

糸掛け曼荼羅と J806 の gl2 グラフィックス

SHIMURA Masato

2018 年 6 月 20 日

目次

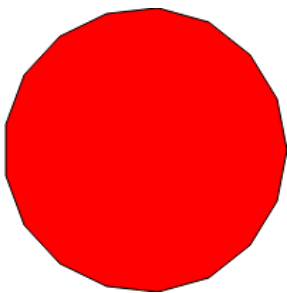
1	グラフィックスの練習	1
2	gl2 版の糸掛け曼荼羅を	3

1 グラフィックスの練習

1.1 簡易グラフィックス

J は基本機能の練習用に簡易グラフィックスを用意している
ガウスの 17 角形を作図してみよう

```
require 'graph'  
RED gdpolygon gddraw ; +. r. 2p1 * (i.17)% 17
```



1.2 JQT の本格グラフィックス

J8QT 版も gl2 をサポートしているがフォームエディタは無くなった。
一般にグラフィックスはデリケートだが、J-QT 版は特別だと感じる。

確実なコードの抽出」 とにかく *HelpStudio/QTdemos* や *addons/demos/qt demos* などから

「絶対確実」なコードを抽出することから始める

```
require'gl2'      NB. gl2
coinsert'jgl2'   NB. seek

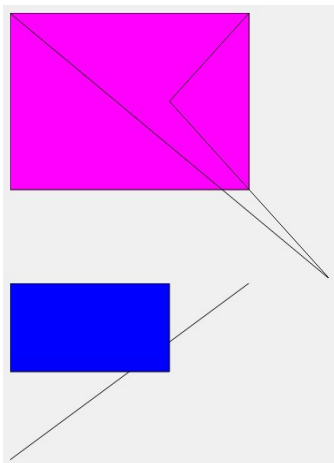
MYFORM=: 0 : 0
pc myform closeok;
minwh 400 300;cc g0 isigraph;
bin s;
minwh 300 200;cc g1 isigraph;
pas 0 0;
rem form end;
)

run=: myform_run=: 3 : 0
wd MYFORM
wd'pshow;'
)

myform_g0_paint=: 3 : 0
glrgb 255 0 255
glbrush ''
glrect 0 0 300 200
gllines 0 0 400 300 200 100 300 0
)

myform_g1_paint=: 3 : 0
glrgb 255 0 0      NB. RGB red
glbrush ''        NB. red brush
gllines 0 200 300 0
glrgb 0 0 255     NB. RGB blue
glbrush ''        NB. red brush
glrect 0 0 200 100 NB. red rectangle
)
```

実行例 実行例とコメント



1. 基点は左上が $(0, 0)$
2. Y 軸は下向きがプラス
3. マイナスは入らない。
4. 座標は絶対座標。データの方でこれに合わせて変換しなければならない
5. 整数のみ。実数を使うとハングする。
6. 大変なのはハングしたときはもう回復しないことが多く、リセットが必要である。(忍耐力が問われる)

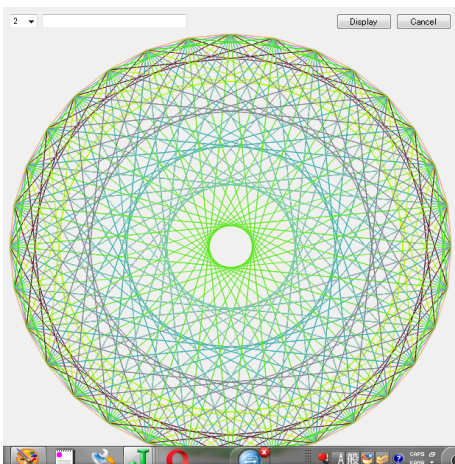
作成上の要点 把握しておくポイント。

1. 絵一枚に額縁一枚である
2. *FORM paint run* の 3 点セット
3. *FORM* と *cc* で指定するチャイルド名は重要
4. 3 点セットの名前は上の 2 の名を用いる

5. 小数、マイナスの数は使えない (必ずハングし、リセットしなければならない)

2 gl2 版の糸掛け曼荼羅を

先に dwin 版で糸掛け曼荼羅が快調に動いた。gl2 版ではボタンを組み込み糸一本ずつ描くスタイルに挑戦する



2.1 計算関数のコピー

dwin 版とは次の点が異なる。コピーして修正する

1. データの書式は横型 $x\ y\ x\ y\ x\ y\ x\ y\ x\ y\ \dots$
2. 座標は絶対座標 (0,0) は左上, Y 軸は下方がプラス
3. 正の整数しか受けつけない。小数が出れば必ずハング。マイナスははみ出る
4. 座標をアジャストする動詞
後ろから 4 作用。複素数を実虚部に分離し、全部プラスにし、300 倍して、整数にする
`adj_isedit=: 3 : ' ; >. 300 * 1 1 + "1 +. y'`

2.2 Script

プリアンプル まずロードする

```
require'gl2' NB. load gl2 definitions in jgl2 locale
coinsert'jgl2' NB. allow use of gl2... without _jgl2_
```

フォーム 窓 (edit) とボタンを設ける。JQT 版はフリーデザインでは無くプレファブで

ある。

窓の大きさは指定しなければ、自動化。積む方向は改行コード (bin) に方向を指定する。

詳細は [WIKI/MainPage/Guides/FAQs/WindowDriver/Forms/Layout](#) で

```
ISEDIT=: 0 : 0
pc isedit closeok;
bin vh;
cc primes combobox;
set primes items 2 3 5 7 11 13 17 19 23 29 31;
cc color edit;
bin s;
cc display button;cn "&Display";
cc cancel button;cn "&Cancel";
bin z;
minwh 600 600;cc graph2 isigraph;
bin z;
pas 0 0;
rem form end;
)
```

run チャイルド *graph2* を起動

```
run=:isedit_run=: 3 : 0
wd ISEDIT
gsel graph2 NB. OK
wd 'pshow'
)
```

paint 一気に書き上げる。

```
isedit_graph2_paint=: 3 : 0
for_ctr. i. # PRIME do.
tmpcolor=: rnd_color ''
glrgb tmpcolor NB. line
glpen 1 1
glbrush '' NB. red brush
gllines adj_isedit@calc_pos ctr{PRIME
end.
)
```

- 色は *RGB* を乱数で打ち出す
- 32 ピンに固定したので計算関数は単項に変更した
- *Display* ボタンで再描画できる
- *chkg12* 関数が厳密で *glpen glbrush* によくエラーが入る (消せば続行する)