CGV ASSIGNMENT

Interactive Mesh

Code:

```
#include <GL/glut.h>
#include <stdio.h>
#include <stdlib.h>
#define M 15
#define N 15

GLfloat theta[3];
GLfloat viewer[] = {0.5, 0.5, 0.5};
GLint y[15][15];
void display()
{
   glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
   glLoadIdentity();
   gluLookAt(viewer[0], viewer[1], viewer[2], 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);
   glRotatef(theta[0], 1.0, 0.0, 0.0);
   glRotatef(theta[1], 0.0, 1.0, 0.0);
   glRotatef(theta[2], 0.0, 0.0, 1.0);
   for (int i = 0; i < N - 1; i++)
   {
      for (int j = 1; j < M - 1; j++)
      {
         glBegin(GL_POLYGON);
```

```
            glColor3f(0.2, 0.6, 0.3);
            glVertex3i(i, y[i][j], j);
            glVertex3i(i + 1, y[i + 1][j], j);
            glVertex3i(i + 1, y[i + 1][j + 1], j + 1);
            glVertex3i(i, y[i][j + 1], j + 1);
            glEnd();
            glutSwapBuffers();
        }
    }
}

void keys(unsigned char k, int x, int y)
{
    if (k == 'x')
        viewer[0] -= 1.0;
    if (k == 'a')
        viewer[0] += 1.0;
    if (k == 'y')
        viewer[1] -= 1.0;
    if (k == 'b')
        viewer[1] += 1.0;
    if (k == 'z')
        viewer[2] -= 1.0;
    if (k == 'c')
        viewer[2] += 1.0;
    glutPostRedisplay();
```

```
}

void Reshape(int w, int h)
{

glViewport(0, 0, w, h);
   glMatrixMode(GL_PROJECTION);
   glLoadIdentity();
   if (w <= h)
   {
      glFrustum(-2.0, 2.0, -2.0 * (GLfloat)h / (GLfloat)w, 2.0 * (GLfloat)h /
(GLfloat)w, 2.0, 20.0);
   }
   else
   {
      glFrustum(-2.0, 2.0, -2.0 * (GLfloat)w / (GLfloat)h, 2.0 * (GLfloat)w /
(GLfloat)h, 2.0, 20.0);
   }
   glMatrixMode(GL_MODELVIEW);
}

int main(int argc, char **argv)
{

 glutInit(&argc, argv);
   glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
   glutInitWindowSize(980, 980);
```

```
    glutCreateWindow("Interactive Mesh");
    glutReshapeFunc(Reshape);
    glutDisplayFunc(display);
    glutKeyboardFunc(keys);
    glEnable(GL_DEPTH_TEST);
    glutMainLoop();
}
```

OUTPUT: