

## 短講

# GLUT を使ってみよう！

星 貴之

平成 18 年 6 月 27 日

## 1. はじめに

研究で開発したセンサの特性を調べるときは、データを記録して後から解析するより、直感的にわかる形でリアルタイムに観察したほうが効率がよい。例えば電圧出力を音として鳴らしたり、測定値をグラフや絵として描画したりである。測定値の描画は、GLUT を利用すると簡単に実現できる [1]。今回は GLUT のサンプルプログラムを紹介し、その内容について説明する。

## 2. セットアップ

### 2.1 ダウンロード

GLUT (OpenGL Utility Toolkit) はライブラリとして提供されており、Windows 版は

<http://www.xmission.com/~nate/glut.html>

からダウンロードできる (glut-3.7.6-bin.zip)。また次のドキュメントも入手しておくとも役立つかも。

<http://www.opengl.org/resources/libraries/glut/glut-3.spec.pdf>

### 2.2 インストール

glut-3.7.6-bin.zip を解凍し、展開された各ファイルを以下のように配置する。開発環境は Visual C++ 6.0 を想定している。

- glut.h → C:\Program Files\Microsoft Visual Studio\VC98\Include\GL
- glut32.lib → C:\Program Files\Microsoft Visual Studio\VC98\Lib
- glut32.dll → C:\WINDOWS\system

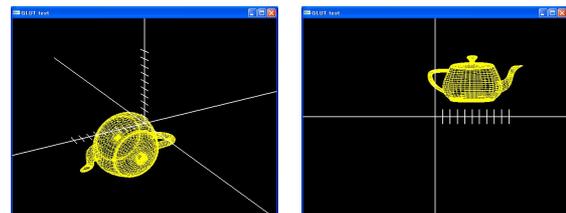
### 2.3 プロジェクト

1. Visual C++ 6.0 を起動する。
2. [ファイル] - [新規作成] で “Win32 Console Application” を選択し、プロジェクト名 (= 実行ファイル名) を決め、“空のプロジェクト” を作成する。
3. main.cpp を作成し、プロジェクトに追加。プログラムを記述する。
4. 通常はデフォルトの設定でビルドできるはず。うまくいかない場合は、[プロジェクト] - [設定] の “リンク” タブの “オブジェクト / ライブラリ モジュール” に “glut32.lib glu32.lib opengl32.lib” の 3 つを追加する。

## 3. サンプルプログラム

ティーポット (glutWireTeapot) と座標軸を表示するプログラムである。r キーで  $(x, y, z)$  表示と  $(x, y)$  表示を切り替えられる (Fig.1)。

<http://star.web.nitech.ac.jp/pdf/060627short.zip>



(a) (b)

Fig.1 サンプル. (a) 3次元表示, (b) 2次元表示.

```
int main(int argc, char **argv)
```

【必須】メイン関数. OpenGL 環境の初期化、ウィンドウ作成 (glutCreateWindow)、関数の設定をして、そのあとは無限ループ (glutMainLoop) で待機。

```
void resize(int w, int h)
```

【必須】表示領域の設定. 最初やウィンドウサイズが変わったときに実行される。w, h はウィンドウの幅と高さ。今回はウィンドウサイズを変えても表示される内容がひしゃげず、適宜拡大縮小されるようにしてある。フラグによって視点位置を変えることで、2次元と3次元を切り替えている。

```
void display(void)
```

【必須】glutPostRedisplay を呼んだときに描画する内容を記述しておく。球を描く関数 (glutSolidSphere) なども用意されているが、多くの場合は空間中に点を打ち、glBegin(GL\_LINES) などですつないでいく (Fig.2)。

- glClear / 画面の消去。
- glPushMatrix ~ glPopMatrix / 変換行列の保存と復帰。これらで挟まれた区間での座標変換は他に影響しない。
- glTranslated / 平行移動。球やティーポットなどの関数は原点に描画するため、あらかじめ平行移動を施しておく必要がある。回転は glRotated。
- glColor3d / これから描画するものの色を指定。RGB の値を 0~1 で設定する。
- glLineWidth / 線の太さを設定。
- glBegin ~ glEnd / これらの間に並べられた点をつないで描画する。
- glVertex3d / 3次元空間中に点を配置。

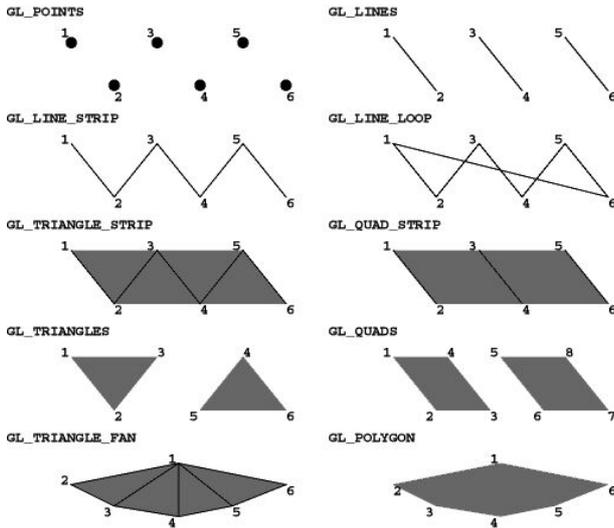


Fig.2 glBegin による点の接続の仕方. 文献 [1] より転載.

#### void timer(int value)

【必須】データの読みやその処理など、繰り返し実行したい内容を記述しておく。再帰呼び出しにより一定周期 (glutTimerFunc の第一変数, ms 単位) で繰り返す。glutTimerFunc の第三変数は timer に渡される value で、複数のタイマーの ID 情報。今回はひとつのタイマーを用いてティーポットの位置を更新している。

他には、main 関数内で glutIdleFunc を使って“暇なときに実行する”こともできる。

#### void keyboard(unsigned char key, int x, int y)

【適宜】キーボード入力があったときに実行したい内容を記述しておく。key は押されたキー、(x, y) はそのときのマウスの位置。

#### void skeyboard(int key, int x, int y)

【適宜】特殊キー (矢印, PgUp, PgDn, etc.) 入力があったときに実行したい内容を記述しておく。key は押されたキー、(x, y) はそのときのマウスの位置。

#### void mouse(int button, int state, int x, int y)

【適宜】マウスボタン入力があったときに実行したい内容を記述しておく。button は左中右, state は押したか放したか、(x, y) はそのときのマウスの位置。

マウスの動きに関しては、main 関数内で glutMotionFunc (ボタンを押しながら) や glutPassiveMotionFunc (ボタンを押さずに) を使って関数を指定する。

#### void init(void)

【自作】最初に一回だけ行なう環境・パラメータの初期設定を記述しておく。main 関数内に直に書いてもよいが、まとまりをよくするために関数化した。

#### void CloseWnd(void)

【自作】プログラム終了時に実行したい内容を記述しておく。デバイスクローズやデータ保存などはここに。

#### void DataSave(char \*folder, char \*databuf)

【自作】時刻をファイル名としてデータを保存する関数。folder は保存先フォルダ名, databuf はデータの文字列。windows.h のファイル操作関数を使っている。

#### 4. おわりに

紹介したサンプルプログラムはデータ表示という用途に限定した一例です。やや稚拙なコーディングですが、とりあえずデータを見るには十分かと思います。また GLUT (OpenGL) はグラフィックライブラリなので、もちろん光源、陰影、材質の設定やテクスチャマッピングもできます。それらについては文献 [1] が Google で調べてみてください。

#### 参考文献

- [1] 床井浩平: GLUT による「手抜き」OpenGL 入門, <http://www.center.wakayama-u.ac.jp/~tokoi/opengl/libglut.html>.

#### A ダブルバッファリング

ソースの冒頭で #ifdef によってダブルバッファリングするかしないかを指定している。ダブルバッファリングとはバックグラウンドに仮想的な画面を用意して、そこへの描画が完了したら表示している画面と一気に入れ替える方法で、通常の毎回画面を書き換えていく方法では表示がチラつくときに使用する。しかしダブルバッファリングに対応していないのか描画が乱れる PC もあったため、#ifdef で対処している。