

SORU:

SORU 4) Adınızdaki blok harfleri taban poligon kabul eden ve sonra bu poligonu z yönünde $h=1$ birim yükselterek üç boyutlu üç harfi elde eden mesh dosyasını elde ediniz. Sonra bu mesh dosyasını okuyup çizdiren cpp uygulamasını yazınız. ('Ç' yerine 'C', 'Ğ' yerine 'G', 'İ' yerine 'I' ve 'Ş' yerine 'S' alabilirsiniz.).

CEVAP :

Program içerisinde GL_LINE fonksiyonu aktif edilebilir olduğundan çizgi olarak şekilde bağlantı noktalarının belirgin olması amacıyla daha fazla Vertex oluşturmaya rağmen bu noktalar özellikle çizilmiştir.

Dosya içerisindeki name.dat dosyası C:/ dizininde olmalıdır.

Tanımlamalar

```
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <GL/glut.h>
#include<iostream>
#include<fstream>

float x[240],y[240],z[240]; //dosyadan okunan değerlerin yazıldığı
değişkenler

GLfloat _x, _y, spin_x = 0, spin_y = 0;
```

reshape fonksiyonu

```
void reshape(int width, int height)
{
    glViewport(0, 0, width, height);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
```

```
gluPerspective(60, (GLfloat)width/height, 0.1,
1000.0);

glMatrixMode(GL_MODELVIEW);

glLoadIdentity();
```

Bakış açısı ayarı

```
gluLookAt(15, 0, 50, 15, 0, 0, 0, 1, 5);

glShadeModel(GL_FLAT);

glEnable(GL_DEPTH_TEST);
```

Başlangıç polygon mode ayarları

```
glPolygonMode(GL_FRONT, GL_FILL);

glPolygonMode(GL_BACK, GL_FILL);

}
```

display fonksiyonu

glRotatef fonksiyonu ile x ve y de spin_x ve spin_y değişkenleriyle döndürme tanımlanmıştır.

```
void display(void)
{
    int i, k, swap = 0;

    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    glPushMatrix();
```

İlk harf için başlangıç koordinatları

```
glTranslatef(-10/2, 0, -10*2);

glRotatef(spin_x, 0, 1, 0);

glRotatef(spin_y, 1, 0, 0);
```

C harfi çizimi 3 parça şeklide

C harfi birinci parça

```
glBegin(GL_POLYGON);

for(int i = 0; i<16; i++)

{
```

```
        glVertex3f(x[i], y[i], z[i]);
    }
    glEnd();
```

C harfi ikinci parça

```
glBegin(GL_POLYGON);
for(int i = 16; i<32; i++)
{
    glVertex3f(x[i], y[i], z[i]);
}
glEnd();
```

C harfi üçüncü parça

```
glBegin(GL_POLYGON);
for(int i = 32; i<48; i++)
{
    glVertex3f(x[i], y[i], z[i]);
}
glEnd();
glPopMatrix(); // C harfi popMatrix yapıldı.
```

A harfi çizimi 4 parça şeklinde

```
glPushMatrix();
```

glTranslatef ile x ekseninde 10 birim kaydırılmıştır.

```
glTranslatef(5, 0, -10*2);
glRotatef(spin_x, 0, 1, 0);
glRotatef(spin_y, 1, 0, 0);
```

A harfi birinci parça

```
glBegin(GL_POLYGON);
for(int i = 48; i<64; i++)
{
```

```
        glVertex3f(x[i], y[i], z[i]);
    }
    glEnd();
```

A harfi ikinci parça

```
glBegin(GL_POLYGON);
for(int i = 64; i<80; i++)
{
    glVertex3f(x[i], y[i], z[i]);
}
glEnd();
```

A harfi üçüncü parça

```
glBegin(GL_POLYGON);
for(int i = 80; i<96; i++)
{
    glVertex3f(x[i], y[i], z[i]);
}
glEnd();
```

A harfi dördüncü parça

```
glBegin(GL_POLYGON);
for(int i = 96; i<112; i++)
{
    glVertex3f(x[i], y[i], z[i]);
}
glEnd();

glPopMatrix(); //A harfi popMatrix yapıldı.
```

G Harfi çizimi 4 parça şeklinde

```
glPushMatrix();
```

glTranslatef ile x ekseninde 10 birim kaydırılmıştır.

```
glTranslatef(15, 0, -10*2);
```

```
glRotatef(spin_x, 0, 1, 0);  
glRotatef(spin_y, 1, 0, 0);
```

G harfi çizimi birinci parça

```
glBegin(GL_POLYGON);  
for(int i = 112; i<128; i++)  
{  
    glVertex3f(x[i], y[i], z[i]);  
}  
glEnd();
```

G harfi çizimi ikinci parça

```
glBegin(GL_POLYGON);  
for(int i = 128; i<144; i++)  
{  
    glVertex3f(x[i], y[i], z[i]);  
}  
glEnd();
```

G Harfi çizimi üçüncü parça

```
glBegin(GL_POLYGON);  
for(int i = 144; i<160; i++)  
{  
    glVertex3f(x[i], y[i], z[i]);  
}  
glEnd();
```

G Harfi çizimi dördüncü parça

```
glBegin(GL_POLYGON);  
for(int i = 160; i<176; i++)  
{  
    glVertex3f(x[i], y[i], z[i]);  
}
```

```
}  
glEnd();
```

```
glPopMatrix(); //G harfi popMatrix yapıldı.
```

I harfi çizimi 1 parça şekilde

```
glPushMatrix();
```

glTranslatef ile x ekseninde 10 birim kaydırılmıştır.

```
glTranslatef(25, 0, -10*2);  
glRotatef(spin_x, 0, 1, 0);  
glRotatef(spin_y, 1, 0, 0);
```

I harfi çizimi birinci parça

```
glBegin(GL_POLYGON);  
for(int i = 176; i<192; i++)  
{  
    glVertex3f(x[i], y[i], z[i]);  
}  
glEnd();  
glPopMatrix(); //I harfi popMatrix yapıldı.
```

N harfi çizimi 3 parça şeklinde

```
glPushMatrix();
```

glTranslatef ile x ekseninde 5 birim kaydırılmıştır.

```
glTranslatef(30, 0, -10*2);  
glRotatef(spin_x, 0, 1, 0);  
glRotatef(spin_y, 1, 0, 0);
```

N harfi çizimi birinci parça

```
glBegin(GL_POLYGON);  
for(int i = 192; i<208; i++)  
{  
    glVertex3f(x[i], y[i], z[i]);
```

```
}  
glEnd();
```

N harfi çizimi ikinci parça

```
glBegin(GL_POLYGON);  
for(int i = 208; i<224; i++)  
{  
    glVertex3f(x[i], y[i], z[i]);  
}  
glEnd();
```

N harfi çizimi üçüncü parça

```
glBegin(GL_POLYGON);  
for(int i = 224; i<240; i++)  
{  
    glVertex3f(x[i], y[i], z[i]);  
}  
glEnd();  
  
glPopMatrix(); // N harfi popMatrix yapıldı.  
glutSwapBuffers();  
}
```

Klavye tuş tanımlamaları

```
void keyboard(unsigned char key, int x, int y)  
{  
    static int wire = 0;  
    if (key == 27) {  
        exit(0);  

```

W tuşuna basıldığında tüm polygonları GL_FILL ile GL_LINE arasında döndürme işlemi tanımlanmıştır.

```
} else if (key == 'w' || key == 'W') {
```

```

    wire ^= 1; //tuşa her basmada değişken tersleniyor
    if (wire)
    {
        glPolygonMode(GL_FRONT, GL_LINE);
        glPolygonMode(GL_BACK, GL_LINE);
    }
    else
    {
        glPolygonMode(GL_FRONT, GL_FILL);
        glPolygonMode(GL_BACK, GL_FILL);
    }
}
glutPostRedisplay();
}

```

Mouse ile tıklanan noktanın x ve y değerlerini `_x` ve `_y` değişkenlerine yazmaktadır.

```

void mouse(int button, int state, int x, int y)
{
    old_x = x;
    old_y = y;
    glutPostRedisplay();
}

```

Mouse hareketini ilk okunan x ve y değerlerinden çıkartarak değişimi açı olarak `spin_x` ve `spin_y` değerlerine yazılmıştır. Bu değerler tüm harf çizimlerinde `glRotatef` fonksiyonu ile açı olarak tanımlandığından tüm harfler fark açısı kadar x ve y düzleminde yer değiştirmektedir.

```

void motion(int x, int y)
{
    spin_x = x - _x;
    spin_y = y - _y;
    glutPostRedisplay();
}

```



```
}
```

Main fonksiyonu

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

myReadFile adında dosya nesneleri oluşturulmuştur.

```
std::ifstream myReadFile;
```

C klasörünün altında olan name.dat dosyasından tüm koordinat değerleri okunmuştur.

```
myReadFile.open("C:/name.dat");
```

```
if (myReadFile.is_open()) {
```

240 değer x, y, z değişkenlerine yazılmıştır.

```
for (int i=0; i<241; i++){
    myReadFile >> x[i] >> y[i] >> z[i];
}
}
```

```
myReadFile.close();
```

```
int i, k;
```

```
glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB |
GLUT_DEPTH);
```

```
glutInitWindowPosition(50, 50);
```

```
glutInitWindowSize(600, 800);
```

```
glutInit(&argc, argv);
```

```
glutCreateWindow("soru1");
```

```
glutDisplayFunc(display);
```

```
glutReshapeFunc(reshape);
```

```
glutKeyboardFunc(keyboard);
```

```
glutMotionFunc(motion);
```

```
    glutMouseFunc(mouse);  
    glutMainLoop();  
    return 0;  
}
```

Ek

Finalsoru4.cpp

Finalsoru4.exe

Ekran çıktısı



Açı verilmiş ekran çıktısı



Açı verilmiş GL_LINE şeklinde (w tuşu ile) olan ekran çıktısı

