Bresenham Circle Routine

Implementation by Scout/Silicon Ltd.

This example is textmode (40×25) only but can easily be converted into any other mode. Enjoy and don't forget to credit me if you use it 😉

I hope those solid circle tunnels show up now in a demo.

```
; Draws a circle using the Bresenham algorithm in textmode
; [C]2006 Scout/Silicon Ltd.

Radius  = 12
CenterX = 20
CenterY = 12
Char    = 160
Cx  = $50
Cy  = $51
Cd  = $52

*=$0810

circle
        lda #Radius
        sta Cy
        asl     ; 2 * radius
        sta CalcD

        lda #0
        sta Cx

        lda #3
        sec
CalcD = *+1
        sbc #$00
        sta Cd

C_loop
        lda Cx
        cmp Cy
        bmi C_Points
        jmp E_rts
C_points

        lda Cx
        sta x1
        sta y2
        sta x4
        sta y7
```

```
        eor #$ff    ; Maak
        clc        ; getal
        adc #1        ; negatief
        sta y3
        sta x5
        sta y6
        sta x8

        lda Cy

        sta y1
        sta x2
        sta x3
        sta y8

        eor #$ff
        clc
        adc #1
        sta y4
        sta y5
        sta x6
        sta x7

y1=*+1
        lda #0            ; y
        clc
        adc #CenterY
        tax
        lda screenlo,x
        sta $f0
        lda screenhi,x
        sta $f1
x1=*+1
        lda #0            ; x
        clc
        adc #CenterX
        tay
        lda #Char
        sta ($f0),y
;-----

y2=*+1
        lda #0            ; x
        clc
        adc #CenterY
        tax
        lda screenlo,x
        sta $f0
        lda screenhi,x
        sta $f1
x2=*+1
```

```
        lda #0            ; y
        clc
        adc #CenterX
        tay
        lda #Char
        sta ($f0),y

;-----

y3=*+1
        lda #0            ; x
        clc
        adc #CenterY
        tax
        lda screenlo,x
        sta $f0
        lda screenhi,x
        sta $f1
x3=*+1
        lda #0            ; y
        clc
        adc #CenterX
        tay
        lda #Char
        sta ($f0),y
;----

y4=*+1
        lda #0            ; x
        clc
        adc #CenterY
        tax
        lda screenlo,x
        sta $f0
        lda screenhi,x
        sta $f1
x4=*+1
        lda #0            ; y
        clc
        adc #CenterX
        tay
        lda #Char
        sta ($f0),y
;----

y5=*+1
        lda #0            ; x
        clc
        adc #CenterY
        tax
        lda screenlo,x
```

```
        sta $f0
        lda screenhi,x
        sta $f1
x5=*+1
        lda #0            ; y
        clc
        adc #CenterX
        tay
        lda #Char
        sta ($f0),y

y6=*+1
        lda #0            ; x
        clc
        adc #CenterY
        tax
        lda screenlo,x
        sta $f0
        lda screenhi,x
        sta $f1
x6=*+1
        lda #0            ; y
        clc
        adc #CenterX
        tay
        lda #Char
        sta ($f0),y

y7=*+1
        lda #0            ; x
        clc
        adc #CenterY
        tax
        lda screenlo,x
        sta $f0
        lda screenhi,x
        sta $f1
x7=*+1
        lda #0            ; y
        clc
        adc #CenterX
        tay
        lda #Char
        sta ($f0),y

y8=*+1
        lda #0            ; x
        clc
        adc #CenterY
        tax
        lda screenlo,x
```

```
        sta $f0
        lda screenhi,x
        sta $f1
x8=*+1
        lda #0            ; y
        clc
        adc #CenterX
        tay
        lda #Char
        sta ($f0),y

        lda Cd
        bpl D_pos    ; d >= 0
        lda Cx
        asl
        asl    ; 4*x
        clc
        adc #6
        adc Cd
        sta Cd
        inc Cx
        jmp C_loop

D_pos
        lda Cx
        sec
        sbc Cy
        asl
        asl    ;4*(x-y)
        clc
        adc #10
        clc
        adc Cd
        sta Cd
        dec Cy

E_next
        inc Cx
        jmp C_loop

E_rts
        lda Cx
        sta xx1
        sta yy2
        sta xx4
        sta yy7

        eor #$ff
        clc
        adc #1
        sta yy3
```

```
        sta xx5
        sta yy6
        sta xx8

        lda Cy

        sta yy1
        sta xx2
        sta xx3
        sta yy8

        eor #$ff
        clc
        adc #1
        sta yy4
        sta yy5
        sta xx6
        sta xx7

yy1=*+1
        lda #0            ; y
        clc
        adc #CenterY
        tax
        lda screenlo,x
        sta $f0
        lda screenhi,x
        sta $f1
xx1=*+1
        lda #0            ; x
        clc
        adc #CenterX
        tay
        lda #Char
        sta ($f0),y
;-----

yy2=*+1
        lda #0            ; x
        clc
        adc #CenterY
        tax
        lda screenlo,x
        sta $f0
        lda screenhi,x
        sta $f1
xx2=*+1
        lda #0            ; y
        clc
        adc #CenterX
        tay
```

```
        lda #Char
        sta ($f0),y

;-----

yy3=*+1
        lda #0          ; x
        clc
        adc #CenterY
        tax
        lda screenlo,x
        sta $f0
        lda screenhi,x
        sta $f1
xx3=*+1
        lda #0          ; y
        clc
        adc #CenterX
        tay
        lda #Char
        sta ($f0),y
;----

yy4=*+1
        lda #0          ; x
        clc
        adc #CenterY
        tax
        lda screenlo,x
        sta $f0
        lda screenhi,x
        sta $f1
xx4=*+1
        lda #0          ; y
        clc
        adc #CenterX
        tay
        lda #Char
        sta ($f0),y
;----

yy5=*+1
        lda #0          ; x
        clc
        adc #CenterY
        tax
        lda screenlo,x
        sta $f0
        lda screenhi,x
        sta $f1
xx5=*+1
```

```
        lda #0            ; y
        clc
        adc #CenterX
        tay
        lda #Char
        sta ($f0),y

yy6=*+1
        lda #0            ; x
        clc
        adc #CenterY
        tax
        lda screenlo,x
        sta $f0
        lda screenhi,x
        sta $f1
xx6=*+1
        lda #0            ; y
        clc
        adc #CenterX
        tay
        lda #Char
        sta ($f0),y

yy7=*+1
        lda #0            ; x
        clc
        adc #CenterY
        tax
        lda screenlo,x
        sta $f0
        lda screenhi,x
        sta $f1
xx7=*+1
        lda #0            ; y
        clc
        adc #CenterX
        tay
        lda #Char
        sta ($f0),y

yy8=*+1
        lda #0            ; x
        clc
        adc #CenterY
        tax
        lda screenlo,x
        sta $f0
        lda screenhi,x
        sta $f1
xx8=*+1
```

```
        lda #0            ; y
        clc
        adc #CenterX
        tay
        lda #Char
        sta ($f0),y

        rts

.align 256

screenlo    ; you could also get this from kernal address $EFCO, if kernal
is enabled.

        .byte <$0400+(40*0),
<$0400+(40*1),<$0400+(40*2),<$0400+(40*3),<$0400+(40*4),<$0400+(40*5)
        .byte <$0400+(40*6),
<$0400+(40*7),<$0400+(40*8),<$0400+(40*9),<$0400+(40*10),<$0400+(40*11)
        .byte <$0400+(40*12),
<$0400+(40*13),<$0400+(40*14),<$0400+(40*15),<$0400+(40*16),<$0400+(40*17)
        .byte <$0400+(40*18),
<$0400+(40*19),<$0400+(40*20),<$0400+(40*21),<$0400+(40*22),<$0400+(40*23)
        .byte <$0400+(40*24)
screenhi
        .byte >$0400+(40*0),
>$0400+(40*1),>$0400+(40*2),>$0400+(40*3),>$0400+(40*4),>$0400+(40*5)
        .byte >$0400+(40*6),
>$0400+(40*7),>$0400+(40*8),>$0400+(40*9),>$0400+(40*10),>$0400+(40*11)
        .byte >$0400+(40*12),
>$0400+(40*13),>$0400+(40*14),>$0400+(40*15),>$0400+(40*16),>$0400+(40*17)
        .byte >$0400+(40*18),
>$0400+(40*19),>$0400+(40*20),>$0400+(40*21),>$0400+(40*22),>$0400+(40*23)
        .byte >$0400+(40*24)
```