

g12 グラフィックスの J の版による仕様の違い —簡単なテストプログラムを作って—

西川 利男

1. J のバージョンとグラフィックス

J のシステムはウィンドウズ版にしても、J3, J4, J5 そして最新では J6 とバージョンアップを重ねてきた。

J6 になっての言語仕様の変更として、関数の引数が x. から x へ、y. から y へなどあるが、J6 の新しいプリミティブ M. (Memorize) について先に報告した[1]。

J6 のグラフィックスについても、わずかだが仕様の変更が行われているようである。筆者が J4 など以前作ったプログラムが J6 ではうまく動かないとのクレームが寄せられ、これを明らかにするため、簡単なテストプログラムを作って比較検討してみた。

ところで、J のグラフィックスには次のようにいろいろな方法がある。

- ① wd や g12 コマンドによるプログラミング
- ② plot ツール
- ③ viewmat ツール

まず、②と③とは簡便でかつ高性能であり、これについては志村正人氏の詳細な解説がある[2]。しかし、筆者はやや特殊なグラフィックスを行うことが多く、そのため①の基本的なコマンドでプログラミングを行わざるをえない[3]。

2. J のバージョンによる g12 コマンドの扱い方の違い

J のグラフィックスでは、例えば線を引くといった操作は gline などのコマンドによりおこなうが、この仕様が J のバージョンにより異なる。

J3 まで …… グラフィックス操作は wd (=windows driver) の文字列引数として行う

J4, J5 …… グラフィックスは専用の g12 コマンド群によりおこなう

J6 …… g12 コマンドの仕様の一部が廃止や変更された

ここで g12 コマンドとは J のプリミティブ 11!:2000+n に名前を付けたものである。

さらに、J の各バージョンによる基本の相違点を比較するとつぎのようになる。

	J402	J504	J602
クラスファイル jg12	不要	必要 coinsert 'jg12'	必要 coinsert 'jg12'
グラフの表示	glshow'	glshow'	glpaint''
表示の座標値	左下(0, 0) —右上(1000, 1000)	左下(0, 0) —右上(1000, 1000)	左上(0, 0) —右下(200, 200)

文献

[1] 西川利男「J602 の新しい機能 M. について」JAPLA シンポジウム資料 2007/12/8

[2] 志村正人「J のグラフィックス (基礎編—1) (J6 版)」

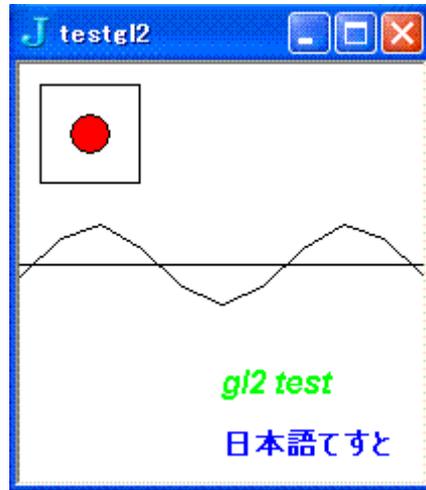
JAPLA 研究会資料 2007/2/21

[3] 西川利男「J グラフィックスによる一般のレムニスケート図形」

JAPLA 研究会資料 2008/2/23

3. J6 上での簡単な gl2 グラフィックスの実行とテストプログラム

例題として、J6 上で次のような簡単なグラフィックスでやってみよう。



そのプログラムは以下のように作られる。gl2 のコマンドを含めて、プログラムを一行ごとに説明しよう。

まず最初に、以下のファイルの読み込みが必要である。

```
require 'gl2'           NB. gl2 コマンドのロード
coinsert 'jgl2'        NB. クラスファイルの組み込み
```

続いて、ウィンドウズ・フォームを作成する。

```
TESTGL2 =: 0 : 0           NB. 名詞としてデータを作成
pc testgl2 closeok;       NB. parent の作成
xywh 0 0 100 100;cc g0 isigraph rightmove bottommove; NB. child の作成
pas 0 0;
pcenter;                  } NB. parent のサイズ調整、位置を決める
pmove 150 50 106 110;
)
```

実行のプログラムはつぎのようになる。

```
testrun =: 3 : 0
wd TESTGL2
glrgb 0 0 0
glrect 10 10 50 50
glrgb 255 0 0
glbrush ''
glellipse 25 25 20 20
gllines 0 100 200 100
XX =: 20 * i.11
YY =: 100 + 20 * 2 o. _5 + i.11
gllines , XX ,. YY
```

NB. 日の丸の図形
(長方形と円の塗りつぶし)

NB. コサインの曲線

```

glfont 'arial 12 bold italic'
glrgb 0 255 0
gltextcolor ''
gltextxy 100 150
gltext 'gl2 test'
glfont 'arial 12 bold'
glrgb 0 0 255
gltextcolor ''
gltextxy 100 180
gltext '日本語ですと'
glpaint ''
wd 'pshow;'
)

```

} NB. 英数文字の印字
 } NB. 日本語文字の印字
 NB. 画像をバッファに格納
 NB. ウィンドウズ画面の表示

4. J4、J5におけるプログラムの差異と実行例

プログラムの最初では、以下のロードは必要である。

```
require 'gl2'
```

NB. gl2 コマンドのロード

しかし、クラスファイルの組み込みは J5 では必要だが、J4 では必要ない。

```
coinsert 'jgl2'
```

NB. クラスファイルの組み込み

大きな違いは、最初の比較一覧のようにピクセル座標値の方向(左上から右下)とそのウィンドウズ画面の大きさ(200x200 から 1000x1000)である。

そのためたとえば、

```
glrect 10 10 50 50 は glrect 100 100 300 300
```

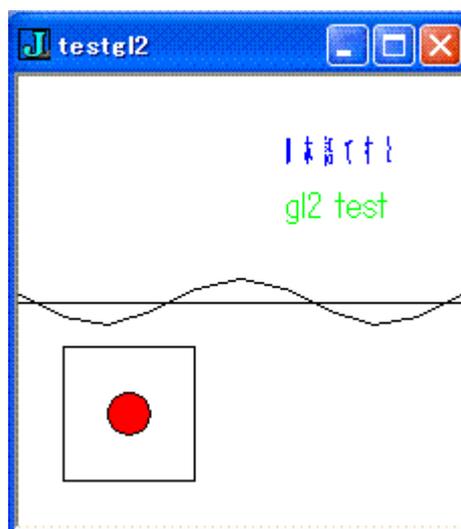
のようが必要になる。

また、

```
glpaint '' は glshow ''
```

とする。

実行した結果は次のように次のようになる。ピクセル座標値の方向は比較のために、あえてそのままにした。



NB. Test J602 gl2 Graphics - test_gl2.ijs =====

```
require 'gl2'  
coinsert 'jgl2'
```

NB. Simple Test =====

```
NB. testrun ''  
TESTGL2 =: 0 : 0  
pc testgl2 closeok;  
xywh 0 0 100 100;cc g0 isigraph rightmove bottommove;  
pas 0 0;  
pcenter;  
pmove 150 50 106 110;  
)
```

```
testrun =: 3 : 0  
wd TESTGL2  
glrgb 0 0 0  
glrect 10 10 50 50  
glrgb 255 0 0  
glbrush ''  
glellipse 25 25 20 20  
gllines 0 100 200 100  
XX =: 20 * i.11  
YY =: 100 + 20 * 2 o. _5 + i.11  
gllines , XX ,. YY  
glfont 'arial 12 bold italic'  
glrgb 0 255 0  
gltextcolor ''  
gltextxy 100 150  
gltext 'gl2 test'  
glfont 'arial 12 bold'  
glrgb 0 0 255  
gltextcolor ''  
gltextxy 100 180  
gltext '日本語ですと'  
glpaint ''  
wd 'pshow;'  
)
```

```

NB. Modified to J504 version, same as J402 =====
testrun =: 3 : 0
wd TESTGL2
glrgb 0 0 0
glrect 100 100 300 300
glrgb 255 0 0
glbrush ''
glellipse 200 200 100 100
gllines 0 500 1000 500
XX =: 100 * i.11
YY =: 500 + 50 * 2 o. _5 + i.11
gllines , XX ,. YY
NB. glfont 'arial 12 bold italic'
glrgb 0 255 0
gltextcolor ''
gltextxy 600 750
gltext 'gl2 test'
NB. glfont 'arial 48 bold'
glrgb 0 0 255
gltextcolor ''
gltextxy 600 880
gltext '日本語てすと'
glshow ''
wd 'pshow;'
)

```