

## キャラクタ入力/出力からマウス入力/CG出力へ From character-input/output to mouse-input/CG-output

嶋田 憲央† 中川 直哉‡ 柳瀬 龍郎† 田村 信介†  
Norio Shimada Naoya Nakagawa Tatsuro Yanase Shinsuke Tamura

### 1.はじめに

GUI入力や3DCG出力は、計算結果の出力及びユーザーとの対話処理の面において非常に有効な手段である。今日、私たちが日常で触れている様々なアプリケーションではそれらのグラフィカルな入出力が用いられているが、C言語に触れている限りでは、キーボードやファイルからの文字入力や、標準出力のような文字(キャラクタ)のみの簡素な出力など、グラフィカルな入出力とは無縁の機能にしか触れることがない。

C言語においてGUI入力や3DCG出力を実装するためには色々な知識が必要となり、プログラミングコスト増大の一因となる。もし前述した高度な入出力が簡単なコード記述のみで行うことが出来れば、C言語でもグラフィカルで対話性溢れるアプリケーションプログラミングが簡便となると考えられる。

本研究の目的は、GUI入力と3DCG出力を標準入出力のように手軽に扱うことができるようにするライブラリの提案と実装である。

### 2.GUI入力

#### 2.1 システム

GUI入力システムは、アプリケーションプログラムとは別に、GUIアプリケーションを構築するのに必要な機能を持つプログラム「GUIマネージャ」を用意し、GUIマネージャをアプリケーションプログラムから通信により操作する事でGUI入力を可能にする、というシステムである。

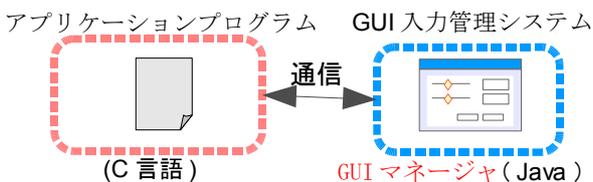


図1: GUI入力システムの概要

GUIマネージャを操作する為のC言語関数ライブラリを予め用意しておき、C言語アプリケーションプログラムはその関数ライブラリを用いてGUIの構築や入力部品の値の読み取り等を行う。関数内では、例えばGUIマネージャの起動やソケット通信の準備、部品追加依頼等を通信により行うが、プログラムはその処理内容を気にかける必要はない。

#### 2.2 GUIマネージャの機能

アプリケーションプログラムでGUI入力を行うためには、次のようなことが必要となる。

- ・入力部品を表示させるためのウィンドウを持つ
- ・入力部品の追加が可能
- ・入力部品の状態(値)の参照が可能

† 福井大学

‡ 株式会社源コーポレーション

- ・入力部品のレイアウトが可能

これらを考慮し、GUIマネージャは以下の機能を持つようにした。

- ・アプリケーション側との資源共有機能
- ・アプリケーション側との通信機能
- ・入力部品を貼り付けるためのウィンドウ生成機能
- ・入力部品の生成機能
- ・生成した入力部品を適切に配置する機能
- ・ユーザー入力に対する応答処理(ステータス変更・描画処理等)機能

#### 2.3 関数ライブラリのAPI

以下に関数のAPIを示す。ライブラリには、GUIマネージャを操作する関数の他に、GUI部品を扱うための構造体も含まれている。

```
typedef struct GMS_GP{...}GMS_GP
: 部品情報記憶用構造体
void GMS_init(void) : 初期化用関数
void GMS_close(void) : 終了処理関数
GMS_GP *GMS_add(int type,char *name,
int value1,int value2,int value3)
```

: 部品の追加を依頼する関数

**type** : 作成する部品の種類

**name** : 部品につける名前

**value1~3** : 用途は部品によって異なる  
(初期値、最大値など)

```
void GMS_geti(GMS_GP* parts, int *x)
```

```
void GMS_getd(GMS_GP* parts, double *x)
```

: 部品が示す値を変数に代入する

**parts** : 代入もとの部品の識別子

**x** : 入力部品から得た値を格納する変数のポインタ

### 3.CG出力

#### 3.1 システム

アプリケーションプログラム内にて関数を記述するだけで簡単にCGが出力できるシステムである。メインプログラムとは別のプロセスで動くCG出力用プログラム「CGS」を、アプリケーションプログラムにおいてgprintfという関数の呼び出しにより起動し、アプリケーションプログラムから受け取った表示データを用いてCG出力を行う。gprintf関数内では、CGSの起動、CGSとのソケット通信の接続、送受信などを行っている。

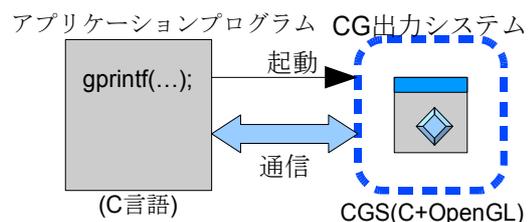


図2: CG出力システムの概要

### 3.2 CGS の機能

CGS で、CG を出力するために必要な機能や、あると便利だと考える機能を持たせた。

- ・ CG 出力用ウインドウを生成する機能
- ・ CG を出力する機能(座標点を頂点とする面を出力)
- ・ 作成した CG をマウスで対話的に視点変更することができる機能
- ・ 再度関数が呼ばれた時、CG 出力用ウインドウを作り直さず CG オブジェクトのみを上書きする機能
- ・ アプリケーションプログラムとの通信機能
- ・ CG 出力に使用したデータを数値としてテキスト出力する機能 (使用するかは選択できる)

### 3.3 CG 出力関数の文脈

出力のための API は次に記す。

```
void gprintf(int type, int line, int row,
            int text, float color[], float point[][]);
```

**type** :表示形式  
**line** :CGデータの行数(pointの2次元目の要素数)  
**row** :CGデータの列数(pointの3次元目の要素数)  
**text** :テキスト表示のON、OFF  
 (0ならoff、1ならon)  
**color** :CGの色データ(1次元配列で要素はRGBA)  
**point** :CGの座標データ(3次元配列で、  
 [xyzの区別][行番号][列番号]の順)

## 4. 使用法

GUI 入力システムと CG 出力システムを併用して、トーラス図形を表示させるプログラムを製作した。

```
#include "gprintf.h"
#include "gms.h"
#include <math.h>
#define PI 3.1415926
#define N 0.5
float x,y,z,R,r1,r2,theta1,theta2;
float p[3][201][201];
GMS_GP* slider1;
GMS_GP* slider2;
int main(int argc, char** argv){
    int i,j,n,line,row;
    int last_line,last_row;
    float color[]={1.0,0.0,0.0,0.6};
    GMS_init();
    slider1 = GMS_add(GMS_ATSLIDER,"Line",10,100,4);
    slider2 = GMS_add(GMS_ATSLIDER,"Row",10,100,4);
    last_line = 10; last_row = 10;
    while(1){
        GMS_getd(slider1, &line); ←②
        GMS_getd(slider2, &row);
        if(line != last_line || row != last_row){
            if(line >= 4 || row >= 4){
                theta1 = 2*PI/(row-1); theta2 = 2*PI/(line-1);
                r1 = N; r2 = N/2.0;
                for(i = 0; i < line; i++){
                    R = r1 + r2 - r2*cos(PI+theta2*i);
                    for(j = 0; j < row; j++){
                        p[0][i][j] = R*sin(theta1*j);
```

```
p[1][i][j] = r2*sin(theta2*j);
p[2][i][j] = R*cos(theta1*j);
    }
    gprint(CG_MESH,line,row,1,color,p); ←③
    }
    last_line = line;
    last_row = row;
    }
}
```

①にてスライダー“Line”と“Row”が生成され、②にて各スライダーから得た値を対応する変数に代入している。そして③で、与えられたデータにより、CG を出力する。出力されるトーラス図形は二つの円に近似した多角形を基にして作られているが、スライダーを操作することによって、多角形の画数を動的に更新することができる。

GUI マネージャにより管理される入力スライダー ↓

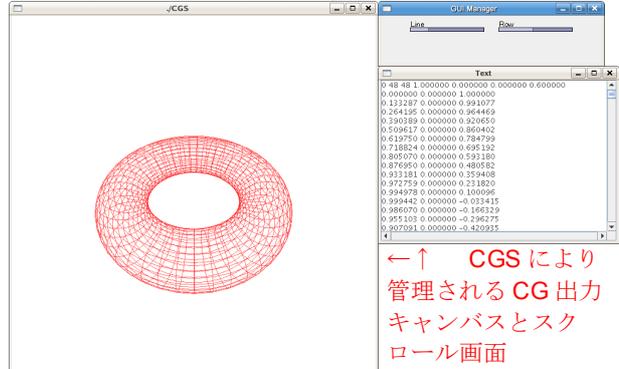


図 3:本研究結果を用いたサンプルプログラムの GUI

このサンプルプログラムでは、GUI 入力と CG 出力を行っているが、プログラムの内容のうち GUI 入力用の命令は 8 行、CG 出力用の命令は 2 行である。GUI 入力と CG 出力が簡単なコード記述で行えた。

## 5. まとめ

GUI 入力と 3DCG 出力を手軽に扱えるようにするライブラリを提案し、実装を行った。その結果、簡単なコード記述のみで、C 言語における GUI 入力と 3DCG 出力することができ、またアプリケーションに負担を与えずに、CG オブジェクトに対する視点変更が可能となった。

本研究を利用することで、GUI を利用した対話性の実現のためのプログラミングコストが抑制され、その結果、アプリケーション記述にプログラミングコストを集中することができると思われる。

## 6. 参考文献

- [1]嶋田憲央：“C 言語において CG を簡単に出力する関数” 福井大学工学部 情報・メディア工学科平成 18 年度卒業研究論文(2007.3)
- [2]中川直哉：“GUI システムと新しい入力部品の実装” 福井大学工学部 情報・メディア工学科平成 18 年度卒業研究論文(2007.3)
- [3]ALAN W.PAETH：“GRAPHICS GEMS V” AP PROFESSIONAL(1995)