

ColorCyclePlasma

Source code for the end part of [You Know The Routine](#)/Camelot. To get the binary bits'n'pieces needed, simply run this part and save it from there. If the music is turned off I think it's only the sine that is really needed to make it work.



```

;-----
;-----
; ColorCyclePlasma
;-----
;-----
; Coded by Cruzer/CML 2002-2004
; Released in "You Know The Routine"/Camelot
; Compiled with mxass
;-----
;-----
; CONCEPT:
; A bunch of different "colorcycler" frames are precalculated as sprites,
; and switched between on each rasterline in realtime, creating a plasmatic
effect...
;-----
;-----
;0200-05ff *
;0600-07ff * pointer-restore
;0800-08ff * basic
;0900-0d7f * generated data
;1000-227f  music
;2280-35ff * shower
;3400-3cff * plasmer
;3d00-3fff  data
;4000-7fff * sprites/pointer-screens
;7fc0-a0ff  main
;a280-abff  logo bitmap
;ac00-afff  logo sprites
;b000-xxxx * mcpixel LUTs
;b800-xxxx * logo screens
;c000-ffff * more pointer-screens
;e000-ffff * more sprites/pointer-screens
;-----
;-----
.la pointerrestore = $0600
.la mod15 =          $0900
.la mod15d018 =     $0a00
.la mul4 =          $0b00
.la line =          $0c00
.la mul3 =          $0d00
.la xpointers1 =    $0e00

```

```
.la xpointers2 =      $0e60
.la xpointers3 =      $0ec0
.la xpointers4 =      $0f20
.la shower =          $2280
.la plasmer =          $3400
.la sine =             $3d00
.la div2mod15d018 =   $3f00 ;*
.la sprites =          $4000
.ba                   $7fc0
.la mcpixels =        $b000

.la border=$d0ff

.la width=96
.la height=63
.la cycles=15

;colors ...

.la purple1 = 0
.la blue1   = 1
.la mixed1  = 2
.la blue2   = 3
.la skummel1= 4
.la green1  = 5
.la brown1  = 6
.la red1    = 7
.la purple2 = 8
.la grey2   = 9
.la black   = 10

;zp-variables:
                ;bytes
.la mcl1=$c0    ;2
.la mcl2=$c2    ;2
.la mcl3=$c4    ;2
.la mcl4=$c6    ;2
.la sprs=$c8    ;2
.la tmp =$d0    ;2
.la cycle=$dd   ;1
.la x=  $de     ;1
.la y=  $df     ;1
.la xps=  $e0    ;8
.la yps=  $e8    ;8

;-----
-----

sei
```

```
lda #$36
sta $01

jsr vicinit

lda #$00
jsr $1000

cli
sei
lda #<irq1
sta $0314
lda #>irq1
sta $0315
asl $d019
lda #$7b
sta $dc0d
lda #$81
sta $d01a
lda #$0b
sta $d011
lda #$3c
sta $d012
cli

jsr fetchfx

jsr makemcpixels
jsr makeluts
;jsr clearsprites
jsr calc

jsr pointerinit
jsr showerinit
jsr plasmerinit
jsr pointerrestoreinit

jsr logocolorupdate

lda #$ff
sta $d015
lda #$00
sta cycle

inc initialized
lda #$02
- cmp initialized
bne -
```

```
    jmp outside

initialized .by 0

;-----
-----
outside

-loop

    lda go
    beq -loop

    jsr plasmerinit

    lda #0
    sta go

    lda calcon
    beq +
    jsr calc
+

    jmp -loop

go .by 0
calcon .by 0
;-----
-----
irq1
    asl $d019

    lda border
    sta rasbuf
    lda #$01
    sta border

    ;jsr $1003

    lda rasbuf
    sta border

    lda initialized
    bne +
    jmp ri
+

    lda #<irq2
    sta $0314
    lda #>irq2
    sta $0315
```

```
asl $d019
lda #$7b
sta $dc0d
lda #$81
sta $d01a
lda #$1b
sta $d011
lda #$36 ;3e
sta $d012

inc initialized

jmp ri
```

```
rasbuf .by $00
```

```
;-----
-----
```

```
irq2
```

```
asl $d019

lda #$0b
sta border
sta border

lda #$38 ;40
sta $d001
sta $d003
sta $d005
sta $d007
sta $d009
sta $d00b
sta $d00d
sta $d00f

lda #$ff
sta $d017

lda #$70
ora $d011
sta $d011

lda #$94
sta $dd00
lda #$ff
sta $d018

ldx colors+0
lda colors+1
ldy colors+2
```

```
sta $d027
sta $d028
sta $d029
sta $d02a
sta $d02b
sta $d02c
sta $d02d
sta $d02e
stx $d025
sty $d026
```

```
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
```

```
ldy #$96
ldx #$94
```

```
lda #$00
sta $d020
sta $d021
```

```
jsr shower
```

```
lda #$c8
sta $d016
;show logo...
```

```
lda $d011
ora #$20
sta $d011
lda #$e8
sta $d018
lda #$95
sta $dd00
; lda #$bc
; sta $d001
```

```
; sta $d003
; sta $d005
; sta $d007
; sta $d009
; sta $d00b
; sta $d00d

lda #$ff
sta $d01d
sta $d01b
lda #$00
sta $d017
sta $d01c

lda #$18
sta $d000
lda #$48
sta $d002
lda #$78
sta $d004
lda #$a8
sta $d006
lda #$d8
sta $d008
lda #$08
sta $d00a
lda #$38
sta $d00c
lda #%01100000
sta $d010
lda logocolors+1
sta $d027
sta $d028
sta $d029
sta $d02a
sta $d02b
sta $d02c
sta $d02d

lda #$03
sta border
jsr pointerrestore

jsr showlogo2

inc border
jsr plasma2

inc border
jsr scroll
```

```
inc border
jsr $1003

inc border
jsr fader
jsr fxchange
jsr logofade

;init sprites for plasma...

lda #$58
sta $d000
lda #$70
sta $d002
lda #$88
sta $d004
lda #$a0
sta $d006
lda #$b8
sta $d008
lda #$d0
sta $d00a
lda #$e8
sta $d00c
lda #$00
sta $d00e
lda #%10000000
sta $d010
lda #$00
sta $d01d
sta $d01b
lda #$ff
sta $d017
sta $d01c

lda #$0f
sta border

jmp ri

cnt .wo 0
logocolors .by $00,$00
;-----
-----
ri
pla
tay
pla
tax
```



```

    pla
    rti
;-----
-----
scroll

    lda scrollx
    sec
    sbc #2
    sta scrollx
    bmi +
    rts
+
    lda #$06
    sta scrollx
    ldx #$00
-   lda $04f1,x
    sta $04f0,x
    lda $04f2,x
    sta $04f1,x
    lda $04f3,x
    sta $04f2,x
    lda $04f4,x
    sta $04f3,x
    inx
    inx
    inx
    inx
    cpx #$28
    bne -
    ldx scrollpnt
sl  lda scrolltext,x
    cmp #$ff
    bne +
    lda #>scrolltext
    sta sl+2
    lda #0
    sta scrollpnt
    lda #" "
+
    tax
    sec
    sbc #$60
    bcs +
    txa
+   sta $0517
    inc scrollpnt
    bne +
    inc sl+2
+
    lda scrollpnt

```

```
    cmp #$33
    beq +
    rts
+
    lda #$01
    ldx #$00
-   sta $d8f0,x
    inx
    cpx #$28
    bne -

    rts
scrollx .by 0
scrollpnt .by 0
;-----
-----

fader

    lda go
    beq +
    rts
+

    ldy fadeptnt+1
    lda mul3,y
    ldx fadeicolor
    cpx #10
    beq +black
    clc
    adc mul24,x
    tax
    lda fadeicolors+0,x
    sta colors+0
    lda fadeicolors+1,x
    sta colors+1
    lda fadeicolors+2,x
    sta colors+2
    jmp +
+black
    lda #$00
    sta colors+0
    sta colors+1
    sta colors+2
+
    lda fadeup
    beq +
    lda fadeptnt
    clc
    adc #$80
    sta fadeptnt
```

```
    lda fadeptnt+1
    adc #$00
    sta fadeptnt+1
    cmp #$07
    bne +
    dec fadeup
+
    lda fadedown
    beq +
    lda fadeptnt
    sec
    sbc #$80
    sta fadeptnt
    lda fadeptnt+1
    sbc #$00
    sta fadeptnt+1
    bne +
    dec fadedown
+
    rts

colors .by 0,0,0

fadeptnt .wo 0
fadeup .by 1
fadedown .by 0
palette .by 0

fadecolor .by 0

fadecolors

    .by $00,$00,$00
    .by $00,$00,$09
    .by $00,$00,$0b
    .by $00,$09,$04
    .by $00,$0b,$0a
    .by $09,$04,$0f
    .by $02,$0a,$07
    .by $04,$0f,$01

    .by $00,$00,$00
    .by $00,$00,$06
    .by $00,$00,$0b
    .by $00,$00,$0e
    .by $00,$06,$0e
    .by $06,$0b,$0f
    .by $0b,$0e,$07
    .by $0e,$03,$01
```

.by \$00,\$00,\$00
.by \$00,\$09,\$02
.by \$09,\$02,\$04
.by \$02,\$04,\$0e
.by \$04,\$0e,\$03
.by \$0e,\$03,\$01
.by \$04,\$0e,\$03
.by \$02,\$04,\$0e

.by \$00,\$00,\$00
.by \$00,\$00,\$06
.by \$00,\$0b,\$0e
.by \$09,\$0c,\$03
.by \$0b,\$0f,\$01
.by \$0c,\$0f,\$01
.by \$0b,\$0c,\$03
.by \$09,\$0b,\$0e

; .by \$00,\$00,\$00
; .by \$00,\$00,\$06
; .by \$00,\$06,\$0e
; .by \$06,\$0e,\$0f
; .by \$0e,\$0f,\$01
; .by \$0f,\$01,\$01
; .by \$0e,\$0f,\$01
; .by \$06,\$0e,\$0f

.by \$00,\$00,\$00
.by \$00,\$0b,\$08
.by \$00,\$0c,\$0a
.by \$06,\$0f,\$07
.by \$0e,\$01,\$01
.by \$03,\$0f,\$07
.by \$0e,\$0c,\$0a
.by \$06,\$0b,\$08

.by \$00,\$00,\$00
.by \$00,\$00,\$0b
.by \$00,\$0b,\$05
.by \$0b,\$05,\$0d
.by \$05,\$03,\$01
.by \$03,\$01,\$01
.by \$05,\$03,\$01
.by \$08,\$05,\$03

.by \$00,\$00,\$00
.by \$00,\$00,\$08
.by \$00,\$08,\$0a
.by \$0b,\$0a,\$0f
.by \$08,\$0f,\$01
.by \$0a,\$0f,\$01

```

.by $08,$0a,$0f
.by $0b,$08,$0a

.by $00,$00,$00
.by $00,$00,$02
.by $00,$02,$0a
.by $00,$0a,$0f
.by $02,$0f,$01
.by $0a,$01,$01
.by $02,$0f,$0f
.by $00,$02,$0a

```

```

.by $00,$00,$00
.by $00,$00,$04
.by $00,$04,$0a
.by $09,$0a,$0f
.by $04,$0f,$01
.by $0a,$0f,$01
.by $04,$0a,$0f
.by $09,$04,$0a

```

```

.by $00,$00,$00
.by $00,$00,$0b
.by $00,$0b,$0c
.by $0b,$0c,$0f
.by $0c,$0f,$01
.by $0f,$01,$01
.by $0c,$0f,$01
.by $0b,$0c,$0f

```

```
mul24 .by 0,24,24*2,24*3,24*4,24*5,24*6,24*7,24*8,24*9,24*10
```

```
;------
```

```
-----
```

```
logofade
```

```
;rts
```

```
lda donefading
```

```
beq +
```

```
rts
```

```
+
```

```
lda logopnt+1
```

```
sec
```

```
sbc #$34
```

```
bcs +
```

```
lda #$00
```

```
+
```

```
asl
```

```
tax
```

```
lda logofadecolors,x
```

```
sta logocolors
```

```
lda logofadecolors+1,x
```

```
sta logocolors+1

lda logopnt
clc
adc #$80
sta logopnt
lda logopnt+1
adc #$00
;and #$07
sta logopnt+1
cmp #$10+$34
bne logocolorupdate
inc donefading
rts
```

logocolorupdate

```
lda logocolors
asl
asl
asl
asl
ldx #$27
-
sta $b828,x
sta $b850,x
sta $b878,x
sta $bca0,x
sta $bcc8,x
sta $bcf0,x
dex
bpl -

rts
```

logopnt .wo 0

logofadecolors

```
.by $00,$00
.by $00,$00
.by $00,$00
.by $00,$00
.by $00,$00
.by $00,$00
.by $00,$06
.by $06,$0b
.by $0b,$0e
.by $0e,$0f
.by $0f,$07
```

```
.by $03,$01
.by $0f,$01
.by $0e,$07
.by $0b,$0f
.by $06,$0e
```

```
donefading .by 0
```

```
;-----
-----
```

```
showlogo2
```

```
;show lower half of logo...
```

```
lda #$00
sta border
```

```
- lda #$d0
  cmp $d012
  bne -
```

```
inc border
```

```
lda #$d1
sta $d001
sta $d003
sta $d005
sta $d007
sta $d009
sta $d00b
sta $d00d
```

```
nop
nop
nop
nop
```

```
lda #$f8
sta $d018
```

```
;jsr showscroll
rts
```

```
;-----
-----
```

```
showscroll
```

```
inc border
lda #$e8
```

```
-   cmp $d012
   bne -

   inc border

   lda #$1b
   sta $d011
   lda #$16
   sta $d018
   lda #$07
   sta $dd00
   lda scrollx
   sta $d016
   rts
```

;------

fxchange

```
   lda go
   beq +
   rts
```

+

```
   lda fxcnt
   sec
   sbc #$01
   sta fxcnt
   lda fxcnt+1
   sbc #$00
   sta fxcnt+1
   bmi +newfx
```

+fadecheck

```
   lda fxcnt+1
   bne +
   lda fxcnt+0
   cmp #$28
   bne +
```

```
   lda #1
   sta fadedown
   lda #0
   sta fadeup
   lda #$00
   sta fadeptnt+0
   lda #$07
   sta fadeptnt+1
```

+

```
   rts
```



```
+newfx
    lda fx
    clc
    adc #1
    cmp numfx
    bne +
    lda #1
+   sta fx

    lda #1
    sta go

fetchfx

;fetch parameters for new fx ...
lda fx
asl
tax
lda fxs,x
sta $fe
lda fxs+1,x
sta $ff
ldy #$00
lda ($fe),y
sta plapa+0
iny
lda ($fe),y
sta plapa+1
iny
lda ($fe),y
sta spread+0
iny
lda ($fe),y
sta spread+1
iny
lda ($fe),y
sta fadeicolor
iny

lda ($fe),y
sta fxcnt
iny
lda ($fe),y
sta fxcnt+1
iny

lda ($fe),y
sta calcon
iny

ldx #$00
```

```
-
    lda ($fe),y
    sta xpa,x
    iny
    inx
    cpx #8
    bne -

    ldx #$00
-
    lda ($fe),y
    sta ypa,x
    iny
    inx
    cpx #8
    bne -

    lda #1
    sta fadeup
    lda #0
    sta fadedown
    lda #$00
    sta fadeplt
    sta fadeplt+1

    rts

fx      .by 0
fxcnt  .wo $0100

numfx   .by 12
fxs     .wo fx0,fx1,fx2,fx2b,fx3,fx4,fx7a,fx5,fx6,fx7,fx8,fx9

;by speed (plasma)
;by spread (plasma)
;by color
;wo duration
;by calc on/off
;wo xpa (cycler)
;wo ypa (cycler)

fx0
    .by 0,0
    .by 0,0
    .by black
    .wo $0133
    .by 1
    .wo $fb80,$fa50,$05ee,$0433
```

```
.wo $0601,$fb30,$05e0,$fa17
```

```
fx1
```

```
.by $02,$01
.by $03,$02
.by brown1
.wo $0240
.by 1
.wo $fb80,$fa50,$05ee,$0433
.wo $0601,$fb30,$05e0,$fa17
```

```
fx2
```

```
.by $fe,$03
.by $ff,$02
.by purple1
.wo $0240
.by 0
.wo 0,0,0,0
.wo 0,0,0,0
```

```
fx2b
```

```
.by $01,$02
.by $fe,$ff
;.by $01,$02
;.by $fe,$01
;.by $02,$ff
;.by $fd,$04
.by green1
.wo $0240
.by 0
.wo 0,0,0,0
.wo 0,0,0,0
```

```
fx3
```

```
.by $fd,$04
.by $ff,$02
.by blue2
.wo $0240
.by 1
.wo $fd80,$fc50,$03ee,$0233
.wo $0401,$fc30,$04e0,$fb17
```

```
fx4
```

```
.by $02,$ff
.by $ff,$02
.by mixed1
.wo $0200
.by 0
.wo 0,0,0,0
.wo 0,0,0,0
```

fx5

```
.by $fd,$02
.by $fe,$03
.by grey2
.wo $0190
.by 0
.wo 0,0,0,0
.wo 0,0,0,0
```

fx6

```
.by $fd,$04
.by $ff,$02
.by purple2
.wo $0200
.by 0
.wo 0,0,0,0
.wo 0,0,0,0
```

fx7a

```
.by $01,$02
.by $ff,$01
.by red1
.wo $0199
.by 0
.wo 0,0,0,0
.wo 0,0,0,0
```

fx7

```
.by $01,$02
.by $ff,$01
.by skummell
.wo $0280
.by 1
.wo $f880,$fa50,$06ee,$0833
.wo $0601,$f730,$05e0,$f817
```

fx8

```
.by $01,$fe
.by $fd,$04
.by blue1
.wo $0240
.by 0
.wo 0,0,0,0
.wo 0,0,0,0
```

fx9

```
.by $02,$01
.by $03,$03
;.by $fe,$03
;.by $ff,$02
.by purple1
.wo $0280
```

```

    .by 0
    .wo 0,0,0,0
    .wo 0,0,0,0

;fx8
;  .by $01,$02
;  .by $ff,$01
;  .by skummell
;  .wo $0400
;  .by 1
;  .wo $ff80,$ff50,$01ee,$0133
;  .wo $0101,$ff30,$01e0,$ff17

;-----
-----

plasma2

    ldx plap+0
    ldy plap+1

    jsr plasmer
    lda plap+0
    clc
    adc plapa+0
    sta plap+0
    lda plap+1
    clc
    adc plapa+1
    sta plap+1
    rts

plap    .by $00,$27
plapa  .by $01,$fe

d018s  .by $0f,$1f,$2f,$3f,$4f,$5f,$6f,$7f,$8f,$9f,$af,$bf,$cf,$df,$ef,$ef
;-----
-----

plasmerinit

    lda #<plasmer
    sta $fe
    lda #>plasmer
    sta $ff

```

```
    ldy #$00
-
    lda plasmersrc1,y
    sta plasmersrc,y
    iny
    cpy #17
    bne -

    ldx #0

-loop
    ldy #0
-
    lda plasmersrc,y
    sta ($fe),y
    iny
    cpy #17
    bne -

    tya
    clc
    adc $fe
    sta $fe
    lda #0
    adc $ff
    sta $ff

    tya
    clc
    adc +ps1+1
    sta +ps1+1
    lda #0
    adc +ps1+2
    sta +ps1+2

    cpx #33
    bne +

    ldy #0
-
    lda jsrshowscroll,y
    sta ($fe),y
    iny
    cpy #3
    bne -

    tya
    clc
    adc $fe
```

```
sta $fe
lda #0
adc $ff
sta $ff
```

```
tya
clc
adc +ps1+1
sta +ps1+1
lda #0
adc +ps1+2
sta +ps1+2
```

+

```
lda #35
clc
adc +ss1+1
sta +ss1+1
lda #0
adc +ss1+2
sta +ss1+2
```

```
lda +sl1+1
clc
adc spread+0
sta +sl1+1
lda +sl2+1
clc
adc spread+1
sta +sl2+1
```

```
inx
cpx #height*2
bne -loop
```

```
ldy #$00
lda #$60
sta ($fe),y
```

```
rts
```

```
spread
.by $ff,$02
```

```
plasmersrc
```

```
+sl1   lda sine,x
        clc
+sl2   adc sine,y
        ror
+ps1   sta plasmer+12
        lda div2mod15d018
+ss1   sta shower+6
```

plasmersrc1

```
    lda sine,x
    clc
    adc sine,y
    ror
    sta plasmer+12
    lda div2mod15d018
    sta shower+6
```

jsrshowscroll

```
    jsr showscroll
```

;------

pointerrestoreinit

```
    lda #<pointerrestore
    sta $fe
    lda #>pointerrestore
    sta $ff
```

-loop

```
    lda +lx+1
    and #$0f
    bne +
    ldy #0
```

-

```
    lda pointerrestoresrc1,y
    sta ($fe),y
    iny
    cpy #5
    bne -
```

```
    jmp +skip
```

+


```
    ldy #0
-   lda pointerrestoresrc2,y
    sta ($fe),y
    iny
    cpy #4
    bne -
```

+skip

```
    tya
    clc
    adc $fe
    sta $fe
    lda #0
    adc $ff
    sta $ff
```

```
    lda +sx+2
    clc
    adc #$04
    sta +sx+2
    sta +sx2+2
    cmp #$7f
    bne +
    lda #$43
    sta +sx+2
    sta +sx2+2
    inc +sx+1
    inc +sx2+1
    inc +lx+1
```

```
    lda +sx+1
    beq +end
```

+ inc +lx+1

```
    jmp -loop
```

+end

```
    ldy #$00
    lda #$60
    sta ($fe),y
```

```
    rts
```

pointerrestoresrc1

```
+lx ldx #$00
+sx stx $43f8
```

```
pointerrestoresrc2
```

```
    inx
+sx2  stx $43f8
```

```
;------
-----
```

```
showerinit
```

```
    lda #<vics
    sta $fc
    lda #>vics
    sta $fd
```

```
    lda #<shower
    sta $fe
    lda #>shower
    sta $ff
```

```
    lda #$00
    sta cnt
```

```
-loop
```

```
    ldy #$00
```

```
-
```

```
    lda showersrc,y
    sta ($fe),y
    iny
    cpy #35
    bne -
```

```
    tya
    clc
    adc $fe
    sta $fe
    lda #$00
    adc $ff
    sta $ff
```

```
    lda +d11+1
    clc
    adc #$01
    and #$07
    ora #$18
```

```
    sta +d11+1

;change vic-updates (d001/pointers)

lda #$ff
sta +v1+3
sta +v2+3
sta +v3+3
sta +v4+3
lda #$d0
sta +v1+4
sta +v2+4
sta +v3+4
sta +v4+4

ldx #$00

-vicloop

ldy #$00
lda ($fc),y
cmp cnt
bne +

iny
lda ($fc),y
sta +v1+1,x
iny
lda ($fc),y
sta +v1+3,x
iny
lda ($fc),y
sta +v1+4,x
lda $fc
clc
adc #$04
sta $fc
lda $fd
adc #$00
sta $fd
inx
inx
inx
inx
inx

jmp -vicloop

+
```

```
    lda #$ff
    sta +dd+1
    lda #$d0
    sta +dd+2
    lda cnt
    cmp #39
    bne +
    lda #$8e
    sta +dd
    lda #$00
    sta +dd+1
    lda #$dd
    sta +dd+2
+
    lda cnt
    cmp #81
    bne +
    lda #$8c
    sta +dd
    lda #$00
    sta +dd+1
    lda #$dd
    sta +dd+2
+
    inc cnt
    lda cnt
    cmp #height*2
    beq +
    jmp -loop
+
    ldy #$00
    lda #$60
    sta ($fe),y

    rts
```

showersrc

```
+d11    lda #$18
        sta $d011
        lda #$00
        sta $d018
+dd     sty $dd00
+v1    lda #$00
        sta $d0ff
+v2    lda #$00
        sta $d0ff
+v3    lda #$00
```

```
    sta $d0ff
+v4 lda #$00
    sta $d0ff
    nop
    nop
```

vics

```
.by $01,$62,$01,$d0
.by $01,$62,$03,$d0
.by $01,$62,$05,$d0
.by $01,$62,$07,$d0
.by $02,$62,$09,$d0
.by $02,$62,$0b,$d0
.by $02,$62,$0d,$d0
.by $02,$62,$0f,$d0

.by $2d,$8c,$01,$d0
.by $2d,$8c,$03,$d0
.by $2d,$8c,$05,$d0
.by $2d,$8c,$07,$d0
.by $2e,$8c,$09,$d0
.by $2e,$8c,$0b,$d0
.by $2e,$8c,$0d,$d0
.by $2e,$8c,$0f,$d0

.by $2f,$80,$f8,$43
.by $2f,$81,$f8,$47
.by $2f,$82,$f8,$4b
.by $2f,$83,$f8,$4f
.by $30,$84,$f8,$53
.by $30,$85,$f8,$57
.by $30,$86,$f8,$5b
.by $30,$87,$f8,$5f
.by $31,$88,$f8,$63
.by $31,$89,$f8,$67
.by $31,$8a,$f8,$6b
.by $31,$8b,$f8,$6f
.by $32,$8c,$f8,$73
.by $32,$8d,$f8,$77
.by $32,$8e,$f8,$7b

.by $33,$90,$f9,$43
.by $33,$91,$f9,$47
.by $33,$92,$f9,$4b
.by $33,$93,$f9,$4f
.by $34,$94,$f9,$53
.by $34,$95,$f9,$57
.by $34,$96,$f9,$5b
.by $34,$97,$f9,$5f
.by $35,$98,$f9,$63
```

.by \$35,\$99,\$f9,\$67
.by \$35,\$9a,\$f9,\$6b
.by \$35,\$9b,\$f9,\$6f
.by \$36,\$9c,\$f9,\$73
.by \$36,\$9d,\$f9,\$77
.by \$36,\$9e,\$f9,\$7b

.by \$37,\$a0,\$fa,\$43
.by \$37,\$a1,\$fa,\$47
.by \$37,\$a2,\$fa,\$4b
.by \$37,\$a3,\$fa,\$4f
.by \$38,\$a4,\$fa,\$53
.by \$38,\$a5,\$fa,\$57
.by \$38,\$a6,\$fa,\$5b
.by \$38,\$a7,\$fa,\$5f
.by \$39,\$a8,\$fa,\$63
.by \$39,\$a9,\$fa,\$67
.by \$39,\$aa,\$fa,\$6b
.by \$39,\$ab,\$fa,\$6f
.by \$3a,\$ac,\$fa,\$73
.by \$3a,\$ad,\$fa,\$77
.by \$3a,\$ae,\$fa,\$7b

.by \$3b,\$b0,\$fb,\$43
.by \$3b,\$b1,\$fb,\$47
.by \$3b,\$b2,\$fb,\$4b
.by \$3b,\$b3,\$fb,\$4f
.by \$3c,\$b4,\$fb,\$53
.by \$3c,\$b5,\$fb,\$57
.by \$3c,\$b6,\$fb,\$5b
.by \$3c,\$b7,\$fb,\$5f
.by \$3d,\$b8,\$fb,\$63
.by \$3d,\$b9,\$fb,\$67
.by \$3d,\$ba,\$fb,\$6b
.by \$3d,\$bb,\$fb,\$6f
.by \$3e,\$bc,\$fb,\$73
.by \$3e,\$bd,\$fb,\$77
.by \$3e,\$be,\$fb,\$7b

.by \$3f,\$c0,\$fc,\$43
.by \$3f,\$c1,\$fc,\$47
.by \$3f,\$c2,\$fc,\$4b
.by \$3f,\$c3,\$fc,\$4f
.by \$40,\$c4,\$fc,\$53
.by \$40,\$c5,\$fc,\$57
.by \$40,\$c6,\$fc,\$5b
.by \$40,\$c7,\$fc,\$5f
.by \$41,\$c8,\$fc,\$63
.by \$41,\$c9,\$fc,\$67
.by \$41,\$ca,\$fc,\$6b
.by \$41,\$cb,\$fc,\$6f

```
.by $42,$cc,$fc,$73
.by $42,$cd,$fc,$77
.by $42,$ce,$fc,$7b

.by $43,$d0,$fd,$43
.by $43,$d1,$fd,$47
.by $43,$d2,$fd,$4b
.by $43,$d3,$fd,$4f
.by $44,$d4,$fd,$53
.by $44,$d5,$fd,$57
.by $44,$d6,$fd,$5b
.by $44,$d7,$fd,$5f
.by $45,$d8,$fd,$63
.by $45,$d9,$fd,$67
.by $45,$da,$fd,$6b
.by $45,$db,$fd,$6f
.by $46,$dc,$fd,$73
.by $46,$dd,$fd,$77
.by $46,$de,$fd,$7b

.by $47,$e0,$fe,$43
.by $47,$e1,$fe,$47
.by $47,$e2,$fe,$4b
.by $47,$e3,$fe,$4f
.by $48,$e4,$fe,$53
.by $48,$e5,$fe,$57
.by $48,$e6,$fe,$5b
.by $48,$e7,$fe,$5f
.by $49,$e8,$fe,$63
.by $49,$e9,$fe,$67
.by $49,$ea,$fe,$6b
.by $49,$eb,$fe,$6f
.by $4a,$ec,$fe,$73
.by $4a,$ed,$fe,$77
.by $4a,$ee,$fe,$7b

.by $4b,$f0,$ff,$43
.by $4b,$f1,$ff,$47
.by $4b,$f2,$ff,$4b
.by $4b,$f3,$ff,$4f
.by $4c,$f4,$ff,$53
.by $4c,$f5,$ff,$57
.by $4c,$f6,$ff,$5b
.by $4c,$f7,$ff,$5f
.by $4d,$f8,$ff,$63
.by $4d,$f9,$ff,$67
.by $4d,$fa,$ff,$6b
.by $4d,$fb,$ff,$6f
.by $4e,$fc,$ff,$73
.by $4e,$fd,$ff,$77
.by $4e,$fe,$ff,$7b
```

```

.by $5a,$bc,$01,$d0
.by $5a,$bc,$03,$d0
.by $5a,$bc,$05,$d0
.by $5a,$bc,$07,$d0
.by $5b,$bc,$09,$d0
.by $5b,$bc,$0b,$d0
.by $5b,$bc,$0d,$d0

.by $00

;-----
-----

pointerinit

;sprite setup:
;
;cycle#00: 00,10,20,30.... [bank 1]
;          80,90,a0,b0.... [bank 3] (bank 1 pnters changed here by show
routine)
;          80,90,a0,b0.... [bank 1] (bank 3 pnters changed here by show
routine)
;cycle#01: 01,11,21,31.... [bank 1]
;          81,91,a1,b1.... [bank 3] ("-")
;          81,91,a1,b1.... [bank 1] ("-")
;cycle#02: 02,12,22,32.... [bank 1]
;          82,92,a2,b2.... [bank 3] ("-")
;          82,92,a2,b2.... [bank 1] ("-")
;...
;cycle#0e 0e,1e...
;
;

lda #$f8
sta $fc
lda #$43
sta $fd

lda #$f8
sta $fe
lda #$c3
sta $ff

lda #$00
sta cycle

-loop

ldy #$00

```



```
    lda cycle
-
    sta ($fc),y
    ldx #$34
    ora #$80
    sei
    stx $01
    sta ($fe),y
    ldx #$36
    stx $01
    cli
    iny
    and #$7f
    clc
    adc #$10
    bpl -

    lda $fd
    clc
    adc #$04
    sta $fd
    ora #$80
    sta $ff
    inc cycle
    lda cycle
    cmp #cycles
    bne -loop

    lda #$ff
    ldy #$07
-
    sta $fff8,y
    dey
    bpl -
    rts

;-----
-----

vicinit

    lda #$00
    sta $d020
    sta $d021

    lda #$ff
    sta $d018
    lda #$96
    sta $dd00
```

```
    lda #$00
    tax
-
    sta $d800,x
    sta $d900,x
    sta $da00,x
    sta $db00,x
    inx
    bne -

    ;set logo spr-pnts...
    ldx #$06
    lda #$b6
-   sta $bbf8,x
    sec
    sbc #$01
    dex
    bpl -
    ldx #$06
    lda #$bd
-   sta $bff8,x
    sec
    sbc #$01
    dex
    bpl -
    rts

;-----
-----
calc

    lda #$00
    ldx #$07
-
    sta xps,x
    sta yps,x
    dex
    bpl -

    lda xpa+0
    sta +a0+1
    lda xpa+1
    sta +a1+1
    lda xpa+2
    sta +a2+1
    lda xpa+3
    sta +a3+1
    lda xpa+4
    sta +a4+1
    lda xpa+5
    sta +a5+1
```

```
    lda xpa+6
    sta +a6+1
    lda xpa+7
    sta +a7+1

    jsr calcxpointers

    lda #0
    sta y

-loop1

    jsr calcline
    lda #cycles-1
    sta cycle

    jsr paintline2
    inc y
    lda y
    cmp #height
    bne -loop1
    rts

;-----
-----
makeluts

    lda #0
    tax
-loop
    sta mod15,x
    tay
    lda d018s,y
    sta mod15d018,x
    tya
    clc
    adc #1
    cmp #15
    bne +
    lda #0
+   inx
    bne -loop

    ldx #0
    ldy #0
-loop
    lda mod15d018,x
    sta div2mod15d018,y
    sta div2mod15d018+1,y
```

```
inx
iny
iny
bne -loop

ldx #0
-loop
txa
asl
asl
sta mul4,x
inx
bne -loop

ldx #$00
-loop
txa
sta tmp
asl
clc
adc tmp
sta mul3,x
inx
bne -loop

rts

;-----
-----
makemcpixels

;make 4 luts for mc-pixels (1 for each mc-x-pos in a byte)
lda #%11000000
sta mask
ldx #>mcpixels
stx smp+2
inx
stx smp+5

-loop2
ldx #$00
ldy #$00
-loop1

lda mcs+1,y
and mask
smp sta mcpixels,x
sta mcpixels+$0100,x
txa
```

```

    cmp mcs+2,y
    bne +
    iny
    iny
+
    inx
    bne -loop1
    inc smp+2
    inc smp+2
    inc smp+5
    inc smp+5
    lsr mask
    lsr mask
    bcc -loop2
    rts

mcs
    .by $00,%00000000
    .by $22,%01010101
    .by $44,%10101010
    .by $66,%11111111
    .by $98,%10101010
    .by $ba,%01010101
    .by $dc,%00000000
    .by $00

;    .by $00,%00000000
;    .by $24,%01010101
;    .by $48,%10101010
;    .by $6c,%11111111
;    .by $90,%10101010
;    .by $b4,%01010101
;    .by $d8,%00000000
;    .by $00

mask .by %11000000
;-----
-----

calcline

;reset x-pointers...

lda xpss+0
sta xps+0
lda xpss+1
sta xps+1
lda xpss+2
sta xps+2
lda xpss+3
sta xps+3

```

```
    lda xpss+4
    sta xps+4
    lda xpss+5
    sta xps+5
    lda xpss+6
    sta xps+6
    lda xpss+7
    sta xps+7

    lda yps+1
    sta +ya1+1
    lda yps+3
    sta +ya3+1
    lda yps+5
    sta +ya5+1
    lda yps+7
    sta +ya7+1

    ldx #width-1

-loop

    ;get x-pointers...

    lda xpointers1,x
    sta sl1+1
    lda xpointers2,x
    sta sl2+1
    lda xpointers3,x
    sta sl3+1
    lda xpointers4,x
    sta sl4+1

    ;add sines ...

+ya1    ldy #1
sl1    lda sine,y
        clc
+ya3    ldy #3
sl2    adc sine,y
        clc
+ya5    ldy #5
sl3    adc sine,y
        clc
+ya7    ldy #7
sl4    adc sine,y
        sta line,x

    dex
    bpl -loop
```

```
;update y pointers...
lda yps+0
clc
adc ypa+0
sta yps+0
lda yps+1
adc ypa+1
sta yps+1
lda yps+2
clc
adc ypa+2
sta yps+2
lda yps+3
adc ypa+3
sta yps+3
lda yps+4
clc
adc ypa+4
sta yps+4
lda yps+5
adc ypa+5
sta yps+5
lda yps+6
clc
adc ypa+6
sta yps+6
lda yps+7
adc ypa+7
sta yps+7

rts
```

calcxpointers

```
;calc all xpointers once, so calcline doesn't have to calc the same
;on each line...

;reset x-pointers...

ldy xpss+0
sty xps+0
lda xpss+1
sta xps+1
lda xpss+2
sta xps+2
lda xpss+3
sta xps+3
lda xpss+4
sta xps+4
lda xpss+5
```

```
    sta xps+5
    lda xpss+6
    sta xps+6
    lda xpss+7
    sta xps+7

    ldx #width-1

-loop

    ;update x pointers...

    tya
    clc
+a0  adc #0
    tay
    lda xps+1
+a1  adc #1
    sta xps+1
    sta xpointers1,x

    lda xps+2
    clc
+a2  adc #2
    sta xps+2
    lda xps+3
+a3  adc #3
    sta xps+3
    sta xpointers2,x

    lda xps+4
    clc
+a4  adc #4
    sta xps+4
    lda xps+5
+a5  adc #5
    sta xps+5
    sta xpointers3,x

    lda xps+6
    clc
+a6  adc #6
    sta xps+6
    lda xps+7
+a7  adc #7
    sta xps+7
    sta xpointers4,x

    dex
    bpl -loop
```



```
rts
```

```
xpss .wo $1000,$4444,$0000,$dddd  
xpa .wo $fd80,$fc50,$03ee,$0233  
ypa .wo $0401,$fc30,$04e0,$fb17
```

```
-----  
-----
```

```
paintline2
```

```
;pixel-convert line to sprites  
;(all cycles)
```

```
;optimized so all 15 cycles are stored for each byte at a time  
;(mcpixel-luts must be $200 long)
```

```
;args:  
;- line ($00-$ff value for each pixel)  
;- y
```

```
;calc sprite target adr...
```

```
ldy y  
ldx mod21,y  
lda mul3,x  
clc  
adc #<sprites  
sta sprs
```

```
ldy y  
ldx div21,y  
lda ssadd,x  
ora #>sprites  
sta sprs+1  
sta tmp
```

```
clc  
lda #>mcpixels  
sta mcl1+1  
adc # $02  
sta mcl2+1  
adc # $02  
sta mcl3+1  
adc # $02  
sta mcl4+1
```

```
ldx #23 ;x=byte_xpos
```

```
-loop

;or 4 pixels together...

ldy mul4,x

lda line+3,y
sta mcl1      ;zp-adrs
lda line+2,y
sta mcl2
lda line+1,y
sta mcl3
lda line+0,y
sta mcl4

ldy #0
lda (mcl1),y
ora (mcl2),y
ora (mcl3),y
ora (mcl4),y
sta (sprs),y

ldy #17
lda (mcl1),y
ora (mcl2),y
ora (mcl3),y
ora (mcl4),y
ldy #$40
sta (sprs),y
ldy #17*2
lda (mcl1),y
ora (mcl2),y
ora (mcl3),y
ora (mcl4),y
ldy #$80
sta (sprs),y
ldy #17*3
lda (mcl1),y
ora (mcl2),y
ora (mcl3),y
ora (mcl4),y
ldy #$c0
sta (sprs),y
inc sprs+1
ldy #17*4
lda (mcl1),y
ora (mcl2),y
ora (mcl3),y
ora (mcl4),y
ldy #$00
sta (sprs),y
```

```
ldy #17*5
lda (mcl1),y
ora (mcl2),y
ora (mcl3),y
ora (mcl4),y
ldy #$40
sta (sprs),y
ldy #17*6
lda (mcl1),y
ora (mcl2),y
ora (mcl3),y
ora (mcl4),y
ldy #$80
sta (sprs),y
ldy #17*7
lda (mcl1),y
ora (mcl2),y
ora (mcl3),y
ora (mcl4),y
ldy #$c0
sta (sprs),y
inc sprs+1
ldy #17*8
lda (mcl1),y
ora (mcl2),y
ora (mcl3),y
ora (mcl4),y
ldy #$00
sta (sprs),y
ldy #17*9
lda (mcl1),y
ora (mcl2),y
ora (mcl3),y
ora (mcl4),y
ldy #$40
sta (sprs),y
ldy #17*10
lda (mcl1),y
ora (mcl2),y
ora (mcl3),y
ora (mcl4),y
ldy #$80
sta (sprs),y
ldy #17*11
lda (mcl1),y
ora (mcl2),y
ora (mcl3),y
ora (mcl4),y
ldy #$c0
sta (sprs),y
inc sprs+1
```



```

    .by 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
    .by 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
;-----
-----

scrolltext
    .text "

    .text "....."
    .text "You've made it to part 5 of 5, which means it's the "The End"
part...  "
    .text "Creditz: Code'n'stuff by Cruzer, except for tunes by "
    .text "Drax, Fanta, JCH and Jeff, and scrolltext in part 4 by Slammer...
"

    .text "Next follows some nerd info about the routines to which you've
been subjected, so you can get to know 'em a bit better...  "
    .text "Disclaimer: I haven't seen all C64 demos ever released, "
    .text "so please excuse me if I make an ass outta myself "
    .text "by taking credit for inventing something which was allready
invented in the 1st place!  "

    .text "OK, with that outta the way, let's start the braggin'... :-)"
    .text "    "

    .text "#01 - Wobbler...  A new gfx-mode is born...  "
    .text "Haven't really settled for a kewl name for it yet, but I guess
you could call it "
    .text "something like "2nd Pixel FLI" or whatever...  "
    .text "It's almost normal multicolor resolution, with each pixel having
its own independent "
    .text "color like in 4x4-mode.  "
    .text "But in 4x4 there's only 4 pixels per char - here there's 16, "
    .text "so it's ofcourse a bit more tricky to make big stuff in this
mode, "
    .text "but I still think it's got some potential...  "

    .text "    "
    .text "#02 - Glenz Each Frame...  "
    .text "Was supposed to be a bit larger, but the bigger version "
    .text "kept fucking up, so after a while I reached the point of agony "
    .text "and decided 2 release this smaller, but working version.  "
    .text "But still it's prolly the fastest filled vector routine ever on
the C64.  "
    .text "The concept is that all y-coords for all lines are precalculated
- "
    .text "and I'm not just talking about the start/end coords, but the y-
values for all pixels "
    .text "in each line.  Sounds like it would fill up way too much for the
scarce memory of the C64, "
    .text "but actually it's packed down to about 8K.  "

```

```
.tex "The trick is that a line pointing in one direction can use the "
.text "same y-coords as other lines with the same slope, even if they're
positioned elsewhere "
.text "and got different lengths. Take a peek at the code, it's pretty
simple actually. "
.text "This means that the linedrawer doesn't have to do any
interpolation - "
.text "it just looks up the y-value of each pixel, and plots it.
Another major "
.text "improvement is that the filler uses immediate addressing mode
(eor #$XX instead of eor $XXXX.) "
.text "This makes the filler 25% faster, but the catch is that the line
drawer must plot the dots "
.text "directly into the filler, which normally would be tricky and take
extra cycles, since "
.text "the y-values hafta be multiplied by 5 to make this work. "
.text "But luckily my linedrawer (aka the "hardliner") allready looks up
the y-values in a table, "
.text "so if the table is pre-multiplied by 5, it works w/o using any
extra cycles. "
.text "Conclusion: This is the way to do filled vector on the C64! :-)"
"

.text "
.text "#03 - FLI Fluffy ... "
.text "The 1st FLI Fluffy ever, AFAIK. "
.text "Guess there's alot of other oldskool dreams "
.text "still waiting to be realised. Just try dreaming your way back to
the late 80s/early 90s, "
.text "and see if you can remember what woulda been considered
impossible dreams on a C64 back then. "
.text "For example "All border realtime texture mapped FLI rubber vector
each frame combined with a DYSP and raster splits" or something :-)"
.text "A good source of inspiration is to take a look at a bunch of
Amiga demos from that period. "
.text "Lots of the routines that were ruling the Amiga scene back then
have ofcourse been C64-ized long ago, "
.text "but there's also quite a few that haven't. "
.text "And don't give up just because there's been a "
.text "lousy C64 version of a routine - that prolly just means there's
lotsa room for improvement... "
.text "
.text "#04 - Copper-plasma ... "
.text "Just some oldskool rasterbender/tech-tech plasma fun. "
.text "Only 2nd line and stuff, but atleast the sines are large, "
.text "like they used to be on the good old Amiga... It coulda been
bigger, if I'd "
.text "bothered to hack the shower, so it didn't waste so much
rastertime. "
.text "It only needs updating each 4th line, and the rest of the cycles
are just pretty much "
```

```
.text "nopped away currently - but if they had been used more
constructively, "
.text "it coulda been about 20% bigger I guess. "

.text "
.text "#05 - Plasma (diz 1) ... "
.text "Only the usual 4 multicolor-colors "
.byte 0
.text " a time, but "
.byte 0
.text "least it looks pretty plasmatic, IMHO. "
.text "And isn't plasmaticity what counts when "
.text "it cums to plasma?  Although I know plasmaticity is not an exact
science, since "
.text "ppls plasmaticity perception is a matter of personal plasma
preferences.... "

.text "Okiez, th"
.by 0
.text " about does it for diz li'l demo... "
.text "But ya never know when it's time for the kR4Zy k4m3Lz 2 bust new
tricks, "
.text "so watch out for future CML prodz!  Cya!..."
.text "
.by 255
```

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

Permanent link:
<https://codebase64.org/doku.php?id=base:colorcycleplasma>

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

