

情報可視化論

2015 年前期

陰山 聡

神戸大学 システム情報学研究科 計算科学専攻

2015.04.21

【重要連絡】 教室変更

- 来週（28日）より情報基盤センター分館第1演習室
- ただし6/9のみ情報基盤センター分館第2演習室

OpenGL の歴史

OpenGL とは

OpenGL 2.X は、OpenGL 1.X の機能を全て含む。

OpenGL 3.0 で上位互換性放棄

シェーダで置き換え可能な機能

OpenGL 3.0 非推奨機能

→ OpenGL 3.1 拡張機能

→ OpenGL 3.2 互換プロファイル

OpenGL Overview

<http://www.opengl.org>

OpenGL was first created as an open and reproducible alternative to Iris GL
...Silicon Graphics workstations... OpenGL 1.0 ...non-SGI 3rd party...

...

OpenGL 2.0...OpenGL Shading Language (also called GLSL), a C like language
with which the transformation and fragment shading stages of the pipeline can be
programmed.

OpenGL 3.0 adds the concept of deprecation (非推奨)... GL 3.1 removed most
deprecated features, and GL 3.2 created the notion of core and compatibility
OpenGL contexts.

Official versions of OpenGL released to date are 1.0, 1.1, 1.2, 1.2.1, 1.3, 1.4, 1.5,
2.0, 2.1, 3.0, 3.1, 3.2, 3.3, 4.0, 4.1, 4.2, 4.3, 4.4, 4.5

OpenGL 関係

<http://www.khronos.org>

- OpenGL 4.5
- OpenGL SL 4.5 (シェーディング言語)
- OpenGL ES 3.1 (組み込みシステム用) iOS, Android, Symbian
- WebGL 1.0.2 (\approx OpenGL ES 2.0)

Shader-based OpenGL による OpenGL 教育

OpenGL はグラフィックスの API として長年教育現場で教えられてきた。仕様の变化も少なかったので、長年同じ教科書、ノートが使われてきた。ところが OpenGL 3.0 から、OpenGL は大きく変化 (Shader-base 化)。それに伴って教育内容も大改訂が必要となった。

OpenGL ver. 3.1 からの大きな変化

全てのアプリケーションは少なくとも一つの頂点シェーダと、少なくとも一つのフラグメントシェーダを与える必要がある。

OpenGL ver. 3.1 からの大きな変化

```
//  
// Hello_World.c  
//  
// To compile on Mac.  
// gcc this_source.c -framework GLUT -framework OpenGL  
//  
#include <GLUT/glut.h>  
  
void display (void)  
{  
    glClearColor(GL_COLOR_BUFFER_BIT);  
    glBegin(GL_POLYGON);  
    glVertex2f(-0.5, -0.5);  
    glVertex2f(-0.5, 0.5);
```

```
glVertex2f( 0.5, 0.5);  
glVertex2f( 0.5, -0.5);  
glEnd();  
glutSwapBuffers();  
}  
  
int main(int argc, char **argv)  
{  
    glutInit(&argc, argv);  
    glutInitDisplayMode(GLUT_RGBA | GLUT_DOUBLE);  
    glutCreateWindow(" Hello World");  
    glutDisplayFunc(display);  
    glutMainLoop();  
}
```

Listing 1: 古い OpenGL コード

はじめに

OpenGL シェーディング言語 (OpenGL SL, GLSL) 4.0

GPU を使うための言語

CG ソースコード = OpenGL ソースコード
+ GLSL(フラグメントシェーダ) ソースコード
+ GLSL(頂点シェーダ) ソースコード

OpenGL シェーディング言語

GPU 内部で走る単体のプログラム

専用言語

シェーディングプログラム（あるいは単にシェーダ）と呼ばれる

要するに数値演算プログラムの一種

シェーディングに限らず様々な数値処理が可能（→ GPGPU）

並列処理

GPUのプロセッサ数

GeForce GTX TITAN

シェーダプロセッサ “CUDA Cores” 2688 個

以前は**固定機能パイプライン**

現在はプログラマブル

固定機能パイプラインは廃止予定

グラフィックス

- immediate-mode
 - 即時モード API
 - 描画のたびにシーン全体を GPU に送る。たとえ変更がなくても。
- retained-mode
 - 保持モード API
 - シーン全体をまず GPU に送る。その後は変更部分だけを送る。

WebGL は即時モード API

OpenGL プロファイル

OpenGL 3.0

deprecation (非推奨) を導入 → 不要な関数を段階的に取り除く

OpenGL 3.2 より

コア プロファイルと互換プロファイル

プロファイルはウィンドウシステム API で選択

glBegin/glEnd は非推奨

標準的な行列 GL_MODELVIEW, GL_PROJECTION もコアプロファイルから削除。

OpenGL ES 2.0

OpenGL for Embedded Systems (組み込みシステム用)

Immediate モード API

glBegin/glEnd なし。頂点配列を使う。

シェータベース。シェータプログラムが必要。

アプリケーションプログラム： C/C++, Objective-C, Java

OpenGL ES 2.0 \approx WebGL

WebGL は JavaScript で呼ぶ。

HTML5 canvas

HTML5：第5世代のHTML

canvas: JavaScript で描画できる長方形領域

→ **【HTML5 canvas サンプルプログラム】**