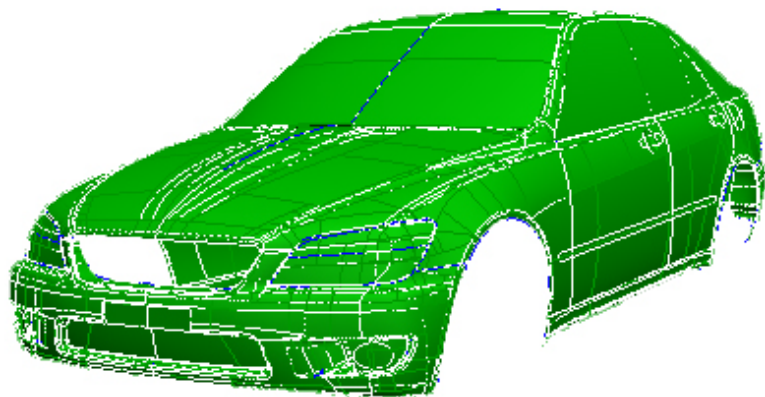


# 情報工学

2022年度後期CG 第1回 [10月5日]

---



静岡大学大学院

創造科学技術大学院

情報科学専攻

工学研究科機械工学専攻

ロボット・計測情報分野

三浦 憲二郎

# 講義アウトライン [10月5日]

---

- 連絡先
- シラバス
- コンピュータグラフィックスの概要
- CGプログラミング入門:C言語によるプログラミング
  - OpenGL

# 情報工学の現状

---

- 人工知能
- シングルコアからマルチコア
- 逐次処理から並列処理へ
- GPU (Graphics Processing Unit)コンピューティング
  - CUDA
  - OpenCL
- 10倍から1000倍のスピードアップ

# 連絡先

---

•電子メールアドレス(静岡大) [miura.kenjiro@shizuoka.ac.jp](mailto:miura.kenjiro@shizuoka.ac.jp)

•ホームページ

<http://mc2-lab.com/>

電話・ファックス

053-478-1074

•授業用ホームページ

<https://mc2-lab.com/lecture.html#B>

•講義室 オンデマンド

# 主題と目標

---

- **目標: C言語を用いた基礎的なグラフィックスプログラミング**

グラフィックス  
プレゼンテーション  
ビジュアライゼーション 等の道具

グラフィックスを有用な道具として使いこなせるように学習を進める。

講義：C言語について復習するとともに、CGについて学習する。  
さらに、形状モデリングの基礎について学ぶ。

# 授業計画 (1-4回)

---

- ・ 第1週[10月5日]

授業の概要。デジタルカメラモデル、グラフィックスライブラリー  
OpenGLの概要。OpenGLプログラムの作成とコンパイル、実行。

- ・ 第2週[10月12日]

座標系とモデリング，ビジュアル情報処理の幾何学的モデル

- ・ 第3週[10月19日]

3D座標系、3D図形の表現とその表示、移動、拡大・縮小について学ぶ。同次座標系を用いた幾何変換についても簡単な解説を行う。

- ・ 第4週[10月26日]

投影変換，いろいろな座標系と変換

# 授業計画 (5-7回)

---

・第5週[11月2日]

モデリング、メニューの作成

・第6週[11月9日]

照明モデルと照光処理

・第7週[12月14日]

期末試験

# 関連情報（その1）

---

<教科書、参考書、資料>

教科書：「ビジュアル情報処理」 CG-ARTS協会

参考書：「OpenGL 3Dグラフィックス入門」三浦憲二郎, 朝倉書店

資料： <https://mc2-lab.com/lecture.html#B>



# 関連情報（その2）

---

## <授業進行>

講義と演習。

授業ごとにプログラミングの課題を提出してもらう。

## <評価、期末試験>

各週の課題

期末試験（筆記試験）を行う。

# ビジュアル情報処理

## 1-1-1 ビジュアル情報処理

■ 図1.1——拡張現実感の例



[a] 室内の実写画像



[b] 実写画像にCGで仮想的な物体(植物, 壺など)を重ね合わせた例

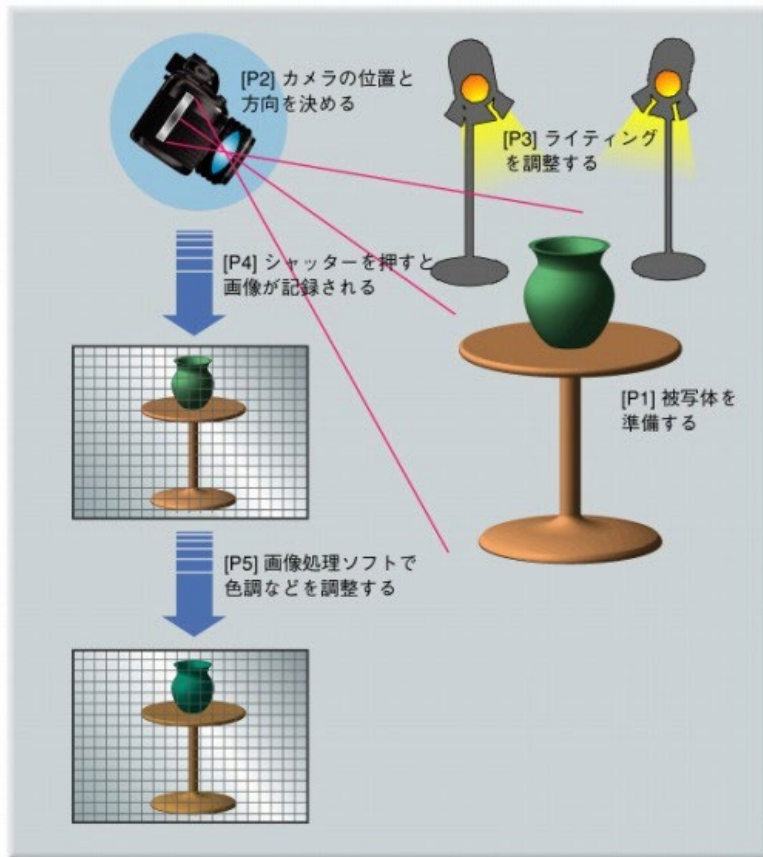
(©2000 株式会社エム・アール・システム研究所)

「ビジュアル情報処理」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

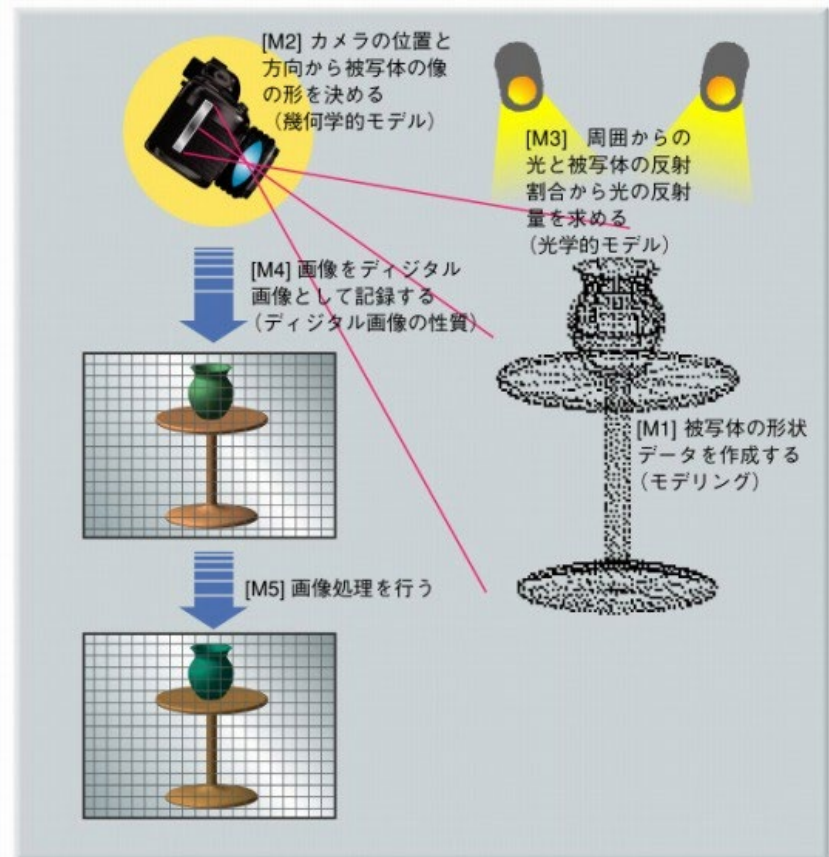
# デジタルカメラモデル

## 1-1-2 デジタルカメラモデル

■ 図1.2——デジタルカメラでの撮影過程



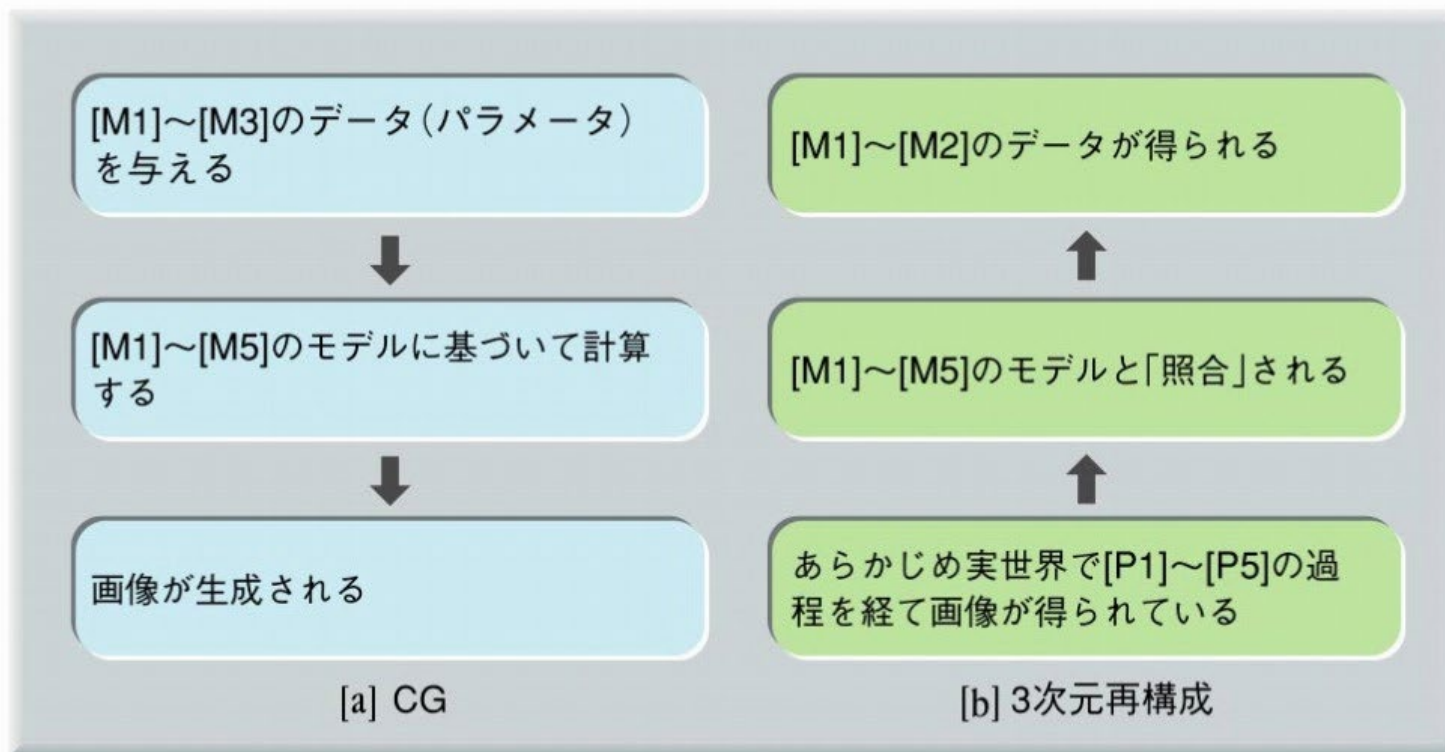
■ 図1.3——デジタルカメラモデル



# デジタルカメラモデル

## 1-1-2 デジタルカメラモデル

■ 図1.4——デジタルカメラモデルにおけるCGと画像処理（3次元再構成）の違い



# OpenGL

---

## 3D グラフィックスライブラリ

高品位な画像生成のためのプログラミングインターフェイス

GL (SGIのグラフィック用WSのグラフィックライブラリ)から発展

1992年 OpenGL Version 1.0

2004年 OpenGL Version 2.0

2008年 OpenGL Version 3.0

2010年 OpenGL Version 4.0

2017年 OpenGL Version 4.6

# Why OpenGL?

---

**最もハードよりの 3DCG API (application PI)**

**多くのCADソフト, CGソフトの採用**

**ex. Maya, 3ds Max, CATIA その他多数**

# OpenGLの特徴

---

1) ウィンドウシステムから独立

Windows 10, 11

X Window

2) イベントに対する機能なし

インプット機能の欠如

他のユーティリティ, 例えば GLUT

3) ステート(状態)マシン

# OpenGLの基本機能

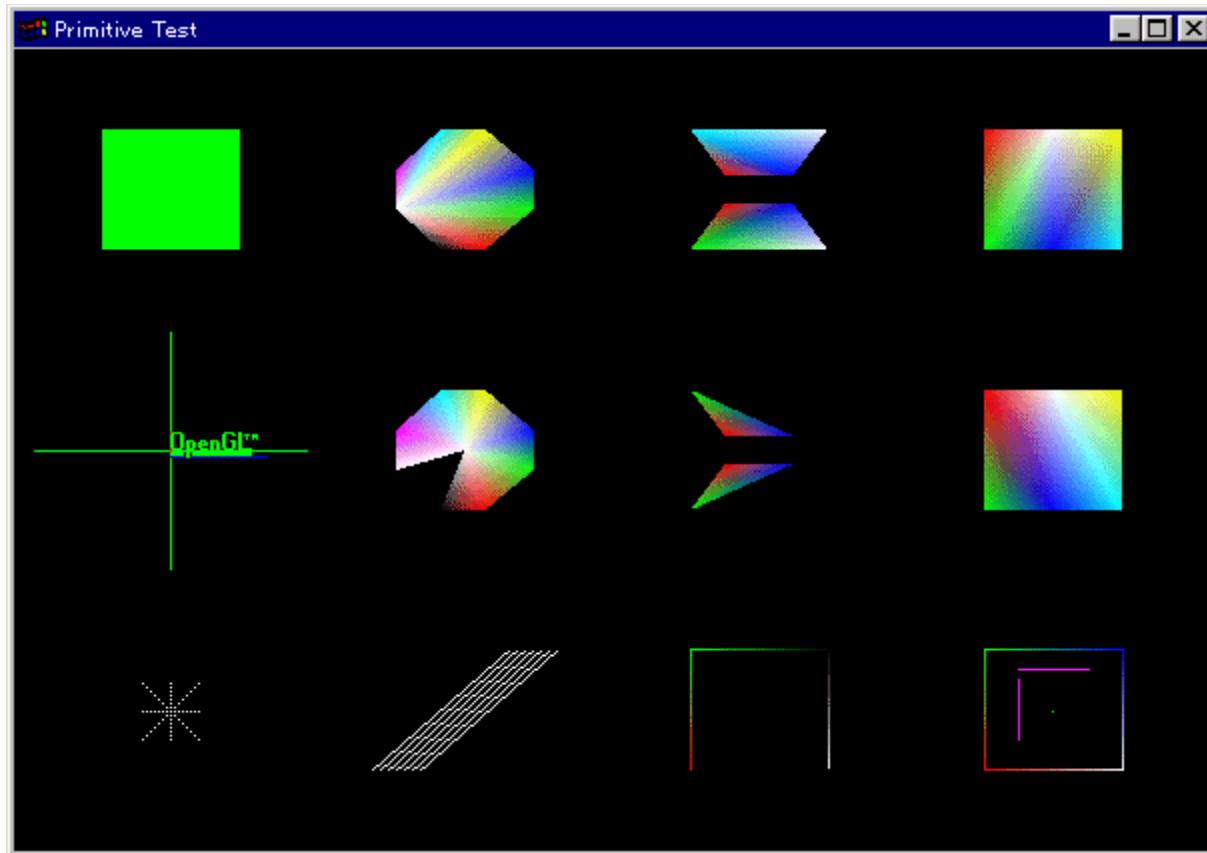
---

- 1) グラフィックスプリミティブの描画
- 2) 座標の変換
- 3) 色と照明
- 4) フレームバッファの処理
- 5) ディスプレイリスト
- 6) エバリュエータ
- 7) テクスチャマッピング
- 8) 色の変調・混合処理
- 9) セレクション機能



# OpenGLの基本機能

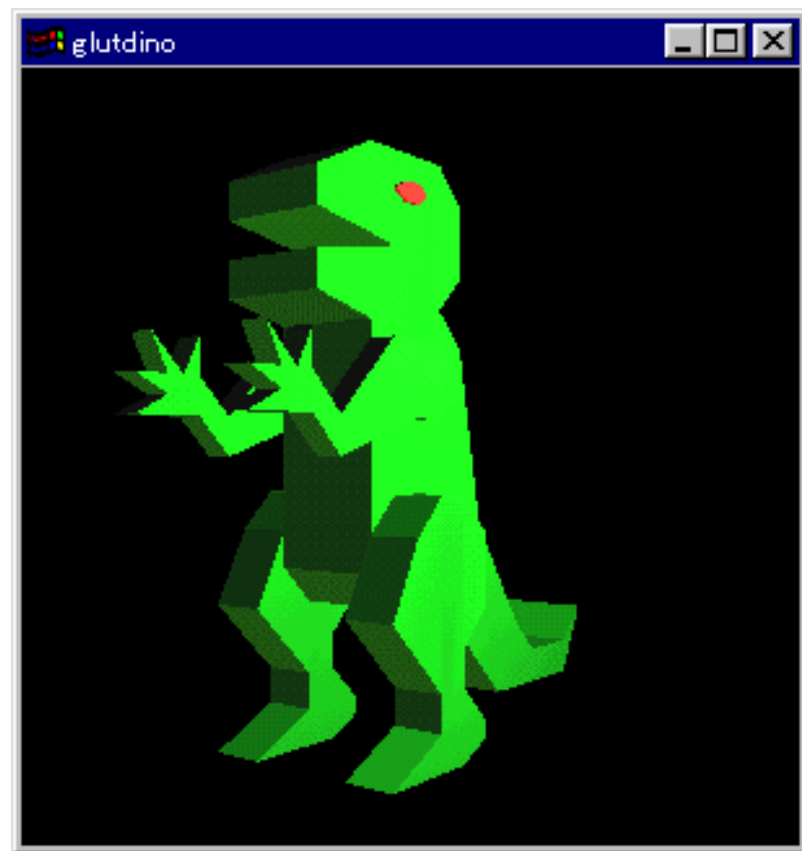
## 1) グラフィックスプリミティブの描画



# OpenGLの基本機能

---

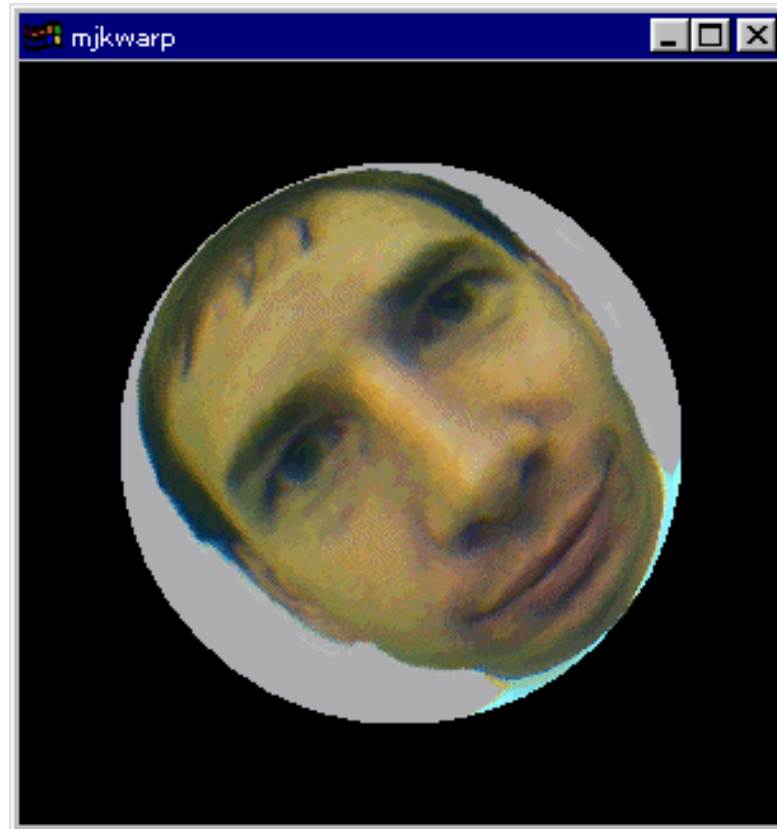
- 2) 座標の変換
- 3) 色と照明
- 4) フレームバッファの処理
- 5) ディスプレイリスト



# OpenGLの基本機能

---

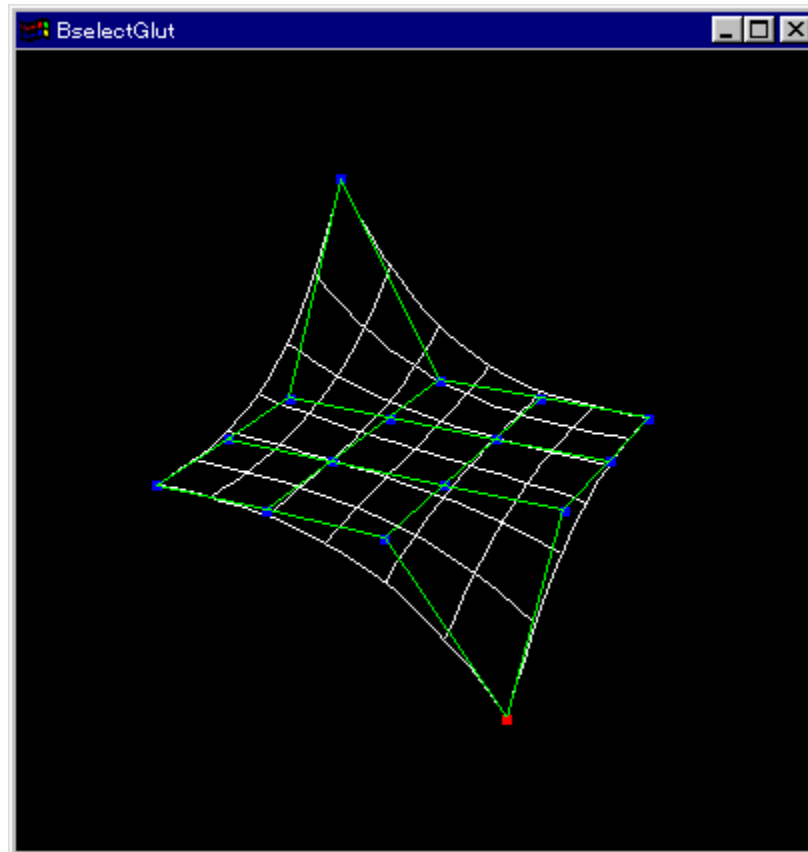
## 7) テクスチャマッピング



# OpenGLの基本機能

---

## 9) セレクション機能



# OpenGLの簡単なプログラム

---

```
#include <GL/glut.h>
void init(void)
{
    glClearColor ( 0.0, 0.0, 0.0, 0.0 ); /*背景色の指定*/
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-1.0, 1.0, -1.0, 1.0, -1.0, 1.0);
                                         /*描画のための投影法の指定*/
}
int main(int argc, char **argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB );
                                         /*表示モードの指定*/
    glutInitWindowPosition (400,400); /*画面の位置・大きさの指定*/
    glutCreateWindow ("sProgGlut" ); /*ウインドウのオープン*/
    init(); /*初期化处理*/
    glutDisplayFunc(display);
    gluMainLoop();
    return 0;
}
```

# OpenGLの簡単なプログラム

---

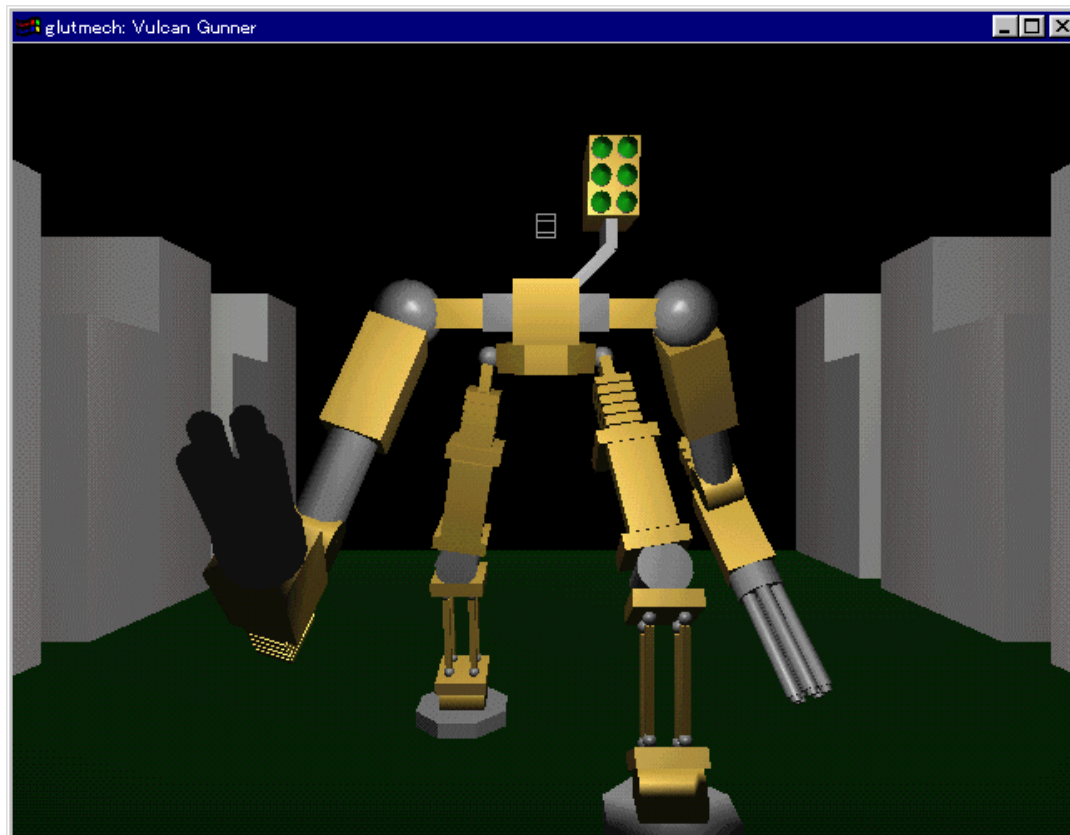
```
void display(void) {
    glClear(GL_COLOR_BUFFER_BIT); /* 背景のクリア */
    glColor3f(1.0, 1.0, 1.0); /* オブジェクトの色の指定 */
    glBegin(GL_POLYGON); /* 長方形の描画 */
        glVertex2f(-0.4, -0.2);
        glVertex2f(-0.4, 0.0);
        glVertex2f(-0.05, 0.0);
        glVertex2f(-0.05, -0.2);
    glEnd();
    glFlush(); /* 描画の強制 */
}
```

# 3D コンテンツ

---

OpenGLのサンプルプログラム

<http://www.opengl.org/>



# 課題

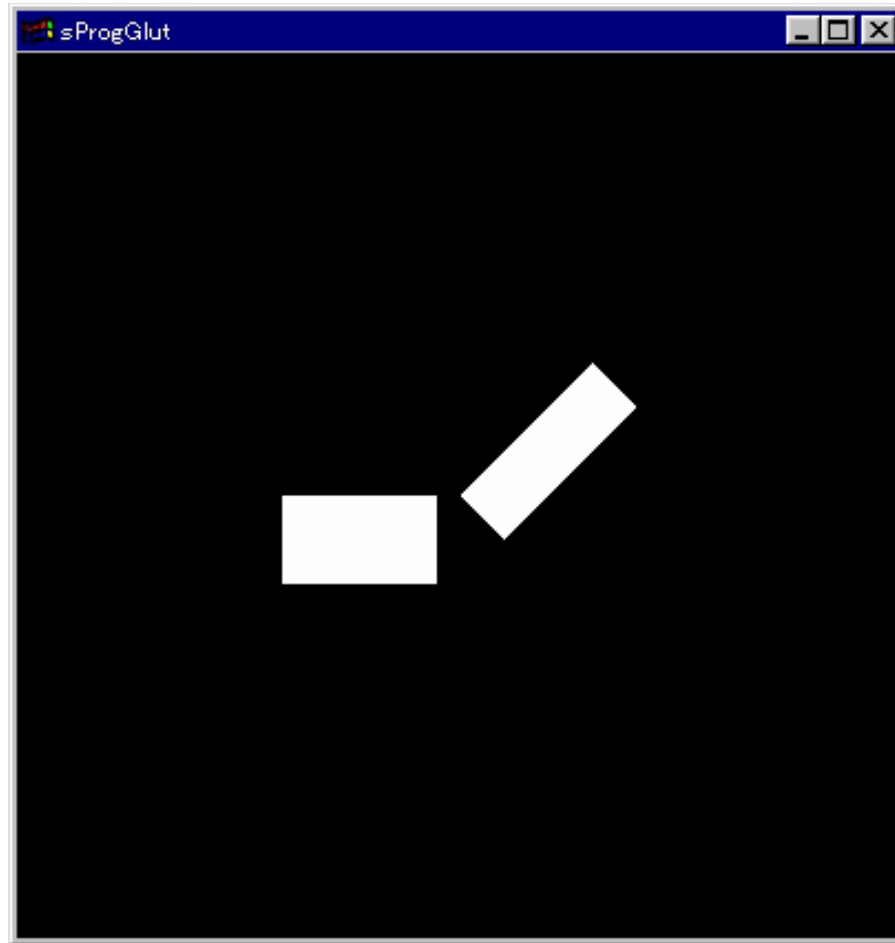
---

\*PC (Windows 10 or 11, Visual C++)



# simpleProg.c(実行結果)

---



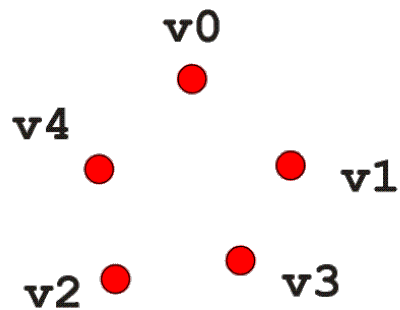
ソースコードは授業用ホームページを参照すること。

# simpleProg.c (display())

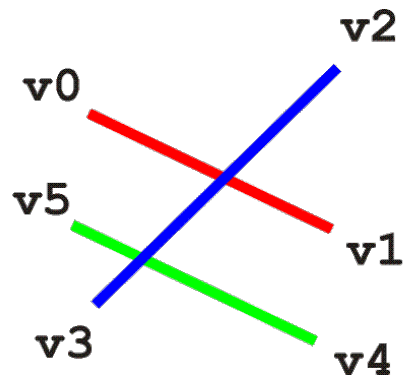
---

```
void display(void) {
    glClear(GL_COLOR_BUFFER_BIT); /*背景のクリア          */
    glColor3f(1.0, 1.0, 1.0);    /*オブジェクトの色の指定*/
    glBegin(GL_POLYGON);        /*長方形の描画          */
        glVertex2f(-0.4, -0.2);
        glVertex2f(-0.4, 0.0);
        glVertex2f(-0.05, 0.0);
        glVertex2f(-0.05, -0.2);
    glEnd();
    glFlush();                  /*描画の強制*/
}
```

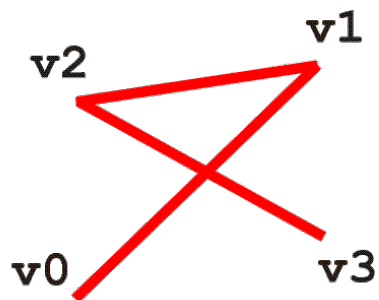
# プリミティブの描画(その1)



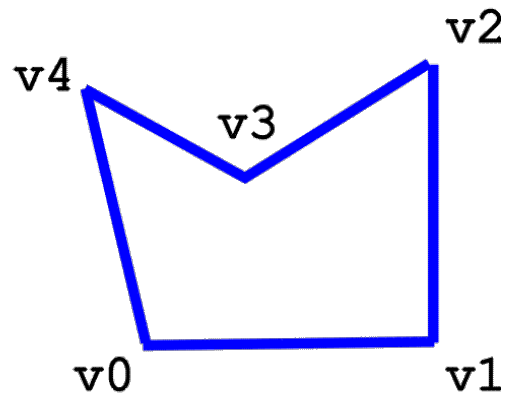
GL\_POINTS



GL\_LINES

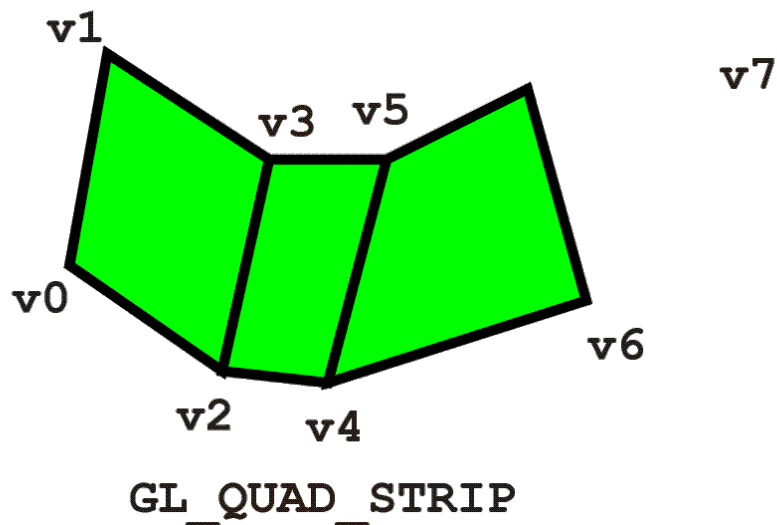
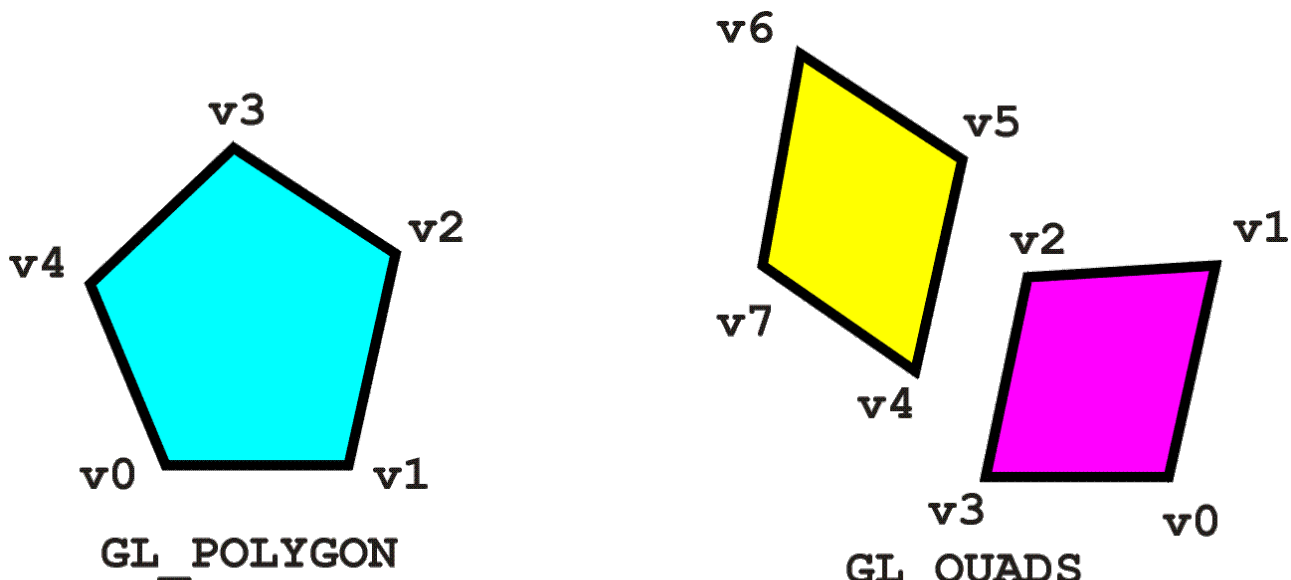


GL\_LINE\_STRIP

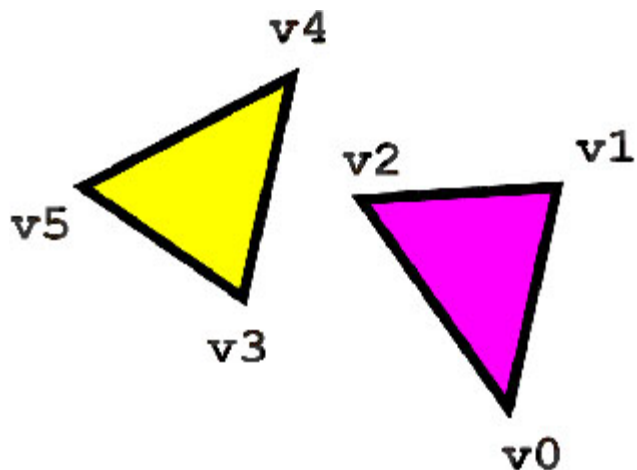


GL\_LINE\_LOOP

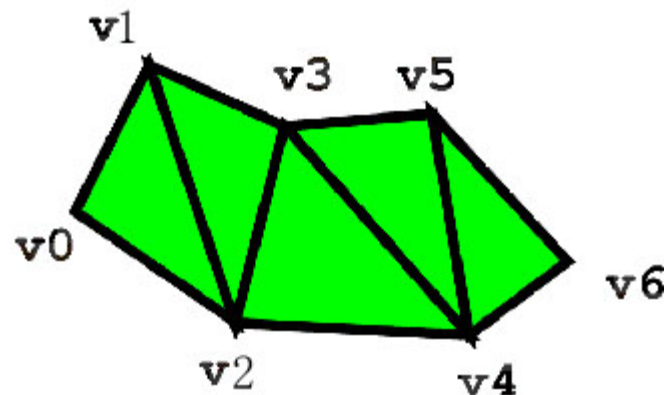
# プリミティブの描画(その2)



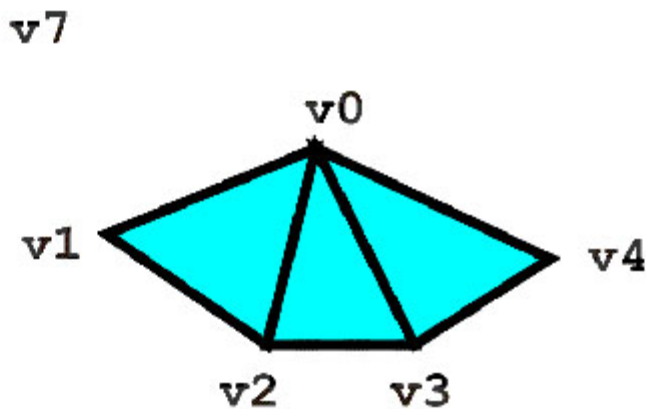
# プリミティブの描画(その3)



GL\_TRIANGLES



GL\_TRIANGLE\_STRIP



GL\_TRIANGLE\_FAN

# 課題1

---

令和4年10月5日（水曜日）

提出締め切り 令和4年10月11日（火曜日） 24 : 00

課題（2D図形の表示に関する課題）

サンプルプログラムsimpleProg.cをコンパイルして実行した後、glBegin()を以下のモード：

**GL\_LINES, GL\_QUADS, GL\_TRIANGLE\_STRIP**

や他のモードを引数として呼び出し、2次元図形を描画せよ。（各プリミティブの色は自由とする。）

提出物：解答プログラム（ソースコードのみ）

提出：watanabe.ryunosuke.18@shizuoka.ac.jp

提出方法：添付ファイル

# まとめ

---

- 連絡先
- シラバス
- コンピュータグラフィックスの概要
- デジタルカメラモデル
- C言語によるOpenGLプログラミング