JAPLA 研究会資料 2018/5/19

２つの円の交点座標と弓形図形のグラフィックスーその３
ーマウス・クリックで画面上、どこにでも現れるー

西川　利男

Ｊを用いたアルハンブラ・グラフィックスを目指すものの、まだいっこうに、準備段階でモタモタしている。
先に２つの円の重なり合いのグラフィックスを報告したが[1]、ウィンドウズ・グラフィックスの処理であるので、マウスを活用してもう少し「人にやさしいグラフィックス」にできないか、とプログラミングを行ってみた。
[1] 西川利男「２つの円の交点座標と弓形図形のグラフィックスーその２
Ｊと非線形連立方程式」JAPLA 研究会資料 2018/3/10

ところで、Ｊの isigraphics では、画面上でマウスのクリックにより画面上の位置の情報を得ることができる。筆者はかなり、以前に move_sin.ijs なるプログラムとして、発表したことがある。
[2] 西川利男　Ｊプログラム「Moving sin_curve driven by the mouse
T. Nishikawa　2001/3/12」

今回のマウスによる画面上の表示の実行は、この手法を利用したものである。

１．マウス・クリックによる情報

isigraph の control events として、いろいろな情報が返される。
そのうち mouse については
mmove, mbldwon, mblup, mbrdown, mbrup などがサポートされている。
例えば、グラフィック画面のある位置で、マウスの左ボタンを押したときには、wdhandler 変数 sysdata として、次のような値が返される。

```
sysdata
118 227 358 348 1 0 0 0
```

この値の意味は

```
x y width height leftbutton rightbutton ctrl shift
```

である。
マウスを別の場所に動かしクリックすれば、次のように変わる。

```
sysdata
86 132 358 348 1 0 0 0
```

このようにして、マウスの位置を知ることができる。

Ｊの上の機能を利用して、「人にやさしいグラフィックス」を実現した。

２．エディット・ボックスからのパラメータ入力によるグラフィックス

先のプログラム[1]のフォームをそのまま利用して、まずエディット・ボックスから２つの円のパラメータを入力する。

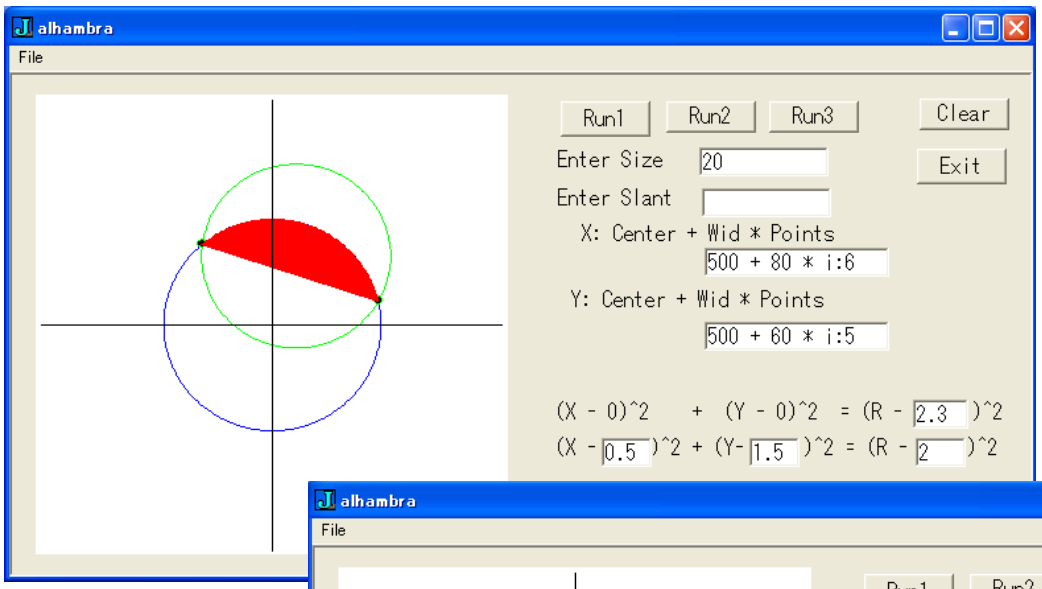
円１　中心位置のＸ座標、Ｙ座標、半径　中心は原点とする、半径は任意

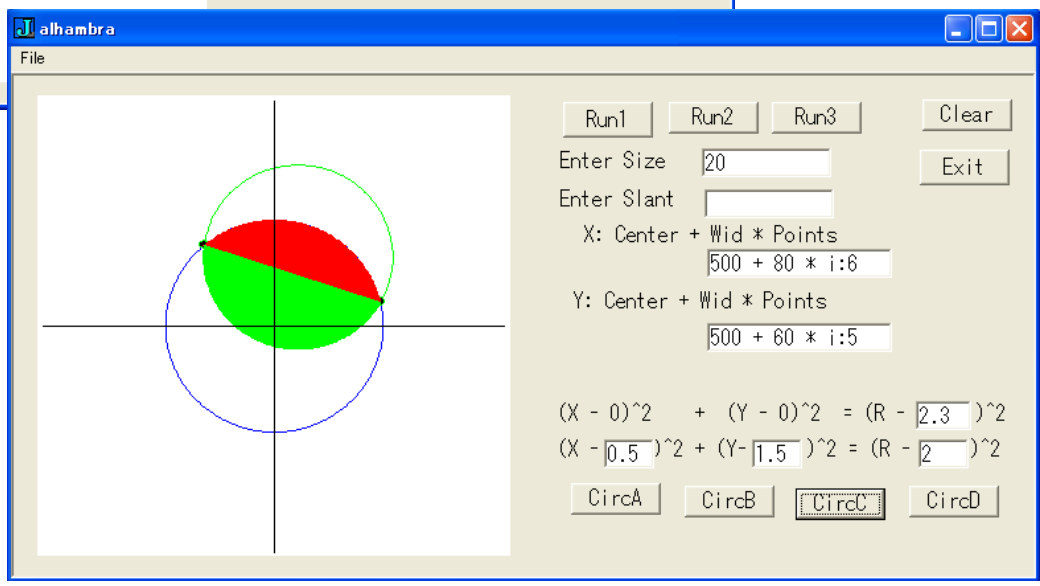
円２　中心位置のＸ座標、Ｙ座標、半径　中心は任意の位置、半径は任意

次にボタンCircA, CircB, CircCを順次押すと、座標軸、円１、円２の順で入力した値に応じて、２つの円の重なった、弓形図形が表示される。
当然、重なりがないときは、円だけが表示される。

ボタンCircDでは、すべてが一度におこなわれる。

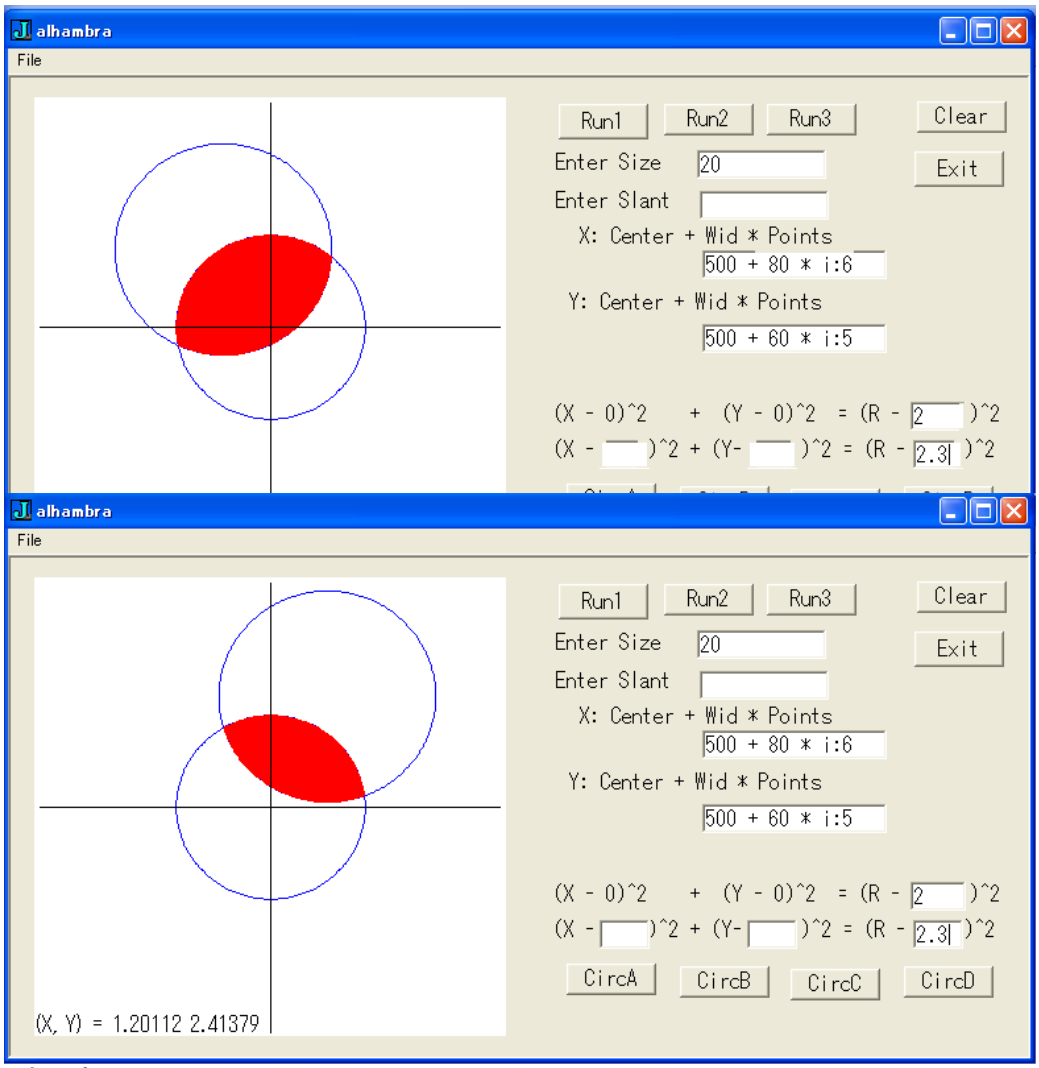
実行の例は次のようになる。　Ｊのプログラムは最後にあげる。
なお、重なりがあるなしのテストは、計算して交点座標の値のが実か虚で判定した。





３．マウス・クリックによるグラフィックス

２つの円、それぞれの半径の値をエディット・ボックスへ入力する。
グラフィック画面の上で、マウスの左ボタンを押すとその位置を円２の中心の位置として、２つの円の重なり図形がただちに表示される。
なお、マウスをドラッグすれば、それに応じて図形は動き、マウスボタンを離せば最後の位置で表示される。



マウスを動かして、位置を変えてみる。

のプログラム・リスト

```
NB. find real or complex
re_im =: 0&=@({:@+.)

adarctan =: 3 : 0
'ax ay' =. y.
reim =. (re_im ax) * (re_im ay)
NB. wr 'ax = ', ": ax, reim
NB. wr 'ay = ', ": ay, reim
if. 0 = reim
 do. z =. 0 return.
 else.
     z =. | darctan (ay % ax)
     if. (ax > 0) *. (ay > 0) do. z end.
     if. (ax < 0) *. (ay > 0) do. z =. 180 - z end.
```

```
     if. (ax < 0) *. (ay < 0) do. z =. 180 + z end.
     if. (ax > 0) *. (ay < 0) do. z =. 360 - z end.

end.
)

alhambra_CircD_button=: 3 : 0

glrgb 0 0 255
glpen 2 0

XA =. 0
YA =. 0
NB. RA, XB, YB, RB are set from entering data on the input edit boxed.

gncircle_empty (circ2_adjXY XA), (circ2_adjXY YA), (circ2_adjR RA)

gncircle_empty (circ2_adjXY XB), (circ2_adjXY YB), (circ2_adjR RB)

glrgb 255 0 0
glpen 4 0
glbrush ''

'C4PX1 C4PY1 C4PX2 C4PY2' =: circross RA, RB, XB, YB  NB. global value
if. 0 = adarctan (C4PX1, C4PY1) do. goto_skip. end.

alpha =: adarctan (C4PX1, C4PY1)
beta  =: adarctan (C4PX2 , C4PY2)
if. ((beta - alpha) < 0) do. beta =: beta + 360 end.
(alpha, beta) gnarc (circ2_adjXY 0), (circ2_adjXY 0),  (circ2_adjR RA)

gamma =: adarctan ((C4PX2 - XB), (C4PY2 - YB))
delta  =: adarctan ((C4PX1 - XB) , (C4PY1 - YB) )
if. ((delta - gamma) < 0) do. delta =: delta + 360 end.
(gamma, delta) gnarc (circ2_adjXY XB), (circ2_adjXY YB), (circ2_adjR RB)

label_skip.
glrgb 0 0 0
glpen 1 0
gllines 10 500 990 500
gllines 500 10 500 990
glshow ''
)

alhambra_Alham_mbldown =: 3 : 0
d =. ". sysdata
x =: (0{d) * 1000 % (2{d)
y =: (1{d) * 1000 % (3{d)
)

alhambra_Alham_mmove =: 3 : 0
d =. ". sysdata
if.-. 4{d do. return. end.
x =: (0{d) * 1000 % (2{d)
y =: (1{d) * 1000 % (3{d)

XX =. 10 * (x - 500) % 1000
YY =. 10 * (y - 500) % 1000
NB. wr XX, YY
```

```
gltextalign TA_BOTTOM
gltext '(X, Y) = ', '': XX, YY
glshow ''

glclear ''
alhambra_CircD_button ''

NB. wr XX, YY
gltextalign TA_BOTTOM
gltext '(X, Y) = ', '': XX, YY
glshow ''
)
```