CSC 470 Computer Graphics Fall’16

Project 1

(submit a **printout of the source code and output attached** as well as **store the above files, exe files** and **explanations (doc file)). Please, make separate directories and subdirectories for each of the two problems and respective parts.**

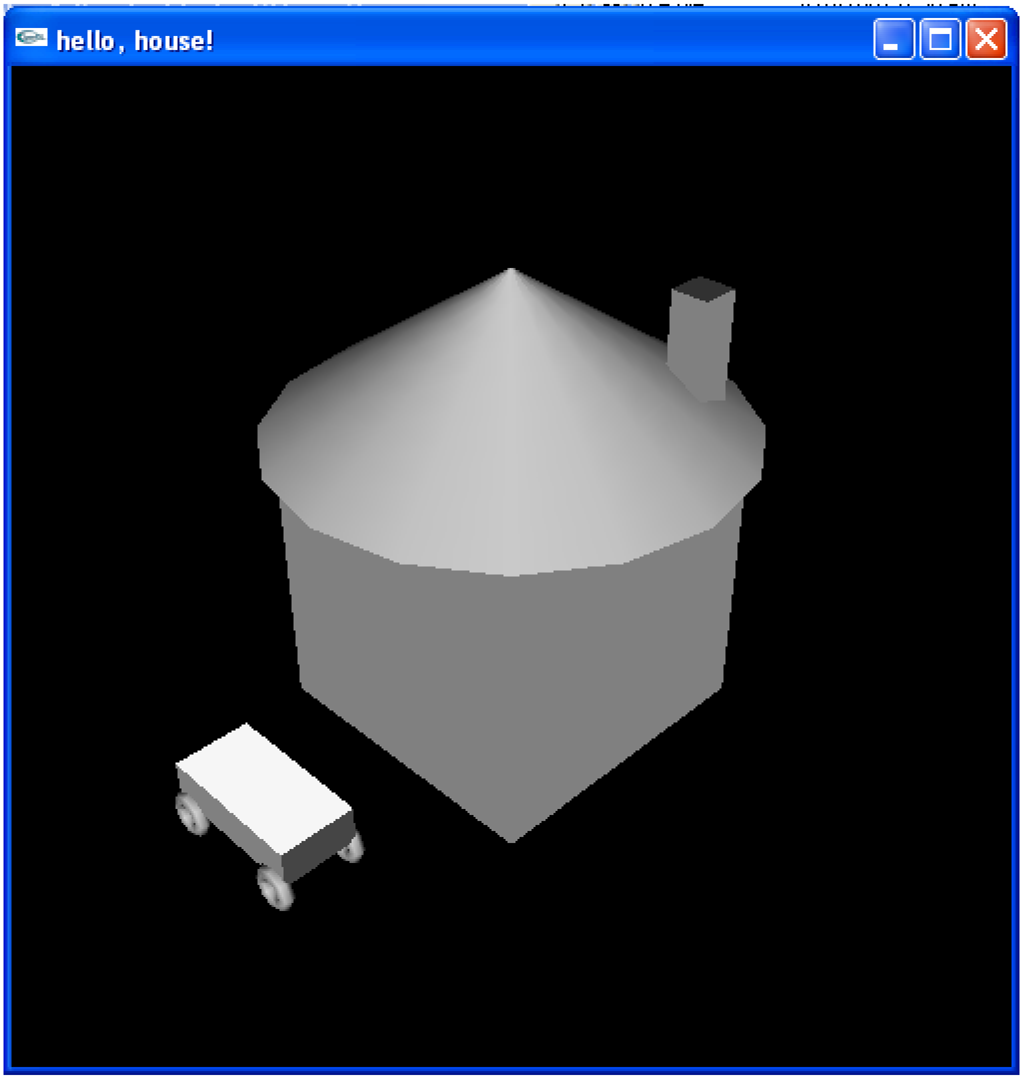
Problem 1.

1. Complete the source code shown below. The missing statements are indicated by dotted lines. These are statements Push( ), Pop( ) and necessary transformations. The required output is shown. **(15 points)**

Note: file house.cpp is on the web

```
// house.cpp

#include <windows.h>
#include <iostream.h>
#include <GL/glut.h>

void display () {

   /* clear window */
   glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

   /* future matrix manipulations should affect the modelview matrix */
   glMatrixMode(GL_MODELVIEW);

   /* draw scene */
   glPushMatrix();

   // house
   glPushMatrix();
   glutSolidCube(2);              // building

   .....
   ........              // roof
   ........
   glutSolidCone(1.5,1,16,8);
   ........
```

```
    ........
    .........                 // chimney
    .........
    glutSolidCube(.25);
    ........
    .......

    ........

    // car
    ........
    .......                  // body
    ........
    glutSolidCube(.5);
    .......
    ........
    ..........
    .........
    glutSolidTorus(.05,.1,8,8);       // wheel
    ........
    glutSolidTorus(.05,.1,8,8);       // wheel
    ........
    ........
    ........
    ........
    glutSolidTorus(.05,.1,8,8);       // wheel
    .........
    glutSolidTorus(.05,.1,8,8);       // wheel
    ...........
    glPopMatrix();

    glPopMatrix();

    /* flush drawing routines to the window */
    glFlush();
}

void reshape ( int width, int height ) {

    /* define the viewport transformation */
    glViewport(0,0,width,height);
}

int main ( int argc, char * argv[] ) {
    /* initialize GLUT, using any commandline parameters passed to the
       program */
    glutInit(&argc,argv);

    /* setup the size, position, and display mode for new windows */
    glutInitWindowSize(500,500);
    glutInitWindowPosition(0,0);
    glutInitDisplayMode(GLUT_RGB | GLUT_DEPTH);

    /* create and set up a window */
    glutCreateWindow("hello, house!");
    glutDisplayFunc(display);
    glutReshapeFunc(reshape);

    /* set up depth-buffering */
    glEnable(GL_DEPTH_TEST);

    /* turn on default lighting */
    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);

    /* define the projection transformation */
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(40,1,4,20);

    /* define the viewing transformation */
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    gluLookAt(5.0,5.0,5.0,0.0,0.0,0.0,0.0,1.0,0.0);

    /* tell GLUT to wait for events */
    glutMainLoop();
            return 0;
}
```



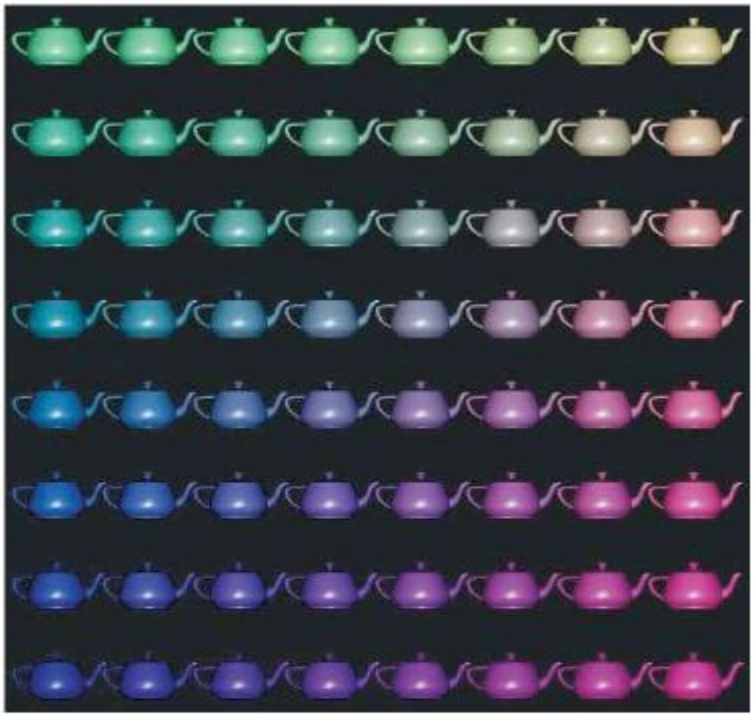
2. If you include 3 additional transformations to the above program and colors, you will get **up to 15 points more.**

3. If you add several different types of lighting and shading and different material properties, you will get additionally **up to 15 more points**.

4. If you add some environment around the scene, you will get **up to 10 more points**.

Problem 2:

1. Draw the teapots as in the figure below. You are given initial code in the attached file and need to rearrange it and add more to recreate the figure shown below. **(15 points)**



For **extra credit:**

2. You can make the teapots spin (each teapot in its place) – **15 points;**
3. Make the lights spin around all teapots – **15 points**.

Initial code for teapot Project1 problem.

```
#include <GL/glu.h>
#include <GL/glut.h>
#include <math.h>
#define W 600
#define H 600
void displaySolid(void)
{
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
glOrtho(-W, W, -H, H, -W, W);
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
gluLookAt(0,0,10,0,0,0,0.0,1.0,0.0);
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
glutSolidTeapot(300);
glFlush();
glutSwapBuffers();
}
int main(int argc, char** argv)
{
glutInit(&argc, argv);
glutInitDisplayMode(GLUT_DOUBLE|GLUT_RGB|GLUT_DEPTH);
glutInitWindowSize(W,H);
glutInitWindowPosition(100, 100);
glutCreateWindow("My Teapot");
glutDisplayFunc(displaySolid);
glClearColor(0.1f,0.1f,0.1f,0.0f);
glViewport(0,0, W, H);
glutMainLoop();
return 1;
}
```

Turning on the lights code to be placed WHERE???

```
glEnable(GL_LIGHTING);
glEnable(GL_LIGHT0);
glShadeModel(GL_SMOOTH);
glEnable(GL_DEPTH_TEST);
glEnable(GL_NORMALIZE);
```

Direction/intensity of the light.

```
GLfloat lightPosition[]={0.0f,0.0f,100.0f, 0.0f};
GLfloat lightIntensity[] = {0.9f, 0.9f, 0.9f, 1.0f};
glLightfv(GL_LIGHT0, GL_POSITION, lightPosition);
glLightfv(GL_LIGHT0, GL_DIFFUSE, lightIntensity);
```

Add ambient light.

Materials.

```
GLfloat mat_ambient[]= {0.5f, 0.5f, 0.6f, 1.0f};
GLfloat mat_diffuse[] = {0.6f, 0.6f, 0.6f, 1.0f};
GLfloat mat_specular[] = {1.0f, 1.0f, 1.0f, 1.0f};
GLfloat mat_shininess[] = {90.0f};
glMaterialfv(GL_FRONT,GL_AMBIENT,mat_ambient);
glMaterialfv(GL_FRONT,GL_DIFFUSE, mat_diffuse);
glMaterialfv(GL_FRONT,GL_SPECULAR, mat_specular);
glMaterialfv(GL_FRONT,GL_SHININESS, mat_shininess);
glutSolidTeapot(300);

GLfloat mat_ambient1[]= {0.5f, 0.5f, 0.6f, 1.0f};
GLfloat mat_diffuse1[] = {0.6f, 0.6f, 0.6f, 1.0f};
GLfloat mat_specular1[] = {1.0f, 1.0f, 1.0f, 1.0f};
GLfloat mat_shininess1[] = {90.0f};
glMaterialfv(GL_FRONT,GL_AMBIENT,mat_ambient1);
glMaterialfv(GL_FRONT,GL_DIFFUSE, mat_diffuse1);
glMaterialfv(GL_FRONT,GL_SPECULAR, mat_specular1);
glMaterialfv(GL_FRONT,GL_SHININESS, mat_shininess1);
glutSolidSphere(50,20,20);
```

NOTE: Max number of points for Project1 is 100 points.