情報工学実験Ⅱ

「グラフィックプログラミング基礎」

担当教員名：赤嶺 有平

提　出　日：２０１０年12月9日

学 籍 番 号：　０９５７０７B

氏　　名：　大城　佳明

課題１ 図形の描画

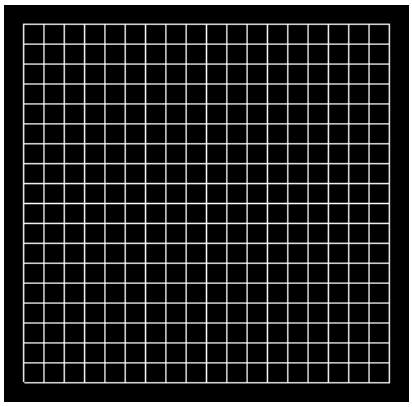
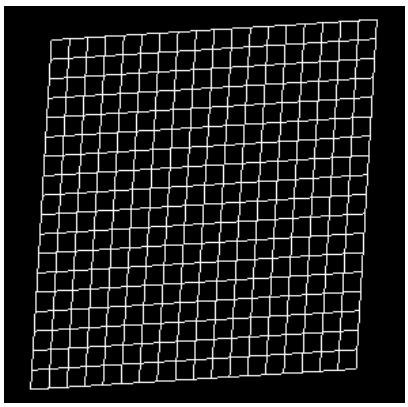
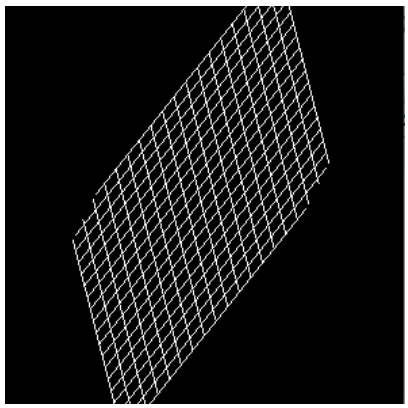
直線やポリゴン，座標変換を用いて，オリジナルの図形をアニメーション表示するプログラムを作成してください．

1. 18×18 のマス

<main.cpp>

```
01 void display(void)
02 {
03     glClearColor(0.0, 0.0, 0, 1.0);
04     glClear(GL_COLOR_BUFFER_BIT);
05     glMatrixMode(GL_PROJECTION);
06     glLoadIdentity();
07
08     static double t=0;
09     //z軸を中心に回転
10     //z軸は，原点からx-y平面に垂直な方向にのびているので
11     //原点を中心に回転する
12     glRotated(t += 0.01, 0.5, 1, 0);
13
14      double i;
15      //変数iの宣言
16      glBegin(GL_LINES);
17
18      //x軸に平行な線を18本引く(for文で18回線を引く動作を繰り返す)
19      for(i=-0.9; i<=0.9; i+=0.1){
20         glVertex2d(i, 0.9);
21         glVertex2d(i, -0.9);
22         }
23
24      //y軸に平行な線を18本引く
25      for(i=-0.9; i<=0.9; i+=0.1){
26         glVertex2d(0.9, i);
27         glVertex2d(-0.9, i);
28      }
29      glEnd();
30      glFlush();
31     printf("display\n");
32 }
```

<実行結果>

<解析>

1.03、04 は初期化である

2.06 は変換行列のリセットである

3.08～12 は回転させるための関数である

4.double 型の変数 i を宣言する。線を引くための引数となる。

5.16～29 は線を引く

A.19～22 は x 軸に平行な線を引く。

B.y の値は常に一定にし、x の値をインクリメントしている。

C.横に 19 本の線を引くことができる

D.25～28 は y 軸に平行な線を引く。

E.x の値は常に一定にし、y の値をインクリメントしている。

F.縦に 19 本の線を引くことができる

6.縦に 19 本、横に 19 本の線を引くことにより、18×18 のマスを作ることができた

7.30 行目で終わらせている

2. ピラミッド型

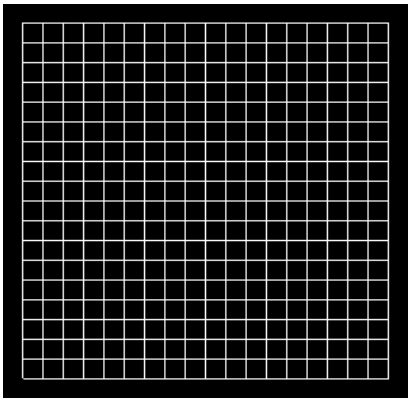
18×18 のマスを作るだけでは足りないと思ったので、付け加えた。
<main.cpp>の 30 行目から付け加えた。

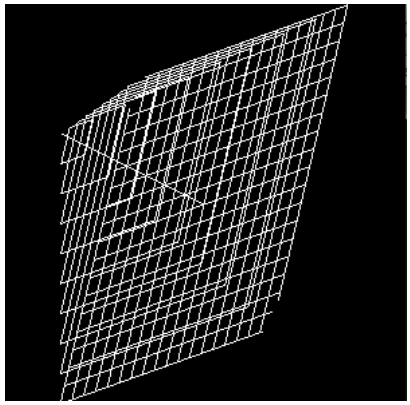
<main.cpp>(追加)

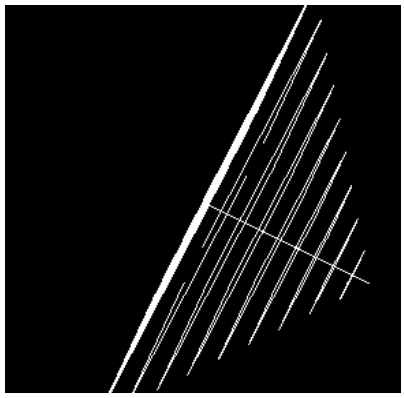
```
29      glEnd();
30      double j;
31      //変数jの宣言
32
```

```
33      //正方形をz軸方向からだんだん大きくして描く
34      for (j=0; j<=0.9; j+=0.1) {
35        //中で宣言することにより、独立した正方形を描く
36        glBegin(GL_LINE_LOOP);
37        glVertex3d(-j, -j,j-0.9);
38        glVertex3d(j, -j,j-0.9);
39        glVertex3d(j, j,j-0.9);
40        glVertex3d(-j, j,j-0.9);
41        glEnd();
42      }
43
44      //中心に直線を引く
45      glBegin(GL_LINES);
46      glVertex3d(0,0,0);
47      glVertex3d(0,0,-0.9);
48      glEnd();
49      glFlush();
50     printf("display\n");
51 }
```

<実行結果>







<解析>

1.30 行目は double 型の変数 j の宣言である。線を引くための変数である。

2.34～42 は正方形を引く

A.z 軸方向からも線を引く

B.奥側からだんだん手前に正方形が大きくなるようにした

C.線の種類は for 文の中で宣言してある。

D.for 文の中で線を独立的に引く。

3.正方形を書くことで、四角錐を描くことができる

4.正方形だけでは物足りないので、45～48 は中心に線をのばすため書いた

課題２：3DCGの作成

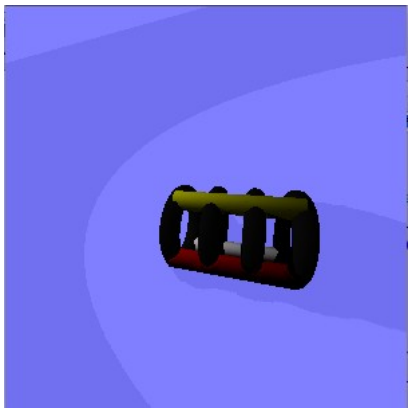
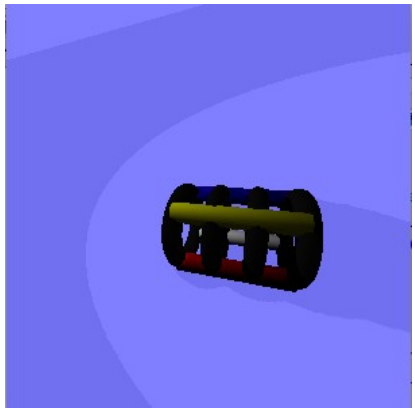
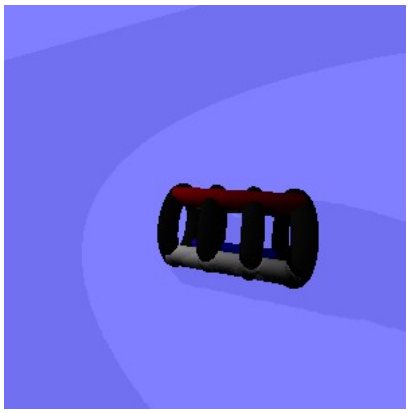
RGEのプロジェクトのアクティブターゲットを「exercise」に変更し，
exercise.cppのcreateScene関数を編集してオリジナルの3DCGを作成せよ．

<exercise.cpp>

```
01 void createScene()
02 {
03     FrameRef root_1 =  g_rge.rootFrame();
04     //grid座標系にroot_1座標系を作成
05     //全体の中心をroot_1とする
06
07     FrameRef orbit1 = root_1->createChild(); //root_1を中心としたorbit1を作成
08     orbit1->setRotation(90,90,90);   //x,y,z軸に90度ずつ回転(ドーナツ型を縦で表現するため)
09
10     orbit1->setTranslation(4,0,0);   //中心よりx方向に4だけずらす
11     orbit1->addRenderable(g_rge.findMesh("Torus"));
12     orbit1->addMaterial(g_rge.findMaterial("black"));
13     //ドーナツ型の黒色を設定した
14
15     orbit1->setScale(4,4,4);        //x,y,zそれぞれの軸の方向に4倍ずつ拡大した
16
17
18     FrameRef orbit2 = root_1->createChild(); //root_1を中心としたorbit2を作成
19     orbit2->setRotation(90,90,90);   //ドーナツ型を縦で表現するため90度ずつ回転
20
21     //ドーナツ型の黒色の図形を他にも３つ用意する
22     //それぞれ軸を等間隔に配置する
23     FrameRef gem1 = orbit2->createChild(); //orbit2を中心としたgem1を作成
24     gem1->addRenderable(g_rge.findMesh("Torus"));
25     gem1->addMaterial(g_rge.findMaterial("black"));
26     //ドーナツ型の黒色を設定した
27     gem1->setScale(4,4,4);        //x,y,zそれぞれの軸の方向に4倍ずつ拡大
28
29     FrameRef gem2 = orbit2->createChild();
30     gem2->setTranslation(0,0,8);
31     gem2->addRenderable(g_rge.findMesh("Torus"));
32     gem2->addMaterial(g_rge.findMaterial("black"));
33     gem2->setScale(4,4,4);
34
35     FrameRef gem3 = orbit2->createChild();
36     gem3->setTranslation(0,0,12);
37     gem3->addRenderable(g_rge.findMesh("Torus"));
38     gem3->addMaterial(g_rge.findMaterial("black"));
39     gem3->setScale(4,4,4);
40
41
42     //円周上を回転する円柱を４つ作成する
43     FrameRef orbit3 = root_1->createChild(); //root_1を中心としてorbit3を作成
44     orbit3->setRotation(90,90,90);   //全体と合わせるためにx,y,z軸を90度ずつ回転
45     orbit3->setAnglerVelo(0,500,0); //回転速度を500としてy軸中心に回転させる
```

```
46      orbit3->setScale(1,1,6);        //y軸方向に6倍拡大し、細長い円柱を作る
47
48      //4つの円柱の作成
49      FrameRef g1 = orbit3->createChild(); //orbit3を中心にg1を作成
50      g1->setTranslation(3,3,1);  //中心の軸をドーナツ型の円周上を回転するようにに設置
51      g1->addRenderable(g_rge.findMesh("Cylinder"));
52      g1->addMaterial(g_rge.findMaterial("red"));
53      //赤色の円柱を作成した
54
55      FrameRef g2 = orbit3->createChild();
56      g2->setTranslation(3,-3,1); //g1と被らない軸に設置
57      g2->addRenderable(g_rge.findMesh("Cylinder"));
58      g2->addMaterial(g_rge.findMaterial("yellow"));
59      //黄色の円柱を作成した
60
61      FrameRef g3 = orbit3->createChild();
62      g3->setTranslation(-3,3,1);
63      g3->addRenderable(g_rge.findMesh("Cylinder"));
64      g3->addMaterial(g_rge.findMaterial("white"));
65      //白色の円柱を作成した
66
67      FrameRef g4 = orbit3->createChild();
68      g4->setTranslation(-3,-3,1);
69      g4->addRenderable(g_rge.findMesh("Cylinder"));
70      g4->addMaterial(g_rge.findMaterial("blue"));
71      //青色の円柱を作成した
72 }
```

<実行結果>

<解析>

1.03～05は全体の軸の作成

2.07～39はドーナツ型の作成

A.08では x,y,z 軸を 90 度回転させる。

B.90 度回転させることにより、ドーナツ型を縦に表現する

C.23～27 で原点を中心とした図の作成

D.29～33、35～39 でもドーナツ型を作成した

3.41～72 でドーナツ型の円周を回る円柱を作成

A.43 で新しく orbit3 を作成

B.45 で円柱の回転をさせる。Z 軸を中心に 500 の早さで回転する

C.46 では円柱を縦に細長くした。

D.49～52 で円柱を作成。軸を(3,3,1)に移動することにより、ドーナツ型の円周を回る

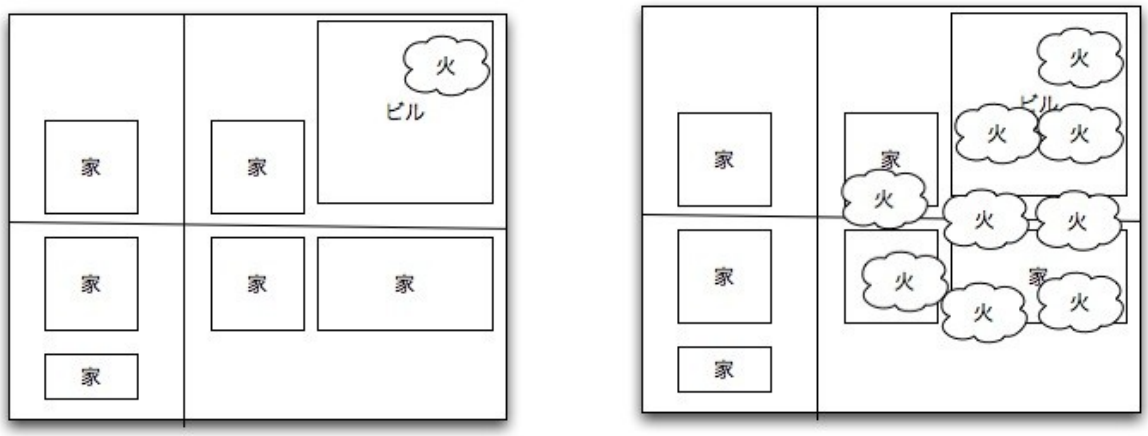
E.同様に 55～59、61～69、67～72 で円柱を作成した

課題 3：ARToolKit の利用例

ARToolKit を利用して，オリジナルのアプリケーションのアイデアを「図」を使って説明せよ．（実際に作る必要はないが，「技術的」に開発可能かどうかの考察は必要・自身の開発スキルは関係ない．）

1.地震シミュレーション

1.住宅密集地帯や森林などの火災のシミュレーションが出来るのではないかと考えた。



2.図のように火が移り変わる瞬間の絵をシミュレーションする。

3.ビルや家などを AR で作り、建物には火への耐久度などを設定する。

4.風向きなどを考え、火の動く道のりを予想する。

5.火を適当な位置にセットすると燃え広がるとう仕組みにする。

6.実際にこのように出来るかを考えると、出来るのではないかと思った

6.マーカは他のマーカに影響を受けるようにできなかればならない。

7.いろいろな条件が設定されるので条件分岐の数が多くなり処理が難しいのではとも考えた。

8.他のマーカに影響を与えることことができれば実現可能だと私は考える

2.オンラインショッピングでの活用

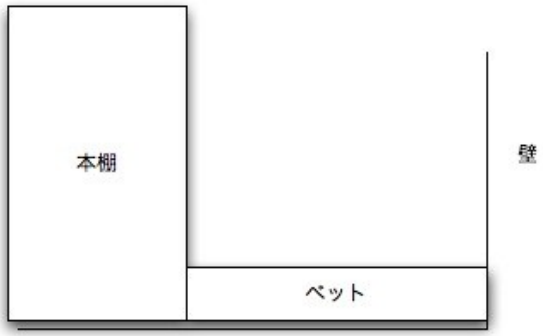
1.家庭用にオンラインのショッピングでの活用を考えた。

2.AR を使って物の大きさを実際に写すことでショッピングスタイルが変わるのではないかと考えた

3.例えば、本棚が欲しいとする。

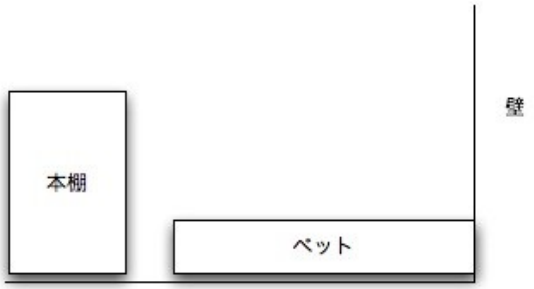
A.下の図の AR(マーカ)の位置に本棚が欲しいとする。

B.AR を使い、本棚を実際に映し出す



C.本棚の大きさがかなり多く感じる。

D.AR を使い違う本棚を表示する



E.ちょうどいいぐらいの本棚を見つけることができた

3.例のようにARをある商品と同じ大きさ、形、色まで表示することで買う時に楽になる

4.ARはiPhoneなどに映し出すことにより、いろいろなパターンの買い物を考えることができる

5.引っ越し時にも応用出来る(大きさが一目のでわかるので部屋に入るかどうかが一目同然)

6.実際に作るには大変な作業になるが不可能ではない

7.ひとつひとつの3Dを作る手間を省くことができるなら実用可能のではないかと考える