

OpenGL勉強会 第一回

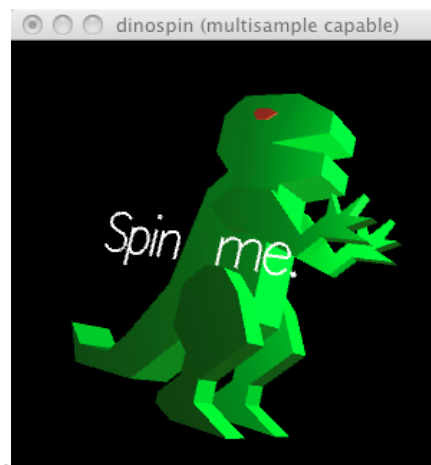
慶應義塾大学大学院
システムデザイン・マネジメント研究科
立山 義祐

SDM OpenGL 勉強会 2013

1

OpenGLデモプログラムをコンパイル、 実行してみよう！

- <http://lab.sdm.keio.ac.jp/tateyama/cprimer/>
- 「OpenGL 勉強会1回目プログラムセット」のopengl1.zipをダウンロード
 - ページのopengl1.zipを右クリックして「ファイルを保存」を選択
 - ダウンロードしたファイルを右クリックして「全てを展開」
- 「cd opengl1」「cd dinospin」
- 「make」
- 「./dinospin」

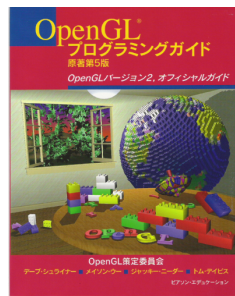
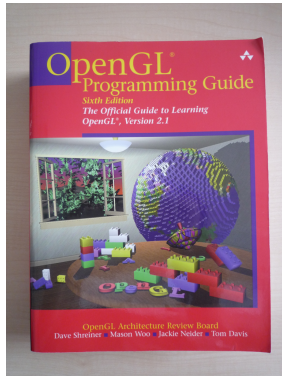


SDM OpenGL 勉強会 2013

2

参考書

- 通称: 赤本
 - OpenGL Programming Guide
 - 7th ed. 6441円
 - OpenGLプログラミングガイド
 - 原著第5版 13,650円
- 通称: 床井本
 - GLUTによるOpenGL入門
 - 1,995円
 - GLUTによるOpenGL入門2 テクスチャマッピング
 - 2,625円



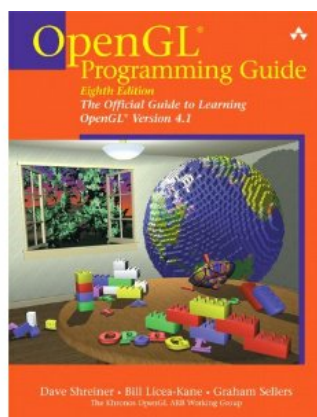
価格は 2013/06/03 Amazon.co.jp調べ

SDM OpenGL 勉強会 2013

3

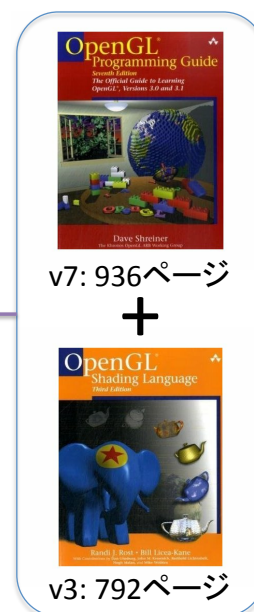
“赤本”じゃなくなりました...

- 第8版から OpenGL Shading Language が合体



984ページ 6,345円 2013/6/3現在

SDM OpenGL 勉強会 2013



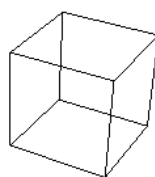
4

OpenGL

- 3D モデルを高速に処理、表示するためのAPI
 - ハードウェア (GPU) による高速3D描画演算
 - Windows でも Mac でも動くデファクトスタンダード
 - iPhone 等でもサブセットが動く (OpenGL ES)
 - 1.x: iPhone, Android, Playstation3, Nintendo 3DS
 - 2.0: iPhone 3GS and later
- カメラ
- ステートマシン

サンプルプログラムのコンパイル

- `cd ../gltest`
- `gcc -o gltest gltest.c -lglut32 -lglu32 -lopengl32`
- `./gltest`



makefile

```
all: gltest
LDFLAGS=-lglut32 -lglu32 -lopengl32

gltest: gltest.o
    gcc -o $@ $^ ${LDFLAGS}
```

実行すべきコマンドの前はTAB

簡単な makefile

```
// doit.c
#include <stdio.h>
```

```
int
doit(int a, int b)
{
    printf("doit!\n");
    return a + b;
}
```

```
// main.c
#include <stdio.h>
```

```
int
main(int ac, char **av)
{
    int val;

    printf("main\n");
    val = doit(10, 20);
    printf("returned value: %d\n", val);
    return 0;
}
```

```
// makefile
```

```
maketest: main.o doit.o
    gcc -o maketest main.o doit.o
```

```
main.o: main.c
    gcc -c main.c
```

```
doit.o: doit.c
    gcc -c doit.c
```

実行すべきコマンドの前はTAB

gltest0.c (Windowを出すだけ)

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

void
display(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    glFlush();
}

void
init(void)
{
    glClearColor(0.0, 0.0, 1.0, 0.0);
}

int
main(int ac, char **av)
{
    glutInit(&ac, av);
    glutInitDisplayMode(GLUT_RGBA);
    glutCreateWindow(av[0]);
    glutDisplayFunc(display);
    init();
    glutMainLoop();
    return 0;
}
```

gltest1.c

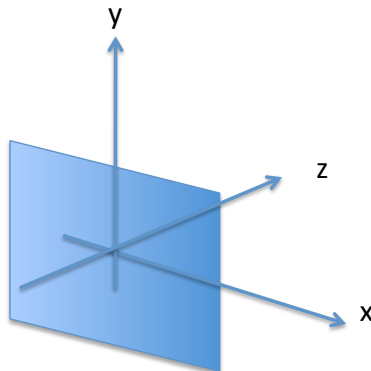
```
#include <GL/glut.h>

void
display(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    // glColor3f(1.0f, 0.0f, 0.0f);
    glBegin(GL_POLYGON);
    glVertex2f(-0.9f, -0.9f);
    glVertex2f( 0.9f, -0.9f);
    glVertex2f( 0.9f,  0.9f);
    glVertex2f(-0.9f,  0.9f);
    glEnd();
    glFlush();
}

void
init(void)
{
    //glClearColor(1.0, 1.0, 1.0, 0.0);
    glClearColor(0.0, 0.0, 0.0, 0.0);
}

int
main(int ac, char **av)
{
    glutInit(&ac, av);
    glutInitDisplayMode(GLUT_RGBA);
    glutCreateWindow(av[0]);
    glutDisplayFunc(display);
    init();
    glutMainLoop();
    return 0;
}
```

OpenGL のデフォルト座標系 (左手系)



- カメラ位置を変更することにより、座標系を変える事は出来る。
- gluLookAt (右手系)

gltest2.c

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

```
void
display(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    //glColor3f(1.0f, 0.0f, 0.0f);
    glBegin(GL_POLYGON);
    glColor3f(1.0f, 0.0f, 0.0f);
    glVertex2f(-0.9f, -0.9f);
    glColor3f(0.0f, 1.0f, 0.0f);
    glVertex2f( 0.9f, -0.9f);
    glColor3f(0.0f, 0.0f, 1.0f);
    glVertex2f( 0.9f,  0.9f);
    glColor3f(1.0f, 1.0f, 0.0f);
    glVertex2f(-0.9f,  0.9f);
    glEnd();
    glFlush();
}
```

```
void
init(void)
{
    //glClearColor(1.0, 1.0, 1.0, 0.0);
    glClearColor(0.0, 0.0, 0.0, 0.0);
}

int
main(int ac, char **av)
{
    glutInit(&ac, av);
    glutInitDisplayMode(GLUT_RGBA);
    glutCreateWindow(av[0]);
    glutDisplayFunc(display);
    init();
    glutMainLoop();
    return 0;
}
```