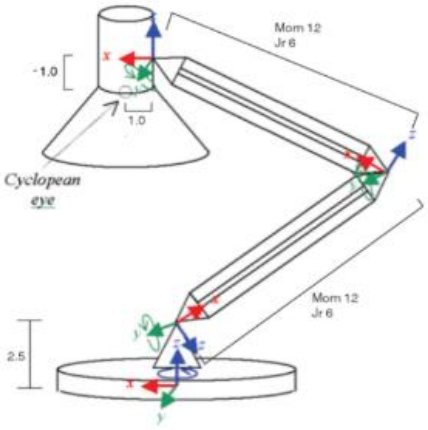
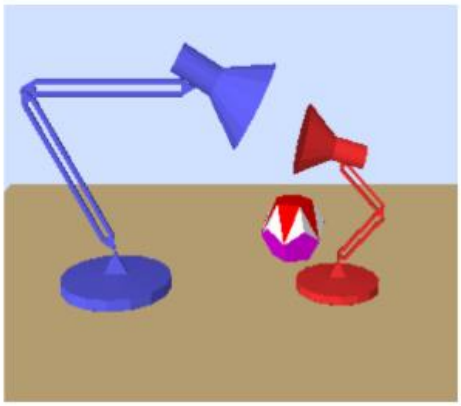
22.01.2020

Çağın Ağırdemir

SORU:

SORU 3) Pixar 'ın Luxor 'ını bir mesh dosyasından okuyup çizdiren ve kullanıcının luxor 'ı eklem yerlerinden klavye ile oynatabildiği bir cpp uygulaması yazınız. Burada luxor'ın parçaları ayrı birer mesh dosyası şeklinde olmalıdır.





CEVAP:

Pixar Luxor mesh dosyası bulunamadığı için glutWireCube ve GL_LINE_STRIP fonksiyonları ile benzeri çizilerek 3 parça ile 3 bağlantı noktasında 3 eksenli kontrol sağlanabilen bir program yazılmıştır.

Elbow parçası hareket tuşları (küçük harf olmalıdır)

z ekseninde -5 için '**z**' tuşu; +5 için '**x**' tuşu

x ekseninde -5 için '**c**' tuşu; +5 için '**v**' tuşu

y ekseninde -5 için '**b**' tuşu; +5 için '**n**' tuşu

Shoulder parçası hareket tuşları (küçük harf olmalıdır)

z ekseninde -5 için '**a**' tuşu; +5 için '**s**' tuşu

x ekseninde -5 için '**d**' tuşu; +5 için '**f**' tuşu

y ekseninde -5 için '**g**' tuşu; +5 için '**h**' tuşu

Head parçası hareket tuşları (küçük harf olmalıdır)

z ekseninde -5 için '**q**' tuşu; +5 için '**w**' tuşu

x ekseninde -5 için '**e**' tuşu; +5 için '**r**' tuşu

y ekseninde -5 için '**t**' tuşu; +5 için '**y**' tuşu

Tanımlamalar

```
#include <GL/glut.h>

static int shoulder_z,shoulder_y,shoulder_x = 0,
elbow_z,elbow_y,elbow_x = 0, head_z,head_y,head_x = 0;

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

wireBox fonksiyonu

```
void wireBox(GLdouble width, GLdouble height, GLdouble depth)
{
  glPushMatrix();
  glScalef(width, height, depth);
  glutWireCube(1.0);
  glPopMatrix();
}
```

Display fonksiyonu

```
void display() {
  glClear(GL_COLOR_BUFFER_BIT);
  glMatrixMode(GL_MODELVIEW);
  glPushMatrix();
```

Mouse kontrol için glRotatef fonksiyonu ile döndürme açıları spin_x ve spin_y değişkeni olarak ayarlanmıştır.

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

Shoulder kontrol ekleminin glRotatef ile x y z eksenlerinde döndürülmesi tanımlanıyor. Açı değerleri eksene göre shoulder_z, shoulder_y, shoulder_x değişkenleri olarak ayarlanmıştır.

```
  glRotatef((GLfloat)shoulder_z, 0.0, 0.0, 1.0);
  glRotatef((GLfloat)shoulder_y, 0.0, 1.0, 0.0);
  glRotatef((GLfloat)shoulder_x, 1.0, 0.0, 0.0);
```

Shoulder parçası çizilmiştir.

```
  wireBox(2.0, 1, 1.0);
```

Elbow parçasının orta noktası shoulder parçasının uç kısmına ayarlanıyor.

```
glTranslatef(1.0, 0.0, 0.0);
```

Elbow kontrol ekleminin glRotatef ile x y z eksenlerinde döndürülmesi tanımlanıyor. Parçaların birlikte hareket etmesini sağlamak için döndürme işlemi tanımlanmadan önce orta nokta diğer parçanın uçuna göre ayarlanmıştır. Çizilmeden öncede yine yeri glTranslatef fonksiyonu ile diğer parçanın uç kısmına alınmıştır. Açı değerleri eksene göre elbow _z, elbow _y, elbow _x değişkenleri olarak ayarlanmıştır.

```
  glRotatef((GLfloat)elbow_z, 0.0, 0.0, 1.0);
  glRotatef((GLfloat)elbow_y, 0.0, 1.0, 0.0);
  glRotatef((GLfloat)elbow_x, 1.0, 0.0, 0.0);
```

Elbow parçası çizilmeden önce yeri Shoulder parçasının ucuna ekleniyor. PopMatrix yapılmadığından diğer parçanın uç noktasına denk gelmektedir.

```
  glTranslatef(1.0, 0.0, 0.0);
```

Elbow parçası çizilmiştir.

```
  wireBox(2.0, 1, 1.0);
```

Head parçasının tamamı diğer parçanın ucuna getiriliyor.

```
  glTranslatef(1.0, 0.0, 0.0);
```

Head kontrol ekleminin glRotatef ile x y z eksenlerinde döndürülmesi tanımlanıyor. Açı değerleri eksene göre head _z, head _y, head _x değişkenleri olarak ayarlanmıştır.

```
  glRotatef((GLfloat)head_z, 0.0, 0.0, 1.0);
  glRotatef((GLfloat)head_y, 0.0, 1.0, 0.0);
  glRotatef((GLfloat)head_x, 1.0, 0.0, 0.0);
```

Head parçası çizilmiştir.

```
glBegin(GL_LINE_STRIP);
glVertex3f(.6,.9,.9);
glVertex3f(0,.5,.5);
glVertex3f(0,.5,-.5);
glVertex3f(.6,.9,-.9);
glVertex3f(.6,.9,.9);
glEnd();


glBegin(GL_LINE_STRIP);
```

```
glVertex3f(.6,.9,.9);
glVertex3f(0,.5,.5);
glVertex3f(0,-.5,.5);
glVertex3f(.6,-.9,.9);
glVertex3f(.6,.9,.9);
glEnd();

glBegin(GL_LINE_STRIP);
glVertex3f(.6,-.9,.9);
glVertex3f(0,-.5,.5);
glVertex3f(0,-.5,-.5);
glVertex3f(.6,-.9,-.9);
glVertex3f(.6,-.9,.9);
glEnd();

glBegin(GL_LINE_STRIP);
glVertex3f(.6,.9,-.9);
glVertex3f(0,.5,-.5);
glVertex3f(0,-.5,-.5);
glVertex3f(.6,-.9,-.9);
glVertex3f(.6,.9,-.9);
glEnd();
```

Head parçasının tamamı diğer parçanın ucuna gelecek şekilde ayarlanıyor.

```
  glTranslatef(1.0, 0.0, 0.0);
  glPopMatrix();
  glFlush();
}
```

reshape fonksiyonu

```
void reshape(GLint w, GLint h) {
  glViewport(0, 0, w, h);
  glMatrixMode(GL_PROJECTION);
```

```
  glLoadIdentity();
  gluPerspective(65.0, GLfloat(w)/GLfloat(h), 1.0, 20.0);
}

void init() {
  glShadeModel(GL_FLAT);
  glMatrixMode(GL_MODELVIEW);
  glLoadIdentity();
  gluLookAt(1,2,8, 0,0,0, 0,1,0);
}
```

Mouse ilk konumu spin_x, spin_y değişkenlerine yazılmıştır.

```
void mouse(int button, int state, int x, int y)
{
    spin_x = x;
    spin_y = y;
    glutPostRedisplay();
}
```

Klavye fonksiyonları

```
void keyboard(unsigned char key, int x, int y)
{
```

//shoulder kontrol seti

```
switch(key) {
    case 'a' : (shoulder_z -= 5) %= 360; glutPostRedisplay();
break;
    case 's' : (shoulder_z += 5) %= 360; glutPostRedisplay();
break;
    case 'd' : (shoulder_x -= 5) %= 360; glutPostRedisplay();
break;
    case 'f' : (shoulder_x += 5) %= 360; glutPostRedisplay();
break;
    case 'g' : (shoulder_y -= 5) %= 360; glutPostRedisplay();
break;
```

```
    case 'h' : (shoulder_y += 5) %= 360; glutPostRedisplay();
break;
}
```

//elbow kontrol seti

```
switch(key) {
    case 'z' : (elbow_z -= 5) %= 360; glutPostRedisplay();
break;
    case 'x' : (elbow_z += 5) %= 360; glutPostRedisplay();
break;
    case 'c' : (elbow_x -= 5) %= 360; glutPostRedisplay();
break;
    case 'v' : (elbow_x += 5) %= 360; glutPostRedisplay();
break;
    case 'b' : (elbow_y -= 5) %= 360; glutPostRedisplay();
break;
    case 'n' : (elbow_y += 5) %= 360; glutPostRedisplay();
break;
}
```

//head kontrol seti

```
switch(key) {
    case 'q' : (head_z -= 5) %= 360; glutPostRedisplay();
break;
    case 'w' : (head_z += 5) %= 360; glutPostRedisplay();
break;
    case 'e' : (head_x -= 5) %= 360; glutPostRedisplay();
break;
    case 'r' : (head_x += 5) %= 360; glutPostRedisplay();
break;
    case 't' : (head_y -= 5) %= 360; glutPostRedisplay();
break;
    case 'y' : (head_y += 5) %= 360; glutPostRedisplay();
break;
}
```

```
}
```

Mouse hareketinde ilk konuma göre olan farklar spin_x, spin_y değişkenlerine yazılmıştır.

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

Main fonksiyonu

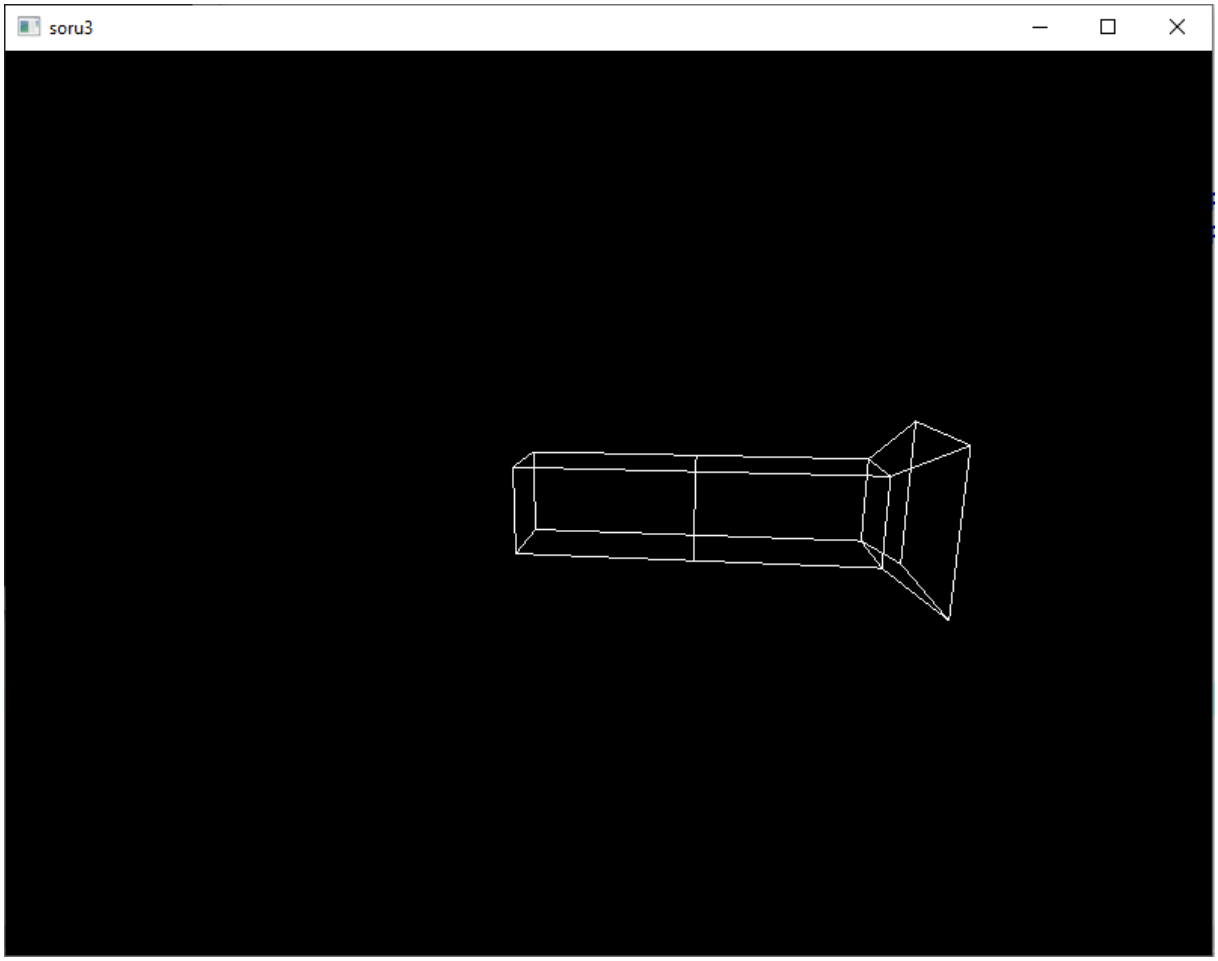
```
int main(int argc, char** argv) {
  glutInit(&argc, argv);
  glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
  glutInitWindowPosition(80, 80);
  glutInitWindowSize(800, 600);
  glutCreateWindow("soru3");
  glutDisplayFunc(display);
  glutReshapeFunc(reshape);
    glutKeyboardFunc(keyboard);
      glutMotionFunc(motion);
    glutMouseFunc(mouse);
  init();
  glutMainLoop();
}
```

Ek

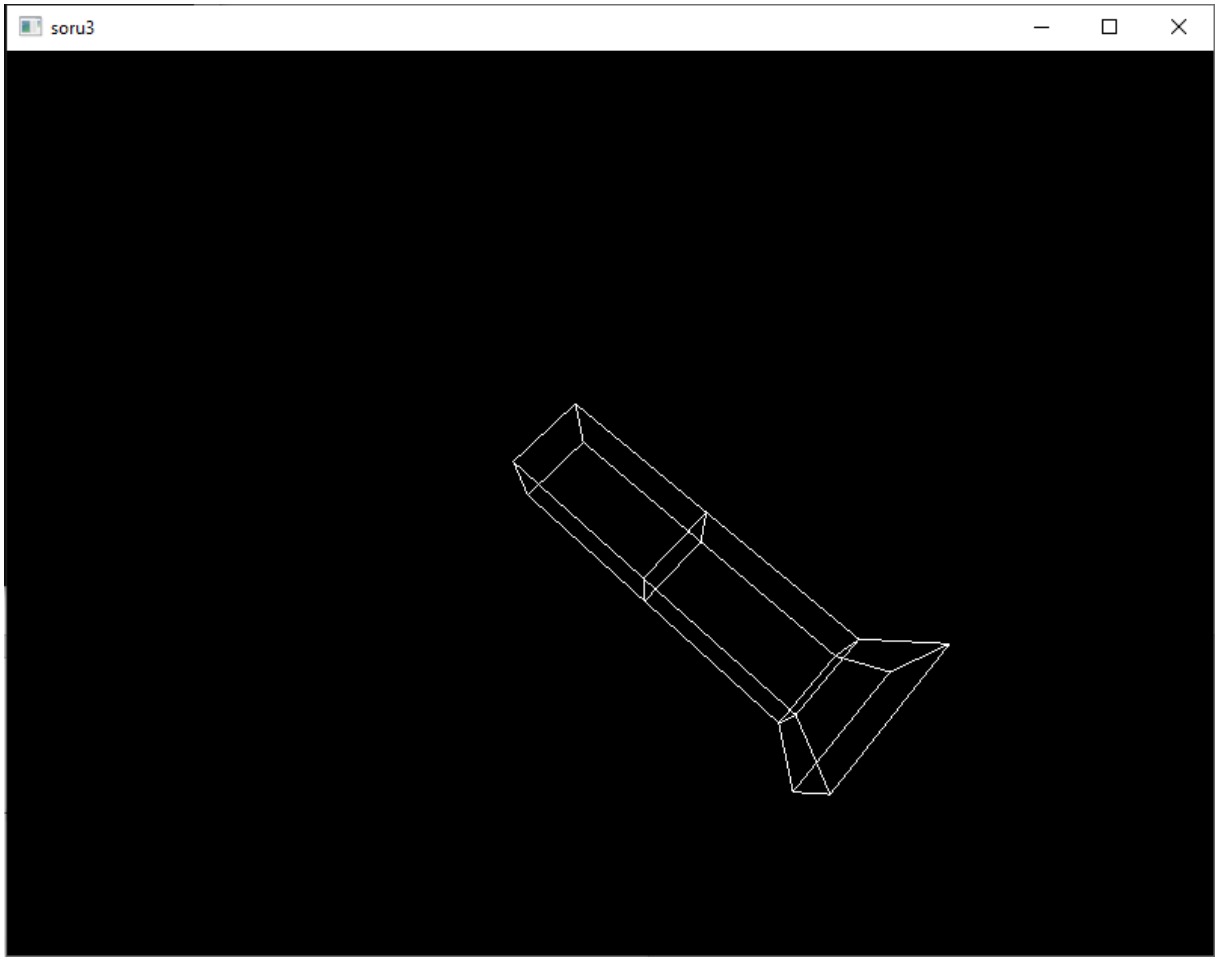
Finalsoru3.cpp

Finalsoru3.exe

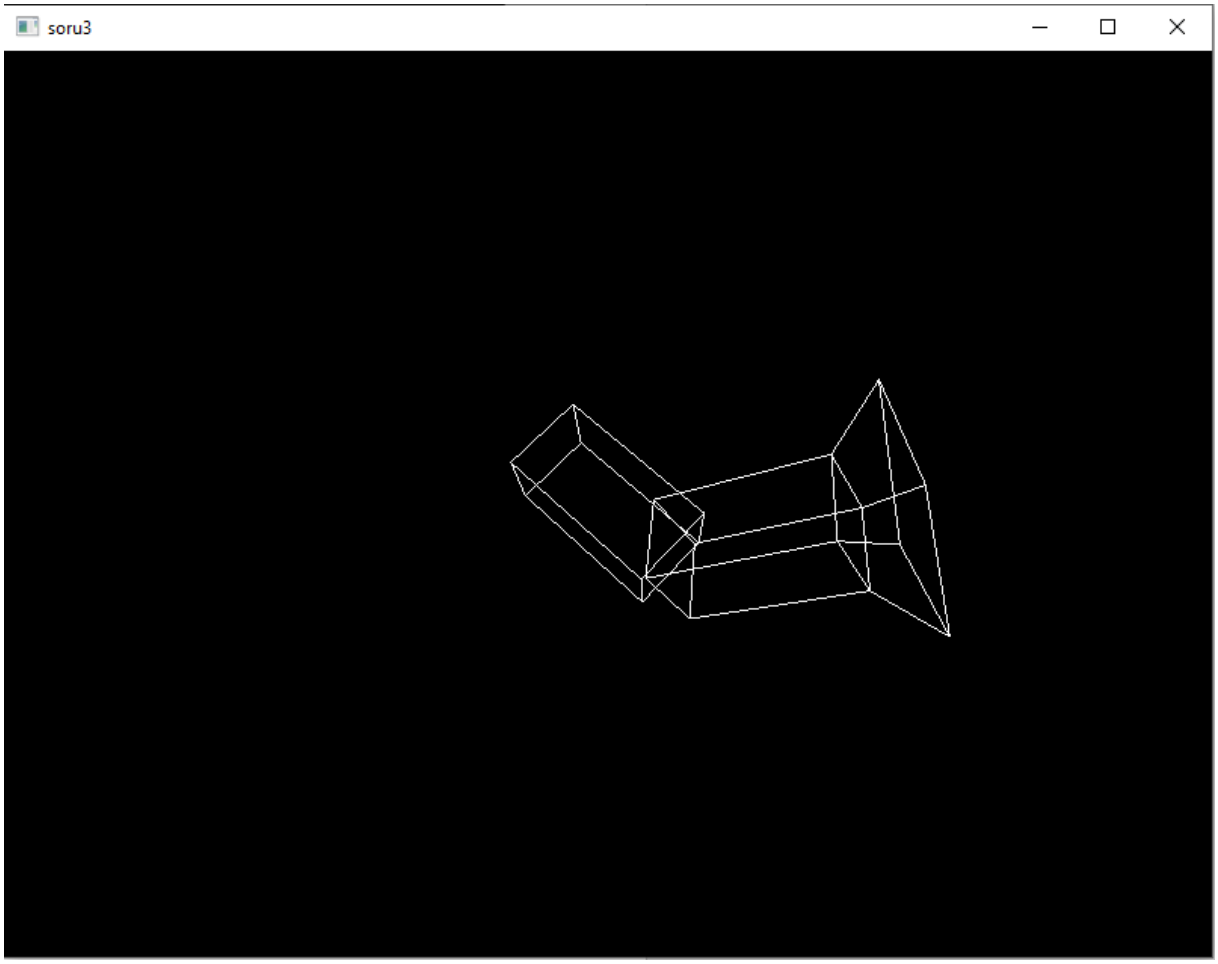
Ekran Çıktısı



Shoulder x,y,z kontrolü (tüm nesneyi hareket ettiriyor) ekran çıktısı



Elbow x,y,z kontrolü (head ile elbow sadece hareket ediyor) ekran çıktısı



Head kontrolü x,y,z de (sadece head hareket ediyor) ekran çıktısı