

GTU Department of Computer Engineering

CSE 461 – Spring 2020

HW1 Documentation

MUHAMMETALP ERDEM

151044006

Deadline: 17 May 2021

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

```
217 int main(int argc, char** argv) {  
218     glutInit(&argc, argv);  
219     glutInitWindowSize(1280, 720);  
220     glutCreateWindow("CSE 461 HW1");  
221     glEnable(GL_DEPTH_TEST);  
222     glutDisplayFunc(render);  
223     glutKeyboardFunc(keyboard_handler);  
224     glutReshapeFunc(reshape_handler);  
225     glutTimerFunc(5, frame_updater, 0);  
226     glutMainLoop();  
227     return 0;  
228 }
```

glutInit : Glut kütüphanesinin başlatıcı fonksiyonudur.

glutInitWindowSize : Açılacak pencerenin büyüklüğü 1280 * 720 olarak ayarlanmıştır.

glutCreateWindow : "CSE 461 HW1" Adlı pencere açılmıştır.

glEnable : Görüntünün sağlıklı çalışabilmesi için aktif hale getirilmiştir, sahnedeki parçalara derinlik testi uygulanır.

glutDisplayFunc : Glut kütüphanesi görüntü fonksiyonuna tarafımdan yazılan fonksiyon atanmıştır.

glutKeyboardFunc : Glut kütüphanesi klavye callback fonksiyonuna tarafımdan yazılan fonksiyon atanmıştır.

glutReshapeFunc : Glut kütüphanesi Reshape fonksiyonuna tarafımdan yazılan fonksiyon atanmıştır.

glutTimerFunc : Frame yenilenmesi için tekrar çağrılan bir fonksiyon gerekmektedir, bu amaç için glutTimerFunc kullanılmıştır.

glutMainLoop : Glut kütüphanesi ana döngüsü çağrılmıştır.

2 void frame_updater(int value)

```
11 void frame_updater(int value) {  
12     glutPostRedisplay();  
13     glutTimerFunc(10, frame_updater, 0);  
14 }
```

glutPostRedisplay : Görüntünün yenilenmesi emri verilmiştir.

glutTimerFunc : Gelecek frame'in oluşturulması için fonksiyon gecikmeli olarak çağrılmıştır.

3 void keyboard_handler(unsigned char key, int x, int y)

```
21 void keyboard_handler(unsigned char key, int x, int y) {
22     switch (key) {
23         case 87:
24             current_position_y += 0.01f;
25             break;
26         case 119:
27             current_position_y -= 0.01f;
28             break;
29         case 83:
30             current_position_x += 0.01f;
31             break;
32         case 115:
33             current_position_x -= 0.01f;
34             break;
35         case 68:
36             current_position_x += 0.01f;
37             break;
38         case 100:
39             current_position_x -= 0.01f;
40             break;
41         case 65:
42             current_position_x += 0.01f;
43             break;
44         case 97:
45             current_position_x -= 0.01f;
46             break;
```

Switch-Case yapısı kullanılarak klavye tuşları yakalanmıştır.

Key - W: Sahne yukarı hareket eder.

Key - S: Sahne aşağı hareket eder.

Key - A: Sahne sola hareket eder.

Key - D: Sahne sağa hareket eder.

Key - Q: Sahne sola döner.

Key - E: Sahne sağa döner.

Key - Z: Sahne büyür.

Key - X: Sahne küçülür.

4 Set fonksiyonları

```
74 void setTranslate(float x, float y, float z) {  
75     glTranslatef(x, y, z);  
76 }  
77   
78 void setScale(float x) {  
79     glScalef(x, x, x);  
80 }  
81 void setRotate(float x) {  
82     glRotatef(x, 0.0f, 1.0f, 0.0f);  
83 }  
84
```

Her yenilenen frame’de klavye girdisi ile değişen global rotasyon, büyüklük ve pozisyon değişkenlerini sahneye uygulamak ile sorumlulardır.

5 void prepare render()

```
85 void prepare_render() {  
86     glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);  
87     glMatrixMode(GL_MODELVIEW);  
88     glLoadIdentity();  
89     setRotate(0.0f);  
90     setTranslate(0.0f, 0.0f, -2.0f);  
91     setScale(1.0f);  
92     glPopMatrix();  
93     glPushMatrix();  
94     setRotate(current_rotation);  
95     setTranslate(current_position_x, current_position_y, current_position_z);  
96     setScale(current_scale);  
97 }  
98
```

Bu fonksiyon gösterimden önceki hazırlıklardan sorumludur.

glClear : Derinlik ve renk bufferlarını temizler.

glMatrixMode : Verilen modu çalışmak üzere matris yığınınına uygular. Bu sayede kamera üzerinde işlem yapılabilir (GL MODELVIEW).

Set fonksiyonları kameranın özelliklerini atar.

glLoadIdentity : Şuanki matris yerine identity matrisini koyar.

glPopMatrix(); : Matris yığının ucundaki matrisi dışarı çıkarır.

glPushMatrix : Matris yığının ucundaki matrisi kopyalayıp bir alta yerleştirir. Bu sayede sahne üzerinde işlem yapılabilir.

Set fonksiyonları sahnenin özelliklerini atar.

6 void draw house()

```
99 void draw_house() {
100
101     glClear(GL_COLOR_BUFFER_BIT);
102
103     // On duvar
104     glColor3f(0.917, 0.686, 0.482);
105     glBegin(GL_POLYGON);
106     glVertex3f(-0.4, 0.2, 0.0);
107     glVertex3f(0.4, 0.2, 0.0);
108     glVertex3f(0.4, -0.2, 0.0);
109     glVertex3f(-0.4, -0.2, 0.0);
110     glEnd();
111
112     // Arka duvar
113     glColor3f(0.917, 0.686, 0.482);
114     glBegin(GL_POLYGON);
115     glVertex3f(-0.4, 0.2, -0.4);
116     glVertex3f(0.4, 0.2, -0.4);
117     glVertex3f(0.4, -0.2, -0.4);
118     glVertex3f(-0.4, -0.2, -0.4);
119     glEnd();
120
121     // Sag duvar
122     glColor3f(0.917, 0.686, 0.482);
123     glBegin(GL_POLYGON);
124     glVertex3f(0.4, 0.2, -0.4);
125     glVertex3f(0.4, 0.2, 0.0);
126     glVertex3f(0.4, -0.2, 0.0);
127     glVertex3f(0.4, -0.2, -0.4);
128 }
```

Burası sahnedeki objeleri oluşturulduğu fonksiyondur.

glClear : Renk bufferını temizler.

glBegin: Çizimin hangi obje (poligon, üçgen...) olarak yapılacağı belirlenir.

glVertex3f: 3 boyutlu uzayda bir köşe oluşturulur.

glEnd: Eğer oluşturulan köşe sayısı seçilen obje ile uyuyorsa bu köşeler temel alınarak çizim yapılır. Bu aşamada köşelerin sırası önemlidir.

7 Çıktılar



