

教育用簡易 XWindow プログラム開発システム

坪井 始*

Simple XWindow Program Development System for Education

Hajime TSUBOI*

ABSTRACT

A simple XWindow program development system for education was developed. In the developed system, the window system program is hidden so that students do not consider it, and students may write only programs for graphics in order to learn programming. The simple XWindow program development system and a library for the system were proposed. Furthermore, sample programs and graphics displayed by the programs were shown.

Keywords: XWindow, UNIX/Linux, window system, programming exercise, C language.

1. まえがき

プログラミング演習は、基礎段階ではキャラクターベースの演習プログラムを提示して行うことが一般的であるが、その後はグラフィックスを用いた演習プログラムの導入によりプログラミングへの興味をもたせることが考えられる。これは、視覚的にプログラムの実行結果が確認できるという意味で、学習者にとってわかりやすい方法である。しかし、現在のコンピュータ環境では、グラフィックス表示を行うにはウインドウシステムを用いる必要があり、プログラミングが容易ではない。すなわち、ウインドウシステムの複雑さのためにプログラミングが難解となり、プログラミングの演習がかえって馴染みにくいものとなる可能性がある。

そこで、学習者がほとんどウインドウシステムを意識することなくグラフィックスを使用できるように、ウインドウシステムのプログラムを隠蔽し、グラフィックス表示プログラム部分のみを作成すればよい簡易グラフィックス表示システムおよびライブラリ提案する。

標準化された汎用性の高いグラフィックスシステムがいくつか発表されている。中でも、Open GLのGLUTやJavaなど、最新のコンピュータシステムと親和性の高いグラフィックスシステムがあるが、ここではUNIX系のシステムに標準でインストールされるXWindow

システムを用いる。さらに、通常のイベント駆動型のプログラムを意識することなく、プログラミングができるようにするとともに、単純化した関数群を用意してグラフィックス表示を行う。

以下に、開発したプログラムの仕様および概要を示すとともに、実行例を示す。

2. XWindow システム

XWindowは、UNIX系（UNIX、Linux、Mac OS X）のOS（Operating System：オペレーティングシステム）で標準的にインストールされるXプロトコルを用い、GUI（Graphical User Interface）を提供するウインドウシステムである。XWindowのプログラミングには、基本的にはXlibを用いるが、様々な開発ツールが存在する。また、Open GL、JavaなどもXWindow上で動作する。

Xlibは、最も下位に位置するXWindow用ライブラリで、基本的なインストールで使用可能である。Xlibを用いた表示プログラムの開発[1]は煩雑であるが、簡易な表示プログラムを作成する場合には問題とならず、必要最小限の機能を実現するために使用する関数は僅かな数である。実用的なXWindow用のプログラム開発では、さらに他の開発ツールやライブラリを使用した方

* 情報処理工学科

が効率的である。

XWindow 用のプログラムは煩雑なイベント駆動型のプログラムになるため、演習用のプログラムでは、これを意識させないようにする必要がある。

3. 簡易 XWindow 用グラフィックス表示プログラム

開発したプログラムは、図1に示すようにイベント駆動型のプログラムで、つぎの3つのイベントを処理している。

- (1) Expose
- (2) ButtonPress
- (3) KeyPress

Expose イベントの処理では、ウィンドウの前面への表示と同時にリサイズに対応している。

ButtonPress イベントにより、マウスボタンの押下(クリック)に対応し、ウィンドウ内の押下位置の座標も入力している。3ボタンマウスを用いて、左ボタン、中ボタン、右ボタンの押下により、押下位置をウィンドウの中心とする表示内容の拡大 (zoom in)、平行移動 (pan)、縮小 (zoom out) をそれぞれ可能にしている。

KeyPress イベントでは、'Q' キーの入力のみをチェックし、これが押下された場合、プログラムの実行を終了 (quit) する。

グラフィックス表示ウィンドウは、ターミナルウィンドウから起動するグラフィックス表示プログラムからオープンされるが、'-display display_name' オプションにより任意のディスプレイに表示できるようにしている。また、グラフィックス表示ウィンドウの初期サイズは 500 x 500 ピクセルとしており、これは表示結果の画面コピーを取り易いように比較的小さいサイズにしたものである。

また、最近の OS では、スレッドを使用してイベントループを独立させることができるが、ここでは逐次処理のみでプログラムを構成している。

プログラムは、ヘッダファイルの形で提供し、学習者は図1の太線で示すつぎの3つの関数を作成する。

- (a) gin
- (b) gmain
- (c) gout

関数 gin では、ファイル入力、表示データ作成などグラフィックス表示に必要でかつ1回のみの実行でよい処理を行う。後述する仮想ウィンドウの設定もここで行う。

関数 gmain が、グラフィックス表示を行う関数で、表示関数の呼び出しにより表示を行う。

関数 gout では、計算結果のターミナルウィンドウへの表示、計算結果のファイル出力などの後処理を行う。

開発したプログラムを用いてグラフィックス表示プロ

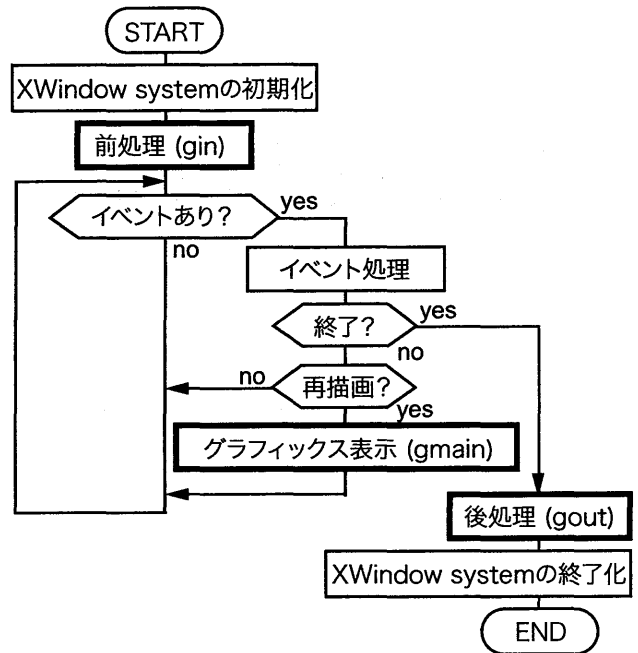


図1 開発したプログラムの流れ図
Fig. 1 Flowchart of the developed program.

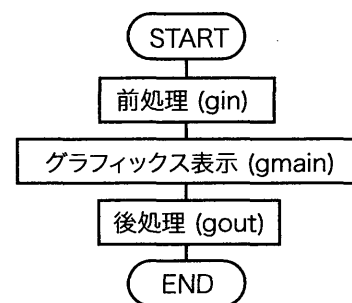


図2 見かけの流れ図
Fig. 2 Appeared flowchart.

グラムを作成する際には、学習者は図2に示す見かけの流れ図のみを意識してプログラム作成すればよく、煩雑なウィンドウプログラミングを考慮する必要がなくなる。

リスト1は、学習者に提示するスケルトンプログラムである。これを使用して作成したプログラムの表示例を図3に示す。実行は、Mac OS X 10.4 上で行ったものである。xterm 上から、プログラム pro2 を実行したもので、ウィンドウ名 Data で表示している。演習では、表示結果のハードコピーを提出させるため、関数 name により、表示ウィンドウ内に学生番号とローマ字の名前を表示している。この名前表示は、画面の拡大、縮小等によって表示位置や文字の大きさが変化しないよう、リスト1の関数 name に示すように工夫している。

グラフィックス表示を行うために、いくつかの関数を作成し、ヘッダファイル gfunc.h の中に含めている。グ

リスト1 スケルトンプログラム

List 1 Skeleton program.

```
#include "gfunc.h"
void name( void );

void gin( void )
{
    /* Preprocess & Initialize */
    Set_Vwindow( xmin, ymin, xmax, ymax );
}

void gmain( void )
{
    /* Draw Graphics */
    name();
}

void gout( void )
{
    /* Finalize */
}

void name( void )
{
    SaveVWpara();
    Set_Vwindow( 0., 0., 100., 100. );
    SetColor( "black" );
    XPrintString( 70., 3., "No. Name" );
    LoadVWpara();
}
```

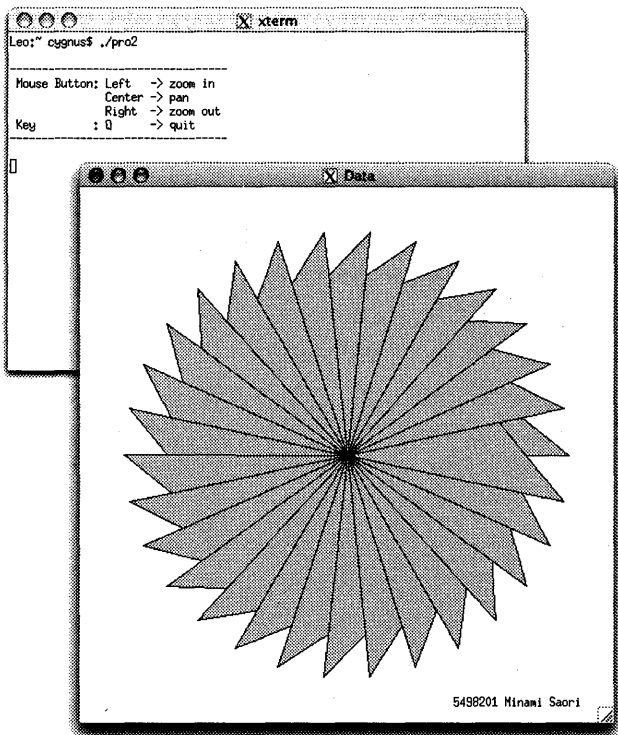


図3 実行例

Fig. 3 Example.

グラフィックス表示関数は、著者が開発した電磁界解析結果可視化用関数ライブラリ [2] のサブセットである。

ここで用意したグラフィックス表示関数は、つぎの通りである。

i) Set_Vwindow(double Xmin, double Ymin,
double Xmax, double Ymax)

表示したい仮想ウィンドウの左下と右上の座標を指定し、この範囲を実際の表示ウィンドウ（初期サイズは 500 x 500 ピクセル）に写像する。したがって、表示したい実座標（仮想座標）で座標を指定すれば、本表示システムが座標変換を行ってピクセル単位（整数）の装置座標で表示することになる。なお、表示ウィンドウは、左下を原点として右手系で座標を表す。

ii) SetColor(char *s)

色の名前を示す文字列 *s によって表示色を指定する関数で、つぎに表示色を変更するまで有効である。

iii) XPrintString(double x, double y, char *s)

*s で示す文字列（英数字）を座標 (x, y) で指定する位置に表示する関数である。

iv) Point_set(double x, double y)

座標 (x, y) で指定した位置に点を打つ関数である。

v) Move_to(double x, double y)

グラフィックポインタを指定した位置 (x, y) に移動する関数で、表示は何もしない。

vi) Line_to(double x, double y)

現在の位置から、グラフィックポインタを指定した座標 (x, y) まで移動するとともに、実行前のグラフィックポインタの位置から指定した位置まで直線を引く。

vii) tri(double x1, double y1,

double x2, double y2, double x3, double y3,
char *Rcolor, char *Bcolor)

3つの頂点の座標 (x1,y1)、(x2,y2)、(x3,y3) を与えて三角形を描く関数で、Rcolor で三角形内部の色を Bcolor で三角形の辺（境界）の色を指定する。

ヘッダファイル gfunc.h は、main 関数、必要な制御関数およびこれらのグラフィックス表示関数を含めて約 350 行の小さいプログラムである。

4. プログラム例

開発した簡易 XWindow 用グラフィックス表示プログラムを使用して作成したプログラムとその表示例をそれぞれリスト 2 と図 4 に示す。

時刻 t の座標が次のように表される落下の軌跡を円で描いたものである。

$$x = 2t \quad (1)$$

$$y = -4.9t^2 \quad (2)$$

ここでは、関数 gin で π の計算と仮想ウィンドウの設

リスト2 演習プログラムの例
List 1 Example of exercise program.

```

#include "gfunc.h"

void name( void );
void circle(double r, double xc, double yc );

double PI;

void gin( void )
{
    PI = 4. * atan(1.);
    Set_Vwindow( -5.5, -12.5, 6.5, 0.5 );
}

void gmain( void )
{
    int i;
    double r=0.1, x=0.,y=0.,t,dt=0.1;

    SetColor( "blue" );

    for( t=0.; t<1.5+dt/2.; t += dt) {
        x = 2. * t;
        y = - 4.9 * t * t;
        circle( r, x, y );
    }
    name();
}

void circle(double r, double xc, double yc )
{
    int i,n=200;
    double xo,yo,xn,yn,dt,cs,sn;

    dt = 2. * PI / (double)n;
    cs = cos(dt);
    sn = sin(dt);
    xo = r;
    yo = 0.;

    Move_to( xo + xc, yo + yc);
    for( i=1; i<=n; i++) {
        xn = xo * cs - yo * sn;
        yn = xo * sn + yo * cs;
        Line_to( xn + xc, yn + yc );
        xo = xn;
        yo = yn;
    }
}

void gout( void )
{
}

void name( void )
{
    SaveVWpara();
    Set_Vwindow( 0., 0., 100., 100. );
    SetColor( "black" );
    XPrintString(70.,3.,
                "5498201 Minami Saori");
    LoadVWpara();
}

```

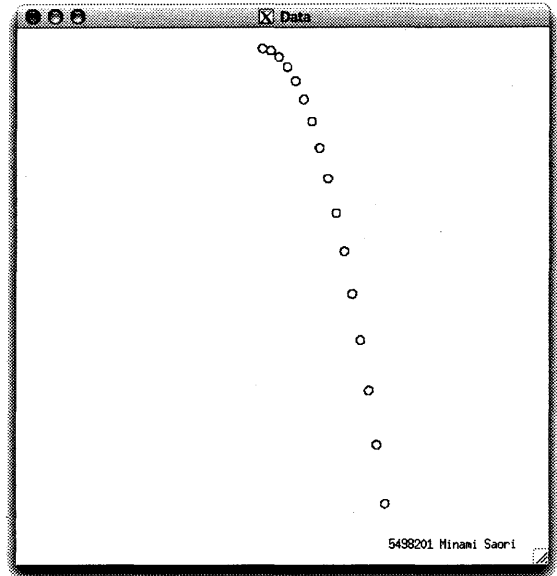


図4 演習プログラムの表示例
Fig. 4 Example of displayed graphics.

定を行い、関数 gout は空の関数である。さらに、円を描く関数を作成し、関数 gmain では、これを使用して落下の軌跡を描いている。これは、円を描く関数を示すことと、関数の使用法を学ばせる課題である。

表示例は、Mac OS X 10.4 上で実行したものであるが、他の OS でも同様に実行することができ、グラフィックス部分は同じ表示となる。

5. むすび

以上、プログラミング演習で用いる XWindow グラフィックス用簡易プログラム開発システムを提案し、その概要を述べた。このシステムは、プログラミング演習で5年間使用の実績があり、効果的に機能している。また、ヘッダファイルにプログラムをすべて収めているため、興味があれば、これを確認することができ、XWindow プログラムに関する自習に役立てることができる。

参考文献

- [1] Oliver Jones : XWindow ハンドブック、アスキー (1990).
- [2] 坪井 始、田中始男、瀬島紀夫 : 電磁界解析結果の視覚化プログラムの開発、電気学会静止器・回転機合同研究会資料 SA-02-26・RM-02-62 (2002).