

TRACÉS DE CERCLES SUR μ C

Nous avons abordé le mois dernier le tracé de droites en utilisant l'algorithme de

Bresenham. Ce dernier a le mérite de ne nécessiter que des calculs utilisant

des nombres entiers et est donc plus rapide que ceux qui utilisent des nombres

flottants. Bresenham a élaboré un algorithme similaire pour le tracé des cercles

que nous allons détailler.

Considérons un cercle de centre $(0, 0)$ et de rayon R . Nous ne nous intéresserons qu'au premier quadrant ($X \geq 0, Y \geq 0$). On peut facilement déduire les trois autres quadrants grâce aux symétries suivant les axes des X et des Y . L'algorithme de Bresenham est un algorithme itératif qui calcule les nouvelles coordonnées X_i, Y_i du nouveau point à tracer en fonction des positions du point précédent. Les observations suivantes simplifient la conception de l'algorithme : l'algorithme commence par le point $Y = R, X = 0$,

Y est une fonction monotone décroissante de X .

Pour tout point x_i, y_i du cercle, le point suivant généré ne peut avoir que l'une des trois positions suivantes :

immédiatement à droite aux coordonnées $x_i + 1, y_i$, diagonalement aux coordonnées $x_i + 1, y_i - 1$, immédiatement en dessous aux coordonnées $x_i, y_i - 1$.

Le point à choisir doit minimiser les distances (ou les carrés de distances) entre une de ces trois positions et le cercle réel soit :

$$mh = |(x_i + 1)^2 + (y_i)^2 - R^2|$$

$$md = |(x_i + 1)^2 + (y_i - 1)^2 - R^2|$$

$$mv = |(x_i)^2 + (y_i - 1)^2 - R^2|$$

Soit $\Delta_i = (x_i + 1)^2 + (y_i - 1)^2 - R^2$, la différence entre le carré de la distance depuis le centre du cercle au pixel diagonal $(x_i + 1), (y_i - 1)$ et la distance à un point du cercle.

Si $\Delta_i < 0$, le point diagonal est à l'intérieur du cercle, aussi le point choisi doit être le point à droite (mh) ou le point vertical (mv). En effet, le point vertical (mv) serait encore plus éloigné à l'intérieur du cercle. Pour choisir le point, il faut calculer la différence entre mh et md .

$$\delta = mh - md$$

$$\delta = |(x_i + 1)^2 + (y_i)^2 - R^2|$$

$$- |(x_i + 1)^2 + (y_i - 1)^2 - R^2|$$

Si $\delta < 0$ la distance du cercle au pixel diagonal (md) est plus grande que la distance au point horizontal (mh).

Aussi, si :

$\Delta_i < 0$ et $\delta \leq 0$, choisir mh

$\Delta_i < 0$ et $\delta > 0$, choisir md

```
; TRACE DE CERCLE. Algorithme de Bresenham
; Trace le cercle de Rayon R4 et de centre X = R6 et Y = R7 dans un espace
; où X et Y sont compris entre 0 et 255.
; Le rayon maximum du cercle est de 255. La fonction PLOT duplique le point
; calculé par CERCLE suivant des axes de symétries passant par le centre du
; cercle.
; Les quatres points ainsi calculés sont tracés par la fonction PIXEL.
; La fonction PLOT réalise du «clipping» si les points calculés sortent de
; l'espace (les points qui sortent de l'espace affichable ne sont pas affichés)
; Si l'afficheur LCD a une résolution inférieure à 256 X 256, il faut modifier
; PLOT de manière à ce que le clipping fonctionne correctement.
;
SEGMENT_DATA SEGMENT DATA
SEGMENT_CODE SEGMENT CODE
    RSEG SEGMENT_DATA
        ; Utilisé par l'algorithme de tracé de cercle pour un quadrant
        XI: DS 1 ; pixel dans le premier quadrant
        YI: DS 1
        DELTA : DS 2 ; carré de l'erreur
        ; Utilisé pour tracer le cercle complet
        CENTREX: DS 1 ; centre du cercle
        CENTREY: DS 1
        MSB EQU 0
        LSB EQU 1
        RSEG SEGMENT_CODE
        ; PIXEL doit être écrit en fonction de l'afficheur graphique
        ; X est passé dans R6, Y dans R7.
        PIXEL: RET
        ; Affiche les 4 points symétriques. Les axes de symétrie passent par le
        ; centre du cercle (CENTREX, CENTREY).
PLOT: MOV A,CENTREX ; position du centre
      ADD A,XI ; translate la position X du cercle
      JC PLOT2 ; dépasse le bord, sort
      MOV R6,A
      MOV A,CENTREY
      ADD A,YI ; idem pour Y
      JC PLOT3 ; n'essaie même pas le deuxième
      MOV R7,A
      ACALL PIXEL ; trace le premier quadrant
      ; deuxième quadrant -X, Y
PLOT2: CLR C ; position du centre
      MOV A,CENTREX ; translate la position X du cercle
      SUBB A,XI ; dépasse le bord, essai quatrième
      JC PLOT4 ; dépasse le bord, sort
      MOV R6,A
      MOV A,CENTREY
      ADD A,YI ; Y idem premier quadrant
      JC PLOT3 ; dépasse essaie le 3ième quadrant
      MOV R7,A
      ACALL PIXEL ; trace le deuxième quadrant
      ; troisième quadrant -X, -Y
PLOT3: CLR C ; position du centre
      MOV A,CENTREX ; translate la position X du cercle
      SUBB A,XI ; dépasse le bord, sort
      JC PLOT4 ; dépasse le bord, sort
      MOV R6,A
      CLR C
      MOV A,CENTREY
      SUBB A,YI ; Y dépasse, sort
      JC PLOT4 ; trace le troisième quadrant
      MOV R7,A
      ACALL PIXEL ; trace le troisième quadrant
      ; quatrième quadrant X, -Y
PLOT4: MOV A,CENTREX ; position du centre
      ADD A,XI ; translate la position X du cercle
      JC PLOTS ; dépasse le bord, sort
      MOV R6,A
      MOV A,CENTREY
      SUBB A,YI ; Y idem troisième quadrant
      JC PLOTS ; dépasse le bord, sort
      MOV R7,A
      ACALL PIXEL ; trace le quatrième quadrant
      ; Transfère DELTA dans R6:R7
PLOTS: RET
      ; GETERR: MOV A,DELTA+MSB ; DELTA+MSB
      ; MOV R6,A
      ; MOV A,DELTA+LSB ; DELTA+LSB
      ; MOV R7,A
      ; RET
```

suite page suivante

Comme le point diagonal est à l'intérieur du cercle, le calcul de δ se simplifie :

$$\begin{aligned}\delta &= (xi + 1)^2 + (yi)^2 - R^2 \\ - (xi + 1)^2 + (yi - 1)^2 - R^2 \\ \delta &= 2 [(xi + 1)^2 + (yi - 1)^2 - R^2] + 2 yi - 1 \\ \delta &= 2 (\Delta_i + yi) - 1\end{aligned}$$

Un raisonnement similaire pour le cas $\Delta_i > 0$ aboutit à un calcul de δ' suivant :

$$\delta' = 2 (\Delta_i - xi) - 1$$

$\Delta_i > 0$ et $\delta' \leq 0$, choisir md

$\Delta_i > 0$ et $\delta' > 0$, choisir mv

$\Delta_i = 0$ choisir md

Le calcul de Δ_i peut être effectué de manière récursive pour les trois cas possibles. Si le nouveau point est le point horizontal (mh) :

$$xi+1 = xi + 1$$

$$yi+1 = yi$$

$$\Delta_{i+1} = (xi+1 + 1)^2 + (yi+1 - 1)^2 - R^2$$

$$\Delta_{i+1} = \Delta_i + 2xi+1 + 1$$

De même si le nouveau point est le point diagonal :

$$\Delta_{i+1} = \Delta_i + 2xi+1 - 2yi+1 + 1$$

Si c'est le point vertical :

$$\Delta_{i+1} = \Delta_i - 2yi+1 + 1$$

L'algorithme de Bresenham complet pour le tracé de cercle dans le premier quadrant s'écrit :

$$xi = 0$$

$$yi = R$$

$$\Delta_i = 2(1-R)$$

boucle :

Plot (xi, yi)

si $yi \leq 0$ va à fin

si $\Delta_i < 0$ alors va à intérieur

si $\Delta_i > 0$ alors va à extérieur

si $\Delta_i = 0$ alors va à diagonal

intérieur :

$$\delta = 2 (\Delta_i + yi) - 1$$

si $\delta \leq 0$ alors va à horizontal

si $\delta > 0$ alors va à diagonal

extérieur :

$$\delta' = 2 (\Delta_i - xi) - 1$$

si $\delta' \leq 0$ alors va à diagonal

si $\delta' > 0$ alors va à vertical

horizontal :

$$xi = xi + 1$$

$$\Delta_i = \Delta_i + 2xi+1$$

va à boucle

diagonal :

$$xi = xi + 1$$

$$yi = yi - 1$$

$$\Delta_i = \Delta_i + 2xi - 2yi + 1$$

va à boucle

vertical :

$$yi = yi - 1$$

$$\Delta_i = \Delta_i - 2 yi + 1$$

va à boucle

fin :

Le listing 1 reprend l'algorithme ci-dessus traduit en assembleur 8051. La fonction CERCLE calcule les points Xi et Yi correspondant au cercle de rayon R4 dans le premier quadrant. CERCLE appelle la fonction PLOT qui trace les points correspondants aux quatre quadrants du cercle du centre CENTREX et CENTREY. Pour faciliter les calculs, l'opposé de Δ_i est calculé à la place de Δ_i . Les signes de tests sont inversés par rapport à la description de l'algorithme. La fonction PLOT appelle la fonction PIXEL qui trace le point sur l'organe graphique choisi. Cette fonction doit être écrite par l'utilisateur. Le rayon doit être supérieur à zéro et inférieur à 256. Le listing est disponible sur le serveur ERP.

```

; R6:R7 = R6:R7 - R4:R5
SUBINT: CLR C
    MOV A,R7
    SUBB A,R5
    MOV R7,A
    MOV A,R6
    SUBB A,R4
    MOV R6,A
    RET

; R6:R7 = R6:R7 + R4:R5
ADDINT: MOV A,R7
    ADD A,R5
    MOV R7,A
    ADDC A,R6
    MOV R6,A
    RET

; R4:R5 = R6:R7
TFRINT: MOV A,R7
    MOV R5,A
    MOV A,R6
    MOV R4,A
    RET

; R6:R7 = 2 * (R6:R7)
DBLINT: MOV A,R7
    ADD A,R7
    MOV R7,A
    ADDC A,R6
    MOV R6,A
    RET

; Etend l'addition 8 bits sur 16 bits dans R4:R5
EXTADD: MOV R5,A ; sauve le LSB
    CLR A ; étend sur 16 bits
    ADDDC A,#0 ; sauve le MSB
    MOV R4,A
    RET

; CERCLE utilise l'algorithme de Bresenham pour tracer un cercle.
; CERCLE ne trace que les points du premier quadrant. Le centre du cercle est
; X = 0, Y = 0. CERCLE est initialisé avec Y = Rayon, X = 0 et s'arrête quand
; le premier quadrant est tracé, c'est à dire quand Y = 0. Le rayon maximum
; est de 255.
; CERCLE appelle la fonction PLOT qui trace les quatre points symétriques pour
; avoir un cercle complet. PLOT utilise les paramètres XI et YI qui sont la
; position calculée de l'arc de cercle pour le premier quadrant.
; CERCLE modifie A, R4, R5, R6, R7. (R0, R1, R2, R3 inchangés)
CERCLE: MOV YI,R4 ; YI = Rayon
    DEC R4 ; R - 1
    MOV A,R4
    ADD A,R4
    MOV DELTA+LSB,A
    CLR A
    MOV XI,A ; XI = 0
    ADDDC A,#0
    MOV DELTA+MSB,A
    ACALL CER1 ; ACALL PLOT ; trace XI, YI et les points symétriques
    PLOT
    MOV A,YI ; si YI = 0, fin
    JZ CER50
    MOV A,DELTA+MSB
    ORL A,DELTA+LSB
    IZ CER20
    ACALL GETERR
    ACALL DBLINT
    IB ACC,7,CER3
    ; signe > 0, calcule 2*DELTA - (2*YI - 1)
CER2: MOV A,YI ; R6:R7 = DELTA
    DEC A ; 2*DELTA, retour dans A le MSB
    ADD A,YI ; teste le signe de 2*DELTA
    ACALL EXTADD ; YI
    ACALL SUBINT ; YI - 1
    JNB ACC,7,CER10 ; 2*YI - 1 (YI toujours >= 1)
    AJMP CER20 ; étend sur 16 bits dans R4:R5
    ; signe < 0, calcule 2*DELTA + 2*XI + 1
CER3: MOV A,XI ; 2*XI
    INC A ; 2*XI + 1 pas de retenue (2*XI pair)
    ACALL EXTADD ; étend sur 16 bits dans R4:R5
    ACALL ADDINT; 2*DELTA + 2*XI + 1
    JNB ACC,7,CER20 ; >= 0
    AJMP CER30 ; < 0
    ; calcule le nouveau pixel
CER10: INC XI ; 2*XI
    MOV A,XI ; calcule Di = Di - (2*XI + 1)
    ADD A,XI ; 2*XI
    INC A ; 2*XI + 1
    ACALL EXTADD ; étend sur 16 bits dans R4:R5
    AJMP CER40 ; va soustraire de DELTA
    ; DEPLACEMENT HORIZONTAL
CER20: INC XI ; XI - YI
    MOV A,XI ; sauve le LSB dans le MSB
    CLR C
    MOV A,XI
    SUBB A,YI
    MOV R6,A
    CLR A
    SUBB A,#0
    XCH A,R6
    ADD A,#1
    MOV R7,A
    MOV A,R6
    ADDDC A,#0
    MOV R6,A
    ACALL DBLINT ; sauve le MSB et récupère le LSB
    ACALL TFRINT ; XI - YI + 1
    AJMP CER40 ; étend le résultat sur 16 bits
    ; DEPLACEMENT EN BIAIS
CER30: DEC YI ; dans R6:R7
    JZ CER31 ; 2*(XI - YI + 1)
    DEC A ; dans R4:R5
    ADD A,YI ; dans R4:R5
    ACALL EXTADD ; va soustraire de DELTA
    ACALL GETERR ; DEPLACEMENT VERTICAL
    ACALL ADDINT ; si YI = 0...
    AJMP CER41 ; YI - 1 positif ou nul
    ; étend sur 16 bits dans R4:R5
    ; prend DELTA
    ; additionne à DELTA
    ; et transfère
    ; R4:R5 = 1
    CER31: MOV R4,A ; R6:R7 = DELTA
    INC A ; DELTA = DELTA - R4:R5
    MOV R5,A ; va boucler jusqu'à YI = 0 (et RI = RAYON)
    ACALL GETERR
    ACALL SUBINT
    CER41: MOV DELTA+MSB,A
    MOV A,R7
    MOV DELTA + LSB,A
    AJMP CER1
    CER50: RET
    END

```

■ Listing 1.