,x  
        sta $d918+3,x  
        sta $d940+3,x  
        sta $d968+3,x  
        sta $d990+3,x  
        sta $d9b8+3,x  
        sta $d9e0+3,x  
c3      lda #$00  
        sta $da08+3,x  
        sta $da30+3,x  
        sta $da58+3,x  
        sta $da80+3,x  
        sta $daa8+3,x  
        sta $dad0+3,x  
        sta $daf8+3,x  
        sta $db20+3,x  
c4      lda #$00  
        sta $db48+3,x  
        sta $db70+3,x  
        sta $db98+3,x  
        sta $dbc0+3,x  
        dex  
        bpl -  
  
        lda tab0c,y  
        sta d021tab+0  
        sta d021tab+1  
        sta d021tab+2  
        sta d021tab+3  
        sta d021tab+4  
        sta d021tab+5  
        sta border3+1  
        lda tab08,y  
        sta d021tab+6  
        sta d021tab+7  
        sta d021tab+8  
        sta d021tab+9  
        sta d021tab+10  
        sta d021tab+11  
        sta d021tab+12  
        sta d021tab+13  
        lda tab04,y
```

```
sta d021tab+14
sta d021tab+15
sta d021tab+16
sta d021tab+17
sta d021tab+18
sta d021tab+19
sta d021tab+20
sta d021tab+21
lda tab01,y
sta d021tab+22
sta d021tab+23

lda tab03,y
sta d022tab+18
sta d022tab+19
sta d022tab+20
sta d022tab+21
sta d022tab+22
sta d022tab+23

lda tab0b,y
sta d023tab+0
sta d023tab+1
sta d023tab+2
sta d023tab+3
sta d023tab+4
sta d023tab+5
sta d023tab+6
sta d023tab+7
lda tab09,y
sta d023tab+8
sta d023tab+9
sta d023tab+10
sta d023tab+11
sta d023tab+12
sta d023tab+13
sta d023tab+14
sta d023tab+15
lda tab0e,y
sta d023tab+16
sta d023tab+17
sta d023tab+18
sta d023tab+19
sta d023tab+20
sta d023tab+21
sta d023tab+22
lda tab0f,y
sta d022tab+0
sta d022tab+1
sta d021tab+24
```

```

    sta d022tab+24
    sta d023tab+23
    sta d023tab+24
    sta border+1

    lda tab02,y
    sta $d026
fo_end

    lda ra_
    ldx rx_
    ldy ry_
    rti

tab00
    !byte $00,$00,$00,$00,$00,$00,$00,$00
    !byte $00,$00,$00,$00,$00,$00,$00,$00
tab01
    !byte $01,$0f,$0c,$0b,$00,$00,$00,$00
    !byte $00,$00,$00,$00,$00,$00,$00,$00
tab02
    !byte $02,$00,$00,$00,$00,$00,$00,$00
    !byte $00,$00,$00,$00,$00,$00,$00,$00
tab02d
    !byte $0a,$08,$08,$08,$08,$08,$08,$08
    !byte $08,$08,$08,$08,$08,$08,$08,$08
tab03
    !byte $03,$0e,$0c,$06,$00,$00,$00,$00
    !byte $00,$00,$00,$00,$00,$00,$00,$00
tab04
    !byte $04,$06,$00,$00,$00,$00,$00,$00
    !byte $00,$00,$00,$00,$00,$00,$00,$00
tab05
    !byte $05,$0c,$09,$00,$00,$00,$00,$00
    !byte $00,$00,$00,$00,$00,$00,$00,$00
tab05d
    !byte $0d,$0c,$0e,$08,$08,$08,$08,$08
    !byte $08,$08,$08,$08,$08,$08,$08,$08
tab06
    !byte $06,$00,$00,$00,$00,$00,$00,$00
    !byte $00,$00,$00,$00,$00,$00,$00,$00
tab06d
    !byte $0e,$08,$08,$08,$08,$08,$08,$08
    !byte $08,$08,$08,$08,$08,$08,$08,$08
tab07
    !byte $07,$0f,$0a,$08,$09,$00,$00,$00
    !byte $00,$00,$00,$00,$00,$00,$00,$00
tab07d
    !byte $0f,$0b,$0d,$0c,$0a,$08,$08,$08
    !byte $08,$08,$08,$08,$08,$08,$08,$08

```

```
tab08
    !byte $08,$09,$00,$00,$00,$00,$00,$00
    !byte $00,$00,$00,$00,$00,$00,$00,$00
tab09
    !byte $09,$00,$00,$00,$00,$00,$00,$00
    !byte $00,$00,$00,$00,$00,$00,$00,$00
tab0a
    !byte $0a,$08,$09,$00,$00,$00,$00,$00
    !byte $00,$00,$00,$00,$00,$00,$00,$00
tab0b
    !byte $0b,$00,$00,$00,$00,$00,$00,$00
    !byte $00,$00,$00,$00,$00,$00,$00,$00
tab0c
    !byte $0c,$00,$00,$00,$00,$00,$00,$00
    !byte $00,$00,$00,$00,$00,$00,$00,$00
tab0d
    !byte $0d,$03,$05,$0c,$0b,$00,$00,$00
    !byte $00,$00,$00,$00,$00,$00,$00,$00
tab0e
    !byte $0e,$04,$06,$00,$00,$00,$00,$00
    !byte $00,$00,$00,$00,$00,$00,$00,$00
tab0f
    !byte $0f,$0c,$0b,$00,$00,$00,$00,$00
    !byte $00,$00,$00,$00,$00,$00,$00,$00
```

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

Permanent link: [https://codebase64.org/doku.php?id=base:4x4\\_charset\\_fire\\_with\\_lots\\_of\\_colors](https://codebase64.org/doku.php?id=base:4x4_charset_fire_with_lots_of_colors)

Last update: **2015-04-17 04:30**

