

Julia Fractal Morpher

```
; Julia Morph
; -----
; Entry for the July 1996 NTSC 4k
; demo contest.
;
; Coded by dW/style, July 1, 1996
;
; Notes:
;
; Assembled using turbo assembler
; macro + version
;
; The code assumes there is a music
; routine at $1000(init)/$1003(irq)

ncolors = 16
negative = $80
width = 72
height = 56
yinit1 = 0
yinit2 = (width-4)*2
cset = $3000
cset2 = $2800
initcptr = cset+((width/4)*height)
itercnt = 32 ; max # of iterations
screen = $0400
color = $d800
pointerl = $0800
pointerh = $0900

aux = $02
tmp = $14
frame = $22
iter = $4b
currx = $4c
currxs = $4d
curry = $4e
currys = $4f
real = $50
reals = $51
imag = $52
imags = $53
rsq = $54
isq = $55
xtogo = $56
ytogo = $57
graph = $58
```

```
gdata      = $59
ystor      = $5a
bit9       = $5b
rsq9       = $5c
isq9       = $5d
rseed      = $5e
rseeds     = $5f
iseed      = $60
iseeds     = $61
mapptr     = $62
colptr     = $64
ystor2     = $65
scrptr     = $68
mapptr2    = $6a
gdata2     = $6c

music      = $1000
hwnmi      = $fffa
hwirq      = $fffe

          *= $1800
start
    sei
    lda #$35
    sta $01
    lda #$7f
    sta $dc0d
    sta $dd0d
    lda $dc0d
    lda $dd0d
    lda #$00
    sta $d015
    jsr music
    ldy #$ff
    sty $dc02
    iny
    sty $dc03
    lda #255-2
    sta $dc00
    lda #<irqmain
    sta hwirq
    sta hwnmi
    lda #>irqmain
    sta hwirq+1
    sta hwnmi+1
    lda #<irq0
    sta jmpaddy
    lda #>irq0
    sta jmpaddy+1
    jmp begin
```

```
setint
    bit $d011
    bpl setint
    lda #$1b
    sta $d011
    lda lines
    sta $d012
    lda #$01
    sta $d019
    sta $d01a
    lda #$34
    sta $01
    cli
    rts

irqmain
    sei
    sta storea+1
    stx storex+1
    sty storey+1
    lda #$35
    sta $01
    inc $d019
    jmp (jmpaddy)

irqcont
    lda lines,y
    sta $d012
    lda #$34
    sta $01
    ldx #$00
    lda addyl,y
    sta jmpaddy,x
    inx
    lda addyh,y
    sta jmpaddy,x

storea    lda #$00
storex    ldx #$00
storey    ldy #$00
          cli
          rti

jmpaddy
    .byte 0,0

lines     .byte $26,$89,$fa
addyl     .byte <irq0,<irq1,<irq2
addyh     .byte >irq0,>irq1,>irq2

irq0
    lda #21
```

```
    sta $d018
    lda #27+64
    sta $d011
    lda #0
    sta $d021
    sta $d020
    lda #06
    sta $d022
    lda #0e
    sta $d023
    lda #05
    sta $d024
    lda #200
    sta $d016
    inc $d020
    dec $d020
    ldy #01
    jmp irqcont

irq1    lda #27
        sta $d011
        lda #216
        sta $d016
        nop
        nop
        nop
        lda #01
        sta $d021
        sta $d020
        jsr delay1
colr1   ldx #0e
colr0   ldy #03
        stx $d022
        sty $d023
colr2   lda #06
        ldx #00
        ldy #28
        sta $d021
        stx $d020
        sty $d018
        ldy #02
        jmp irqcont

irq2    jsr delay0
        lda #21
        sta $d018
        lda #01
        sta $d021
        sta $d020
        jsr delay1
```

```
        jsr delay0
        nop
        nop
        nop
        lda #$00
        sta $d021
        sta $d020
        jsr $1003
        bit $dc01
        bmi skcycle
        dec cycle+1
cycle   lda #$08
        bne skcycle
        lda #$08
        sta cycle+1
        ldx colr0+1
        lda colr1+1
        sta colr0+1
        lda colr2+1
        sta colr1+1

colindex ldy #$00
        lda colors,y
        sta colr2+1
        iny
        cpy #ncolors
        bcc storindx
        ldy #$00
storindx sty colindex+1

skcycle ldy #$00
        jmp irqcont

delay0
        nop
        rts

delay1
        ldx #$04
ddell   dex
        bpl ddell
        rts

begin
        bit $d011
        bpl begin
        lda #$0b
        sta $d011
        jsr clearpic
        lda #$00
        sta tmp+1
```

```
        sta frame
        ldy #$07
        lda #$ff
clrchar  sta cset+$07f8,y
        dey
        bpl clrchar

currbox  ldy #$00
        cpy #$02
        clc
        bne skpbox
        sec
skpbox   lda boxtabl1,y
        tax
        lda boxtablh,y
        tay
        jsr drawbox
        inc currbox+1
        lda currbox+1
        cmp #$07
        bne currbox
        lda #$00
        sta currbox+1

        lda #$37
        sta $01
        ldx #<text0
        ldy #>text0
        jsr prnttext
        ldx #<text1
        ldy #>text1
        jsr prnttext
        ldx #<text2
        ldy #>text2
        jsr prnttext
        lda #$35
        sta $01

        lda #147
        sta $dd00
        jsr setint

        ; seed init
frameini
chngsd   lda #0
        sta rseed
        lda #0
        sta rseeds
chngseed lda #94
        sta iseed
        lda #0
```

```
    sta iseeds
    sta currys
    sta curry

    lda #width
    sta xtogo
    lda #height
    sta ytogo

    lda #0
    sta ystor
    lda #<initcptr
    sta mapptr
    sta mapptr2
    lda #(width*2)-8
    sta ystor2
    lda #>initcptr
    sta mapptr+1
    sta mapptr2+1
    lda #$04
    sta graph

rowinit  lda #negative
        sta currxs
        lda #f8
        sta currx

        ; Mandelbrot iteration init
iterinit
        ldy currx
        sty real
        lda sqrtbl,y
        sta rsq
        lda sqrtbl+256,y
        sta rsq9
        lda currxs
        sta reals
        ldy curry
        sty imag
        lda sqrtbl+256,y
        sta isq9
        lda sqrtbl,y
        sta isq
        ldx currys
        stx imags
        ldx #$00
        stx iter
        clc
        lda isq
        adc rsq
        lda isq9
```

```
    adc rsq9
    cmp #$02
    bcc iterate
maxi   lda #$03
    jmp bye3

    ; Main iteration loop
iterate
    inc iter
    lda iter
    and #$03
    cmp #$03
    bne skipc0
    inc iter
skipc0 lda real
    asl a
    tay
    bcc table0
    lda logtbl+256,y
    bcs iter2
table0 lda logtbl,y
iter2  clc
    ldy imag
    adc logtbl,y
    tay
    bcc iter3
    cpy #$11
    bcs enditer
    lda exptbl+256,y
    ldx #$01
    stx bit9
    bne iter4
iter3  lda exptbl,y
    ldx #$00
    stx bit9
iter4  tay
    lda reals
    eor imags
    tax
    stx imags
    eor iseeds
    bne difsign2
    lda bit9
    bne enditer
    clc
    tya
    adc iseed
    bcs enditer
    bcc positv3

difsign2 tya
```



```
        sec
        sbc iseed
        tax
        lda bit9
        sbc #$00
        bmi sgnchnng
        bne enditer
        txa
        jmp positv3
sgnchnng eor #$ff
        sta tmp
        txa
        eor #$ff
        clc
        adc #$01
        tay
        lda #$00
        adc tmp
        bne enditer
        lda #negative
        eor imags
        sta imags
        tya
positv3  sta imag
        jmp iterat2

enditer
enditer2 jmp byebye

iterat2  ldx #$00
        lda rsq
        sec
        sbc isq
        tay
        lda rsq9
        sbc isq9
        bmi sgnchnng2
        sta tmp
        jmp positv1
sgnchnng2 ldx #negative
        eor #$ff
        sta tmp
        tya
        eor #$ff
        clc
        adc #$01
        tay
        lda tmp
        adc #$00
        sta tmp
positv1  stx reals
```

```

    txa
    eor rseeds
    bne difsign
    tya
    clc
    adc rseed
    tay
    lda #$00
    adc tmp
    bne enditer
    beq positv2

difsign  tya
        sec
        sbc rseed
        tay
        lda tmp
        sbc #$00
        bmi sgnch
        bne enditer
        beq positv2

sgnch   eor #$ff
        sta tmp
        tya
        eor #$ff
        adc #$01
        tay
        lda #$00
        adc tmp
        bne enditer
        lda #negative
        eor reals
        sta reals
        .byte $2c

positv2 sta tmp
        tya
        sta real
        ldx iter
        cpx #itercnt
        bcc notyet
        lda #$03
        bcs bye3

notyet  tay
        lda sqrtbl,y
        sta rsq
        lda sqrtbl+256,y
        sta rsq9
        ldy imag
        lda sqrtbl+256,y
        sta isq9
        lda sqrtbl,y

```

```
        sta isq
        adc rsq
        lda isq9
        adc rsq9
        cmp #$02
        bcs bye2
        jmp iterate

bye2
byebye  lda iter
        and #$03
bye3    tax
        asl gdata
        asl gdata
        ora gdata
        sta gdata
        lsr gdata2
        lsr gdata2
        lda bits,x
        ora gdata2
        sta gdata2
        dec graph
        bne skipg
        lda #$04
        sta graph
        ldy ystor
        lda gdata
        sta (mapptr),y
        tya
        clc
        adc #$08
        sta ystor
        ldy ystor2
        lda curry
        beq skipplot
        lda ytogo
        beq skipplot
        lda gdata2
        sta (mapptr2),y
skipplot tya
        sec
        sbc #$08
        sta ystor2
skipg   dec xtogo
        beq endrow
        lda #currx
        ldx #7
        jsr incrmnt
        jmp iterinit
endrow  lda #width
        sta xtogo
```

```

        lda ystor2
        clc
        adc #(width*2)+1
        cmp #(width*2)
        bne skipp1
        lda #(width*2)
        clc
        adc mapptr2
        sta mapptr2
        lda #(width*2)-8
        bcc skipp1
        inc mapptr2+1
skipp1  sta ystor2
        lda ystor
        sec
        sbc #(width*2)+1
        cmp #$ff
        bne skipp
        sec
        lda mapptr
        sbc #(width*2)
        sta mapptr
        lda #$07
        bcs skipp
        dec mapptr+1
skipp   sta ystor
        dec ytogo
        bmi done
        lda #curry
        ldx #4
        jsr incrmnt
        jmp rowinit

done    jmp nextfrm

clearpic ldy #<cset
        sty mapptr
        lda #>cset
        sta mapptr+1
        ldx #$08
        lda #$00
        tay
clrmap  sta (mapptr),y
        iny
        bne clrmap
        inc mapptr+1
        dex
        bne clrmap
        rts

```

```
nextfrm  dec chngseed+1
         dec chngseed+1
         inc chngsd+1
         inc chngsd+1
         lda chngseed+1
         jmp frameini

incrmnt  sta tmp
         ldy #$01
         lda (tmp),y
         bmi neg
         dey
         txa
         clc
         adc (tmp),y
return   sta (tmp),y
         rts
neg      dey
         lda (tmp),y
         stx aux
         sec
         sbc aux
         bcs return
         eor #$ff
         adc #$01
         sta (tmp),y
         tya
         iny
         sta (tmp),y
         rts

copymem  stx mapptr+1
         sty mapptr2+1
         tax
         ldy #$00
         sty mapptr
         sty mapptr2
copy2    lda (mapptr),y
         sta (mapptr2),y
         iny
         bne copy2
         inc mapptr+1
         inc mapptr2+1
         dex
         bne copy2
         rts

prnttext
         stx currx
         sty currx+1
```

```

    ldy #$01
    lda (currx),y
    tax
    dey
    lda (currx),y
    tay
    clc
    jsr $fff0
    lda currx
    clc
    adc #$02
    pha
    lda currx+1
    adc #$00
    tay
    pla
    jsr $able
    rts

drawbox
    .block
    bcc skip1
    lda #$ea          ; NOP
    bcs skip2
skip1  lda #$2c          ; BIT
skip2  sta incrm
       stx mapptr
       sty mapptr+1
       ldy #$05
setopts lda codetabl,y
       tax
       lda (mapptr),y
       sta code,x
       dey
       bpl setopts
       lda #$00
       sta scrptr+1
code   lda #$00          ; y
       asl a
       asl a
       asl a
       sta scrptr
       asl a
       rol scrptr+1
       asl a
       rol scrptr+1
       clc
       adc scrptr
xval  adc #$00          ; x
       sta scrptr
       sta colptr

```

```

        lda #$00
        adc scrptr+1
        pha
        adc #>screen
        sta scrptr+1
        pla
        adc #$d8
        sta colptr+1
fillch  lda #$00          ; char
        sta aux
hval   lda #$00
        sta ytogo
wval   ldy #$00
        sty xtogo
        ldy #$00
fillrow lda aux
        sta (scrptr),y
fillcol lda #$00          ; color
        sta (colptr),y
incrm  .byte $2c          ; BIT
        inc aux
        iny
        dec xtogo
        bne fillrow
        lda scrptr
        clc
        adc #40
        sta scrptr
        sta colptr
        bcc goon
        inc scrptr+1
        inc colptr+1
goon   dec ytogo
        bne wval
        rts

codetabl
        .byte fillch-code+1
        .byte fillcol-code+1
        .byte xval-code+1
        .byte 1
        .byte wval-code+1
        .byte hval-code+1

        .bend

        ; screen box fill presets
        ;
        ; char, color, x, y, w, h
box0   .byte $20,$00,0,0,40,25
box2   .byte 0,$08,11,11,18,14

```

```
box1      .byte $ff,$00,0,11,40,14

box4      .byte 160,0,10,0,19,3
box3      .byte 96,0,11,1,19,3

box6      .byte 224,0,10,5,19,4
box5      .byte 96,0,11,6,19,4

boxtabll  .byte <box0,<box1,<box2
          .byte <box3,<box4,<box5,<box6
boxtablh  .byte >box0,>box1,>box2
          .byte >box3,>box4,>box5,>box6

bits      .byte $00,$40,$80,$c0

          ; x, y, text
text0     .byte 11,1,5,18
          .null "julia set morpher"

text1     .byte 11,6,158
          .text "CODE"
          .byte 186,160,160,160,160,5
          .text "DW"
          .byte 175
          .null "STYLE"

text2     .byte 11,7,158
          .text "MUSIC"
          .byte 186,160,160,5
          .text "MSK"
          .byte 175
          .null "CREST"

          *= $21f0
colors    .byte $0b
          .byte $05
          .byte $0d
          .byte $07
          .byte $0a
          .byte $08
          .byte $02
          .byte $09

          .byte $0b
          .byte $0c
          .byte $0f
          .byte $01
          .byte $03
          .byte $0e
          .byte $06
```



```
.byte $04

; *= $2200

; square table low
sqrtbl .byte 0,0,0,0,0
        .byte 0,0,0,1,1
        .byte 1,1,1,1,2
        .byte 2,2,2,3,3
        .byte 3,3,4,4,5
        .byte 5,5,6,6,7
        .byte 7,8,8,9,9
        .byte 10,10,11,11,12
        .byte 13,13,14,14,15
        .byte 16,17,17,18,19
        .byte 20,20,21,22,23
        .byte 24,25,25,26,27
        .byte 28,29,30,31,32
        .byte 33,34,35,36,37
        .byte 38,39,41,42,43
        .byte 44,45,46,48,49
        .byte 50,51,53,54,55
        .byte 56,58,59,61,62
        .byte 63,65,66,68,69
        .byte 71,72,74,75,77
        .byte 78,80,81,83,85
        .byte 86,88,89,91,93
        .byte 95,96,98,100,102
        .byte 103,105,107,109,111
        .byte 113,114,116,118,120
        .byte 122,124,126,128,130
        .byte 132,134,136,138,140
        .byte 142,145,147,149,151
        .byte 153,155,158,160,162
        .byte 164,167,169,171,173
        .byte 176,178,181,183,185
        .byte 188,190,193,195,198
        .byte 200,203,205,208,210
        .byte 213,215,218,221,223
        .byte 226,228,231,234,237
        .byte 239,242,245,248,250
        .byte 253,0,3,6,9
        .byte 11,14,17,20,23
        .byte 26,29,32,35,38
        .byte 41,44,47,50,53
        .byte 57,60,63,66,69
        .byte 72,76,79,82,85
        .byte 89,92,95,98,102
        .byte 105,109,112,115,119
        .byte 122,126,129,133,136
```



```
.byte 1,1,1,1,1
.byte 1,1,1,1,1
.byte 1,1,1,1,1
.byte 1,1,1,1,1
.byte 1,1,1,1,1
.byte 1,1,1,1,1
.byte 1,1,1,1,1
.byte 1,1,1,1,1
.byte 1,1,1,1,1
.byte 1

; log table
logtbl .byte 0,0,17,27,34
        .byte 39,44,48,51,54
        .byte 56,59,61,63,65
        .byte 66,68,70,71,72
        .byte 74,75,76,77,78
        .byte 79,80,81,82,83
        .byte 83,84,85,86,87
        .byte 87,88,89,89,90
        .byte 91,91,92,92,93
        .byte 93,94,94,95,95
        .byte 96,96,97,97,98
        .byte 98,99,99,100,100
        .byte 100,101,101,102,102
        .byte 102,103,103,104,104
        .byte 104,105,105,105,106
        .byte 106,106,107,107,107
        .byte 108,108,108,108,109
        .byte 109,109,110,110,110
        .byte 110,111,111,111,111
        .byte 112,112,112,112,113
        .byte 113,113,113,114,114
        .byte 114,114,115,115,115
        .byte 115,116,116,116,116
        .byte 116,117,117,117,117
        .byte 117,118,118,118,118
        .byte 118,119,119,119,119
        .byte 119,120,120,120,120
        .byte 120,121,121,121,121
        .byte 121,121,122,122,122
        .byte 122,122,122,123,123
        .byte 123,123,123,123,124
        .byte 124,124,124,124,124
        .byte 125,125,125,125,125
        .byte 125,125,126,126,126
        .byte 126,126,126,126,127
        .byte 127,127,127,127,127
        .byte 127,128,128,128,128
        .byte 128,128,128,128,129
```

```
.byte 129,129,129,129,129
.byte 129,129,130,130,130
.byte 130,130,130,130,130
.byte 131,131,131,131,131
.byte 131,131,131,132,132
.byte 132,132,132,132,132
.byte 132,132,133,133,133
.byte 133,133,133,133,133
.byte 133,134,134,134,134
.byte 134,134,134,134,134
.byte 134,135,135,135,135
.byte 135,135,135,135,135
.byte 135,136,136,136,136
.byte 136,136,136,136,136
.byte 136,137,137,137,137
.byte 137,137,137,137,137
.byte 137,137,138,138,138
.byte 138,138,138,138,138
.byte 138,138,138,139,139
.byte 139,139,139,139,139
.byte 139,139,139,139,139
.byte 140,140,140,140,140
.byte 140,140,140,140,140
.byte 140,140,141,141,141
.byte 141,141,141,141,141
.byte 141,141,141,141,141
.byte 142,142,142,142,142
.byte 142,142,142,142,142
.byte 142,142,142,143,143
.byte 143,143,143,143,143
.byte 143,143,143,143,143
.byte 143,143,144,144,144
.byte 144,144,144,144,144
.byte 144,144,144,144,144
.byte 144,144,145,145,145
.byte 145,145,145,145,145
.byte 145,145,145,145,145
.byte 145,145,146,146,146
.byte 146,146,146,146,146
.byte 146,146,146,146,146
.byte 146,146,147,147,147
.byte 147,147,147,147,147
.byte 147,147,147,147,147
.byte 147,147,147,147,148
.byte 148,148,148,148,148
.byte 148,148,148,148,148
.byte 148,148,148,148,148
.byte 148,149,149,149,149
.byte 149,149,149,149,149
.byte 149,149,149,149,149
.byte 149,149,149,150,150
```

```
.byte 150,150,150,150,150
.byte 150,150,150,150,150
.byte 150,150,150,150,150
.byte 150,150,151,151,151
.byte 151,151,151,151,151
.byte 151,151,151,151,151
.byte 151,151,151,151,151
.byte 151,152,152,152,152
.byte 152,152,152,152,152
.byte 152,152,152,152,152
.byte 152,152,152,152,152
.byte 152,153,153,153,153
.byte 153,153,153,153,153
.byte 153,153

; exp table
exptbl .byte 0,0,0,0,0
        .byte 0,0,0,0,0
        .byte 0,0,0,0,0
        .byte 0,0,0,0,0
        .byte 0,0,0,0,0
        .byte 0,0,0,0,0
        .byte 0,0,0,0,0
        .byte 0,0,0,0,0
        .byte 0,0,0,0,0
        .byte 0,0,0,0,0
        .byte 0,0,0,0,0
        .byte 0,0,0,0,0
        .byte 0,0,0,0,0
        .byte 0,0,0,0,0
        .byte 0,0,0,0,0
        .byte 0,0,0,0,0
        .byte 0,0,0,0,0
        .byte 0,0,0,0,0
        .byte 0,0,0,0,0
        .byte 0,0,0,1,1
        .byte 1,1,1,1,1
        .byte 1,1,1,1,1
        .byte 1,1,1,1,1
        .byte 1,1,1,1,1
        .byte 1,1,1,1,1
        .byte 1,1,1,1,2
        .byte 2,2,2,2,2
        .byte 2,2,2,2,2
        .byte 2,2,3,3,3
        .byte 3,3,3,3,3
        .byte 4,4,4,4,4
        .byte 4,5,5,5,5
        .byte 5,6,6,6,6
        .byte 7,7,7,7,8
        .byte 8,8,9,9,9
```


