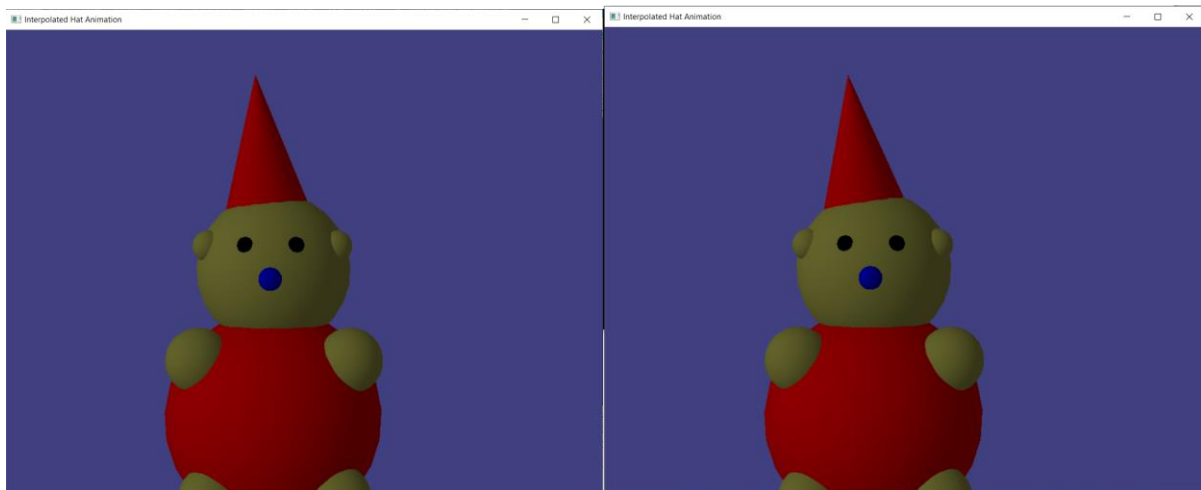


Лаборатори №8

GLU сангийн quadric гадаргууг ашиглан ямар нэг биетийн 3-хэмжээст загварыг байгуулж, хөдөлгөөнд оруулах

Дараах загвартай 3D модел байгуулсан бөгөөд үүнийг малгайг нэг тал руугаа унаж байгаа мэтээр tweening арга ашиглаад хөдөлгөөн оруулсан.

```
1  #include "GL/freeglut.h"
2
3  static GLfloat xRot = 0.0f;
4  static GLfloat yRot = 0.0f;
5  static float t = 0.0f;
6
7  void ChangeSize(int w, int h) {
8      GLfloat fAspect;
9
10     if (h == 0) h = 1;
11
12     glViewport(0, 0, w, h);
13
14     fAspect = (GLfloat)w / (GLfloat)h;
15
16     glMatrixMode(GL_PROJECTION);
17     glLoadIdentity();
18     gluPerspective(35.0f, fAspect, 1.0, 40.0);
19     glMatrixMode(GL_MODELVIEW);
20     glLoadIdentity();
21 }
22
23 void SpecialKeys(int key, int x, int y) {
24     if (key == GLUT_KEY_UP) xRot -= 5.0f;
25     if (key == GLUT_KEY_DOWN) xRot += 5.0f;
26     if (key == GLUT_KEY_LEFT) yRot -= 5.0f;
27     if (key == GLUT_KEY_RIGHT) yRot += 5.0f;
28
29     xRot = (GLfloat)((const int)xRot % 360);
30     yRot = (GLfloat)((const int)yRot % 360);
31     glutPostRedisplay();
32 }
33
34 void interpolatePosition(float t, float Ax, float Ay, float Az, float Bx, float By, float Bz, float &Px, float &Py, float &Pz) {
35     Px = (1 - t) * Ax + t * Bx;
36     Py = (1 - t) * Ay + t * By;
37     Pz = (1 - t) * Az + t * Bz;
38 }
```



```

40 void RenderScene(void) {
41     glClearColor(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
42     GLUQuadricObj *pObj = gluNewQuadric();
43     gluQuadricNormals(pObj, GLU_SMOOTH);
44
45     glMatrixMode(GL_MODELVIEW);
46     glLoadIdentity();
47     glPushMatrix();
48     glTranslatef(0.0, -0.2, -5.0);
49     glRotatef(yRot, 0.0, 1.0, 0.0);
50     glRotatef(xRot, 1.0, 0.0, 0.0);
51
52     // Variables for hat positions
53     float Ax = -0.2, Ay = 0.5, Az = 0.2; // "haganan malgai"
54     float Bx = -0.3, By = 0.4, Bz = 0.2; // "kagah malgai"
55     float hatX, hatY, hatZ;
56
57     interpolatePosition(t, Ax, Ay, Az, Bx, By, Bz, hatX, hatY, hatZ);
58
59     // "malgai"
60     glPushMatrix();
61     glTranslatef(hatX, hatY, hatZ);
62     glRotatef((1 - t) * -75 + t * -90, 1.0, t, t);
63     glColor3f(1, 0.0, 0.0);
64     gluCylinder(pObj, 0.3, 0.0, 0.93, 30, 30);
65     glPopMatrix();
66
67     // "kaganan sphere talgai"
68     glPushMatrix();
69     glTranslatef(-0.2, 0.15, 0.2);
70     glColor3f(0.7, 0.7, 0.3);
71     gluSphere(pObj, 0.5, 25, 25);
72     glPopMatrix();
73
74     // "hagan"
75     glPushMatrix();
76     glTranslatef(-0.2, -0.8, 0.2);
77     glColor3f(1.0, 0.0, 0.0);
78     gluSphere(pObj, 0.7, 25, 25);
79     glPopMatrix();
80
81     // "hamax"
82     glPushMatrix();
83     glTranslatef(-0.2, 0.1, 0.7);
84     glColor3f(0.0, 0.0, 1.0);
85     gluSphere(pObj, 0.07, 15, 15);
86     glPopMatrix();
87
88     // "aud"
89     glPushMatrix();
90     glTranslatef(-0.05, 0.3, 0.64);
91     glColor3f(0.0, 0.0, 0.0);
92     gluSphere(pObj, 0.05, 15, 15);
93     glPopMatrix();
94     glPushMatrix();
95     glTranslatef(-0.35, 0.3, 0.64);
96     glColor3f(0.0, 0.0, 0.0);
97     gluSphere(pObj, 0.05, 15, 15);
98     glPopMatrix();
99
100    // "chih"
101    glPushMatrix();
102    glTranslatef(0.2, 0.3, 0.4);
103    glColor3f(0.7, 0.7, 0.3);
104    gluSphere(pObj, 0.1, 15, 15);
105    glPopMatrix();
106    glPushMatrix();
107    glTranslatef(-0.6, 0.3, 0.4);
108    glColor3f(0.7, 0.7, 0.3);
109    gluSphere(pObj, 0.1, 15, 15);
110    glPopMatrix();
111
112    // "hag"
113    glPushMatrix();
114    glTranslatef(0.2, -1.2, 0.64);
115    glColor3f(0.7, 0.7, 0.3);
116    gluSphere(pObj, 0.2, 25, 25);
117    glPopMatrix();
118    glPushMatrix();

```

```

118     glPushMatrix();
119         glTranslatef(-0.55, -1.2, 0.64);
120         glColor3f(0.7, 0.7, 0.3);
121         gluSphere(pObj, 0.2, 25, 25);
122     glPopMatrix();
123
124     //gar
125     glPushMatrix();
126         glTranslatef(0.28, -0.4, 0.5);
127         glColor3f(0.7, 0.7, 0.3);
128         gluSphere(pObj, 0.2, 25, 25);
129     glPopMatrix();
130     glPushMatrix();
131         glTranslatef(-0.65, -0.4, 0.5);
132         glColor3f(0.7, 0.7, 0.3);
133         gluSphere(pObj, 0.2, 25, 25);
134     glPopMatrix();
135
136     //ax tall ai
137     glPushMatrix();
138         glTranslatef(-0.2, -1.2, -0.4);
139         glColor3f(0.7, 0.7, 0.3);
140         gluSphere(pObj, 0.2, 25, 25);
141     glPopMatrix();
142
143     glPopMatrix();
144     glutSwapBuffers();
145 }
146
147 void TimerFunction(int value) {
148     t += 0.01f;
149     if (t > 1.0f) t = 0.0f;
150     glutPostRedisplay();
151     glutTimerFunc(33, TimerFunction, 1);
152 }
153
154 void SetupRC() {
155     GLfloat whiteLight[] = {0.05f, 0.05f, 0.05f, 1.0f};
156     GLfloat sourceLight[] = {0.25f, 0.25f, 0.25f, 1.0f};
157     GLfloat lightPos[] = {-10.f, 5.0f, 5.0f, 1.0f};
158
159     glEnable(GL_DEPTH_TEST);
160     glFrontFace(GL_CCW);
161     glEnable(GL_CULL_FACE);
162     glEnable(GL_LIGHTING);
163
164     glLightModelfv(GL_LIGHT_MODEL_AMBIENT, whiteLight);
165     glLightfv(GL_LIGHT0, GL_AMBIENT, sourceLight);
166     glLightfv(GL_LIGHT0, GL_DIFFUSE, sourceLight);
167     glLightfv(GL_LIGHT0, GL_POSITION, lightPos);
168     glEnable(GL_LIGHT0);
169
170     glEnable(GL_COLOR_MATERIAL);
171     glColorMaterial(GL_FRONT, GL_AMBIENT_AND_DIFFUSE);
172
173     glClearColor(0.25f, 0.25f, 0.50f, 1.0f);
174 }
175
176 int main(int argc, char *argv[]) {
177     glutInit(&argc, argv);
178     glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
179     glutInitWindowSize(900, 700);
180     glutCreateWindow("Interpolated Hat Animation");
181     glutReshapeFunc(ChangeSize);
182     glutSpecialFunc(SpecialKeys);
183     glutDisplayFunc(RenderScene);
184     SetupRC();
185     glutTimerFunc(33, TimerFunction, 1);
186     glutMainLoop();
187     return 0;
188 }
189

```