CG基礎 第一回課題

# 1

* 01\_simple\_window.cpp
  + コード

#include <GLUT/glut.h> // ライブラリ用ヘッダファイルの読み込み

// 表示部分をこの関数で記入

void display(void)

{

glClearColor(1.0, 1.0, 1.0, 1.0); // 消去色指定

glClear(GL\_COLOR\_BUFFER\_BIT); // 画面消去

/\* ここに描画に関するプログラムコードを入れる \*/

glFlush(); // 画面出力

}

// メインプログラム

int main(int argc, char \*argv[])

{

glutInit(&argc, argv); // ライブラリの初期化

glutInitWindowSize(400, 400); // ウィンドウサイズを指定

glutCreateWindow(argv[0]); // ウィンドウを作成

glutDisplayFunc(display); // 表示関数を指定

glutMainLoop(); // イベント待ち

return 0;

}

* + 実行結果正方形

    低い精度で自動的に生成された説明
* 02\_draw\_lines.cpp
  + コード

#include <GLUT/glut.h> // ライブラリ用ヘッダファイルの読み込み

// 表示部分をこの関数で記入

void display(void)

{

glClearColor(1.0, 1.0, 1.0, 1.0); // 消去色指定

glClear(GL\_COLOR\_BUFFER\_BIT); // 画面消去

glColor3d(0.0, 0.0, 1.0); // 色指定(R,G,B)で0～1まで

glBegin(GL\_LINE\_LOOP); // 描画するものを指定

glVertex2d(-0.9, -0.8); // 頂点位置の指定(1つめ)

glVertex2d(0.9, -0.8); // 頂点位置の指定(2つめ)

glVertex2d(0.0, 0.8); // 頂点位置の指定(3つめ)

glEnd();

glFlush(); // 画面出力

}

// メインプログラム

int main(int argc, char \*argv[])

{

glutInit(&argc, argv); // ライブラリの初期化

glutInitWindowSize(400, 400); // ウィンドウサイズを指定

glutCreateWindow(argv[0]); // ウィンドウを作成

glutDisplayFunc(display); // 表示関数を指定

glutMainLoop(); // イベント待ち

return 0;

}

* + 実行結果図形 が含まれている画像

    自動的に生成された説明
* 03\_draw\_triangle.cpp
  + コード

#include <GLUT/glut.h> // ライブラリ用ヘッダファイルの読み込み

// 表示部分をこの関数で記入

void display(void)

{

glClearColor(1.0, 1.0, 1.0, 1.0); // 消去色指定

glClear(GL\_COLOR\_BUFFER\_BIT); // 画面消去

glColor3d(0.0, 1.0, 1.0); // 色指定(R,G,B)で0～1まで

glBegin(GL\_TRIANGLES); // 描画するものを指定

glVertex2d(-0.9, -0.8); // 頂点位置の指定(1つめ)

glVertex2d(0.9, -0.8); // 頂点位置の指定(2つめ)

glVertex2d(0.0, 0.8); // 頂点位置の指定(3つめ)

glEnd();

glFlush(); // 画面出力

}

// メインプログラム

int main(int argc, char \*argv[])

{

glutInit(&argc, argv); // ライブラリの初期化

glutInitWindowSize(400, 400); // ウィンドウサイズを指定

glutCreateWindow(argv[0]); // ウィンドウを作成

glutDisplayFunc(display); // 表示関数を指定

glutMainLoop(); // イベント待ち

return 0;

}

* + 実行結果図形

    自動的に生成された説明
* 04\_draw\_figures.cpp
  + コード

#include <GLUT/glut.h> // ライブラリ用ヘッダファイルの読み込み

// 表示部分をこの関数で記入

void display(void)

{

glClearColor(1.0, 1.0, 1.0, 1.0); // 消去色指定

glClear(GL\_COLOR\_BUFFER\_BIT); // 画面消去

// 1つ目の図形

glColor3d(1.0, 0.5, 0.0); // 色指定(R,G,B)で0～1まで

glBegin(GL\_QUADS); // 描画するものを指定

glVertex2d(-0.3, 0.0); // 頂点位置の指定(1つめ)

glVertex2d(-0.3, -0.4); // 頂点位置の指定(2つめ)

glVertex2d(0.3, -0.4); // 頂点位置の指定(3つめ)

glVertex2d(0.3, 0.0); // 頂点位置の指定(4つめ)

glEnd();

// 2つ目の図形

glColor3d(0.0, 1.0, 0.0); // 色指定(R,G,B)で0～1まで

glBegin(GL\_TRIANGLES); // 描画するものを指定

glVertex2d(0.0, 0.9); // 頂点位置の指定(1つめ)

glVertex2d(-0.7, 0.0); // 頂点位置の指定(2つめ)

glVertex2d(0.7, 0.0); // 頂点位置の指定(3つめ)

glEnd();

glFlush(); // 画面出力

}

// メインプログラム

int main(int argc, char \*argv[])

{

glutInit(&argc, argv); // ライブラリの初期化

glutInitWindowSize(400, 400); // ウィンドウサイズを指定

glutCreateWindow(argv[0]); // ウィンドウを作成

glutDisplayFunc(display); // 表示関数を指定

glutMainLoop(); // イベント待ち

return 0;

}

* + 実行結果図形

    自動的に生成された説明
* 05\_for\_loop.cpp
  + コード

#include <GLUT/glut.h> // ライブラリ用ヘッダファイルの読み込み

// 表示部分をこの関数で記入

void display(void)

{

glClearColor(1.0, 1.0, 1.0, 1.0); // 消去色指定

glClear(GL\_COLOR\_BUFFER\_BIT); // 画面消去

glColor3d(0.0, 0.0, 0.0); // 色指定(R,G,B)で0～1まで

glBegin(GL\_LINES);

for (int i = 0; i < 10; i++)

{

glVertex2d(i \* 0.1 - 0.5, 0.5);

glVertex2d(i \* 0.1 - 0.5, -0.5);

}

glEnd();

glFlush(); // 画面出力

}

// メインプログラム

int main(int argc, char \*argv[])

{

glutInit(&argc, argv); // ライブラリの初期化

glutInitWindowSize(400, 400); // ウィンドウサイズを指定

glutCreateWindow(argv[0]); // ウィンドウを作成

glutDisplayFunc(display); // 表示関数を指定

glutMainLoop(); // イベント待ち

return 0;

}

* + 実行結果テキスト

    低い精度で自動的に生成された説明

# 2

* コード

#include <GLUT/glut.h> // ライブラリ用ヘッダファイルの読み込み

// 表示部分をこの関数で記入

void display(void)

{

glClearColor(1.0, 1.0, 1.0, 1.0); // 消去色指定

glClear(GL\_COLOR\_BUFFER\_BIT); // 画面消去

glColor3d(0.0, 0.0, 0.0); // 色指定(R,G,B)で0～1まで

glBegin(GL\_QUADS);

int slice = 10; //平面を均等に分割する数（1以上）

for (int i = 0; i <= (slice \* 2) - 1; i++)

{

for (int j = 0; j <= (slice \* 2) - 1; j++)

{

if (i % 2 == 0 && j % 2 == 0)

{

glVertex2d(-1.0 + (2.0 / (slice \* 2) \* i), 1.0 - (2.0 / (slice \* 2) \* j));

glVertex2d(-1.0 + (2.0 / (slice \* 2) \* i), 1.0 - (2.0 / (slice \* 2) \* (j + 1)));

glVertex2d(-1.0 + ((2.0 / (slice \* 2)) \* (i + 1)), 1.0 - (2.0 / (slice \* 2) \* (j + 1)));

glVertex2d(-1.0 + ((2.0 / (slice \* 2)) \* (i + 1)), 1.0 - (2.0 / (slice \* 2) \* j));

}

else if (i % 2 == 1 && j % 2 == 1)

{

glVertex2d(-1.0 + (2.0 / (slice \* 2) \* i), 1.0 - (2.0 / (slice \* 2) \* j));

glVertex2d(-1.0 + (2.0 / (slice \* 2) \* i), 1.0 - (2.0 / (slice \* 2) \* (j + 1)));

glVertex2d(-1.0 + ((2.0 / (slice \* 2)) \* (i + 1)), 1.0 - (2.0 / (slice \* 2) \* (j + 1)));

glVertex2d(-1.0 + ((2.0 / (slice \* 2)) \* (i + 1)), 1.0 - (2.0 / (slice \* 2) \* j));

}

}

}

glEnd();

glFlush(); // 画面出力

}

// メインプログラム

int main(int argc, char \*argv[])

{

glutInit(&argc, argv); // ライブラリの初期化

glutInitWindowSize(400, 400); // ウィンドウサイズを指定

glutCreateWindow(argv[0]); // ウィンドウを作成

glutDisplayFunc(display); // 表示関数を指定

glutMainLoop(); // イベント待ち

return 0;

}

* 実行結果背景パターン が含まれている画像

  自動的に生成された説明