

マウスイベント

前回は引き続き入力装置を利用したプログラムとして、マウスからの入力に対応したプログラムについて学んでいく。

マウスからの入力

マウスのイベントに対応したプログラム例を以下に示す。前回のテキストのプログラム例2からの変更である。変更部分については太字で表した。

プログラム例1

```
#include <stdio.h>
#include <stdlib.h>
#include <GL/glut.h>

void init_opengl(void); // OpenGLの初期化
void display(void); // コールバック関数glutDisplayFunc()用
void resize(int w, int h); // コールバック関数glutReshapeFunc()用
void keyboard(unsigned char key, int x, int y); // コールバック関数glutKeyboardFunc()用
void special_key(int key, int x, int y); // コールバック関数glutSpecialFunc()用
void mouse(int button, int state, int x, int y); // コールバック関数glutMouseFunc()用

unsigned char color = 0; // カラーの設定に用いる

int main(int argc, char *argv[]) {
    glutInitWindowPosition(100, 100); // ウィンドウの表示位置の指定
    glutInitWindowSize(200, 200); // ウィンドウサイズの指定
    glutInit(&argc, argv); // GLUTの初期化
    glutInitDisplayMode(GLUT_RGBA); // 表示モードの指定
    glutCreateWindow("2D oekaki"); // ウィンドウを生成
    glutDisplayFunc(display); // 描画イベント時のコールバック関数の設定
    glutReshapeFunc(resize); // ウィンドウサイズ変更イベント時のコールバック関数の設定
    glutKeyboardFunc(keyboard); // キーボードイベント時のコールバック関数の設定
    glutSpecialFunc(special_key); // 特殊キーイベント時のコールバック関数の設定
    glutMouseFunc(mouse); // マウスイベント時のコールバック関数の設定

    init_opengl(); // OpenGLに関する初期化 一度だけ呼ばれる

    glutMainLoop(); // GLUTに関する無限ループ

    return 0;
}

void init_opengl(void) {
    // 変更無し
}

void display(void) {
    // 変更無し
}

void resize(int w, int h) {
    // 変更無し
}
```

```

}

void keyboard(unsigned char key, int x, int y){
    // 変更無し
}

void special_key(int key, int x, int y){
    // 変更無し
}

void mouse(int button, int state, int x, int y)
{
    printf("mouse() が呼び出されました。¥n");
    printf("座標(%d, %d)で", x, y);
    switch(button)
    {
        case GLUT_LEFT_BUTTON:
            printf("左ボタンが");
            break;

        case GLUT_MIDDLE_BUTTON:
            printf("中ボタンが");
            break;

        case GLUT_RIGHT_BUTTON:
            printf("右ボタンが");
            break;

        default:
            break;
    }

    switch(state)
    {
        case GLUT_UP:
            printf("離されました。¥n");
            break;

        case GLUT_DOWN:
            printf("押されました。¥n");
            break;

        default:
            break;
    }
}
}

```

このプログラムではウィンドウが表示され、2D の図形が描画される。そのほかにターミナルに文字も出力される。

今回のプログラムではマウスからの入力を監視する `glutMouseFunc()` をメイン関数に記述した。対応するコールバック関数 `mouse()` は以下のようにになっている。

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

この関数が呼び出されると、button には押されたボタンの種類（左、中、右）、state にはボタンの状態（アップ、ダウン）、x、y にはマウスカーソルの位置が格納される。この関数ボタンが呼び出されるタイミングはマウスのボタンが押された時と離れた時である。ボタンを押す場合にはウィンドウの描画領域である必要があるが、一度描画領域内で押した後にドラックしてボタンを離す場合には描画領域の外でもかまわない。これは以下の出力例でも確認できる。最後の出力では、マウスのボタンが離された位置がマイナスとなっていることがわかる。

出力例

```
resize()が呼び出されました。ウィンドウの幅は200 高さは200 です。  
mouse()が呼び出されました。  
座標(101, 74)で左ボタンが押されました。  
mouse()が呼び出されました。  
座標(101, 74)で左ボタンが離されました。  
mouse()が呼び出されました。  
座標(2, 4)で左ボタンが押されました。  
mouse()が呼び出されました。  
座標(-161, -158)で左ボタンが離されました。
```

マウスの座標の原点は、前回のキーボードイベントの時に示したものと同じように、ウィンドウの描画領域の左上となっている（図1に再掲）。

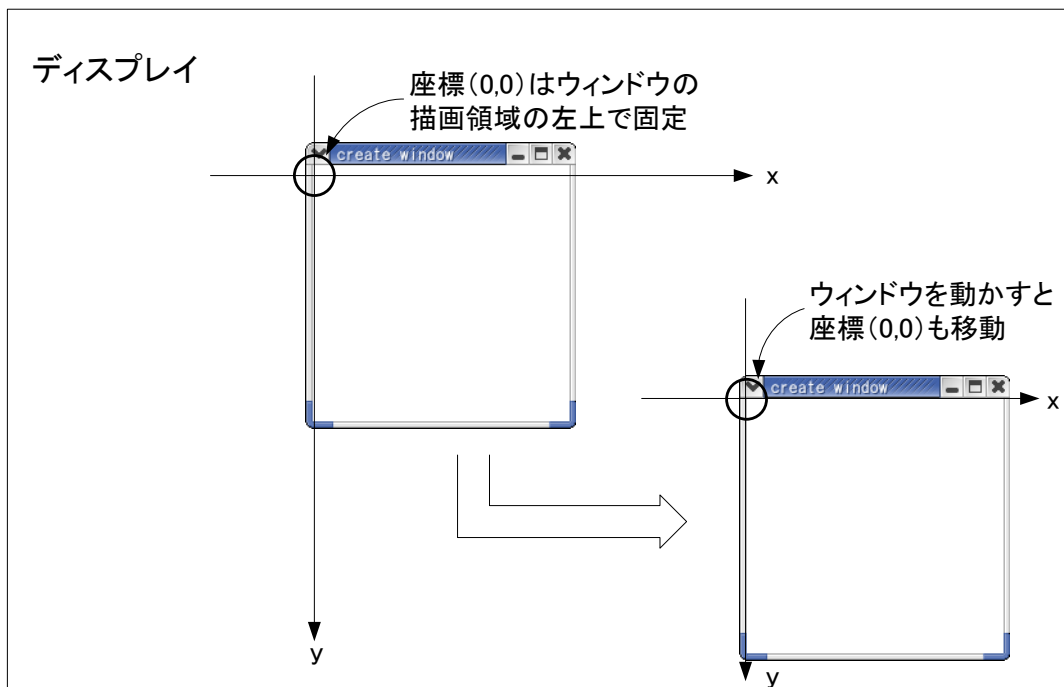


図1 マウスカーソルの座標の取得について

表1 ボタン、状態と GLUT のマクロ定義の対応表

button	GLUT_LEFT_BUTTON	マウスの左ボタン
	GLUT_MIDDLE_BUTTON	マウスの中ボタン
	GLUT_RIGHT_BUTTON	マウスの右ボタン
state	GLUT_UP	マウスボタンを離した
	GLUT_DOWN	マウスボタンを押した

この他にマウスに関するイベント処理は以下がある。

glutMotionFunc()

マウスボタンを押しながらマウスを動かした時に呼び出されるコールバック関数を指定

コールバック関数例

```
void mouse_move(int x, int y)
```

x, y にはマウスカーソルの位置が渡される

glutPassiveMotionFunc()

マウスボタンを押さずにマウスを動かした時に呼び出されるコールバック関数を指定

コールバック関数例

```
void mouse_passive_move(int x, int y)
```

x, y にはマウスカーソルの位置が渡される

glutEntryFunc()

マウスカーソルがウィンドウ領域に入った時と出た時に呼び出されるコールバック関数を指定

コールバック関数例

```
void mouse_entry(int state)
```

state にはマウスカーソルの状態が渡される

GLUT_LEFT ウィンドウの外に出た (フォーカスを失った)

GLUT_ENTERED ウィンドウの中に入った (フォーカスを得た)

以下の例は glutEntryFunc() を使用した例である。プログラム例 1 からの変更として、変更点については太字で示す。

プログラム例 2	
#include <stdio.h>	
#include <stdlib.h>	
#include <GL/glut.h>	
void init_opengl(void);	// OpenGLの初期化
void display(void);	// コールバック関数glutDisplayFunc()用
void resize(int w, int h);	// コールバック関数glutReshapeFunc()用
void keyboard(unsigned char key, int x, int y);	// コールバック関数glutKeyboardFunc()用
void special_key(int key, int x, int y);	// コールバック関数glutSpecialFunc()用
void mouse(int button, int state, int x, int y);	// コールバック関数glutMouseFunc()用
void mouse_entry(int state);	// コールバック関数glutEntryFunc()用

```

unsigned char color = 0; // カラーの設定に用いる

int main(int argc, char *argv[])
{
    glutInitWindowPosition(100, 100); // ウィンドウの表示位置の指定
    glutInitWindowSize(200, 200); // ウィンドウサイズの指定
    glutInit(&argc, argv); // GLUTの初期化
    glutInitDisplayMode(GLUT_RGBA); // 表示モードの指定
    glutCreateWindow("2D oekaki"); // ウィンドウを生成
    glutDisplayFunc(display); // 描画イベント時のコールバック関数の設定
    glutReshapeFunc(resize); // ウィンドウサイズ変更イベント時のコールバック関数の設定
    glutKeyboardFunc(keyboard); // キーボードイベント時のコールバック関数の設定
    glutSpecialFunc(special_key); // 特殊キーイベント時のコールバック関数の設定
    glutMouseFunc(mouse); // マウスイベント時のコールバック関数の設定
    glutEntryFunc(mouse_entry); // フォーカスイベント時のコールバック関数の設定

    init_opengl(); // OpenGLに関する初期化 一度だけ呼ばれる

    glutMainLoop(); // GLUTに関する無限ループ

    return 0;
}

void init_opengl(void) {
    // 変更無し
}

void display(void) {
    // 変更無し
}

void resize(int w, int h) {
    // 変更無し
}

void keyboard(unsigned char key, int x, int y) {
    // 変更無し
}

void special_key(int key, int x, int y) {
    // 変更無し
}

void mouse(int button, int state, int x, int y) {
    // 変更無し
}

void mouse_entry(int state)
{
    printf("mouse_entry()が呼び出されました。%n");
    switch(state)
    {
        case GLUT_LEFT:

```

```
        printf("ウィンドウの外にマウスが出ました。¥n");
        break;

    case GLUT_ENTERED:
        printf("ウィンドウの中にマウスが入りました。¥n");
        break;

    default:
        break;
}
}
```

演習

演習 1 プログラム例 1、2 を作成し、動作を確認しなさい。

演習 2 `glutMotionFunc()`、`glutPassiveMotionFunc()` を利用したプログラムを作成し、動作を確認しなさい。コールバック関数の内容は引数に与えられたマウスの `x`、`y` 座標を出力すること。

演習 3 マウスのクリックした位置によって、背景色を変更されるプログラムを作成すること。

演習 4 なんらかの図形を描画しておき、マウスボタンを押しながらマウスを動かした時にその図形が移動するプログラムを作成すること。