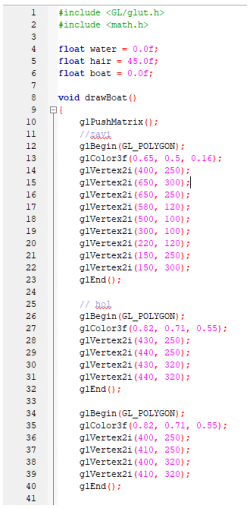
Компьютерын ухаан
21B1NUM1182
А. Гаусар

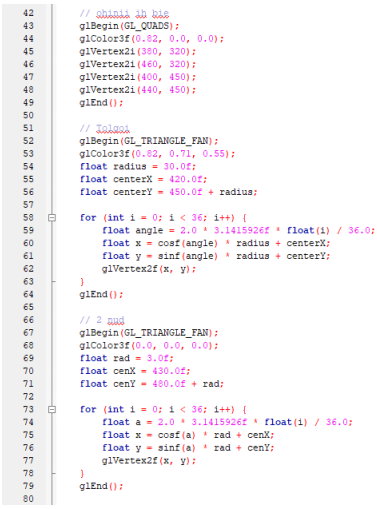
Компьютер графикийн үндэс

Лабораторий №4

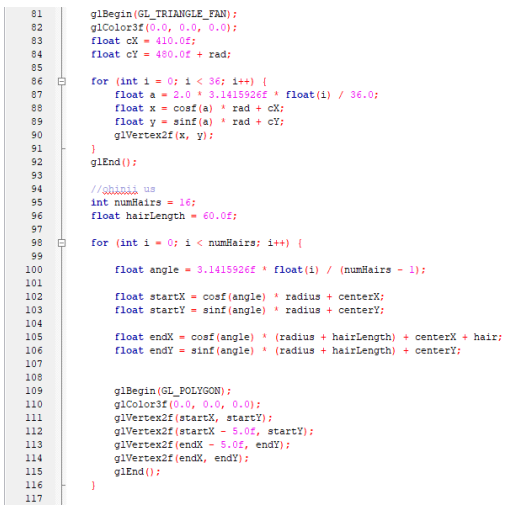
glutTimerFunc()-ийн хэрэглээ: Өгөгдсөн зургийн аль нэг хэсгийг сонгон авч tweening аргаар хөдөлгөх



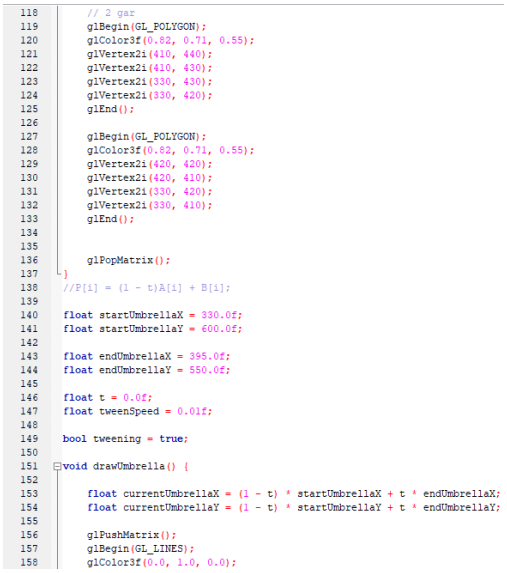
```
#include <GL/glut.h>
#include <math.h>

float water = 0.0f;
float hair = 45.0f;
float boat = 0.0f;

void drawBoat()
{
    glPushMatrix();
    //sayd
    glBegin(GL_POLYGON);
    glColor3f(0.65, 0.5, 0.16);
    glVertex2i(400, 250);
    glVertex2i(650, 300);
    glVertex2i(650, 250);
    glVertex2i(500, 120);
    glVertex2i(500, 100);
    glVertex2i(300, 100);
    glVertex2i(220, 120);
    glVertex2i(150, 250);
    glVertex2i(150, 300);
    glEnd();

    // hol
    glBegin(GL_POLYGON);
    glColor3f(0.82, 0.71, 0.55);
    glVertex2i(430, 250);
    glVertex2i(440, 250);
    glVertex2i(430, 320);
    glVertex2i(440, 320);
    glEnd();

    glBegin(GL_POLYGON);
    glColor3f(0.82, 0.71, 0.55);
    glVertex2i(400, 250);
    glVertex2i(410, 250);
    glVertex2i(400, 320);
    glVertex2i(410, 320);
    glEnd();
```



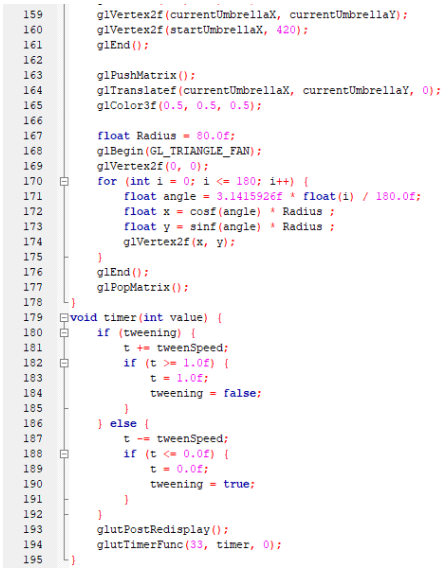
```
    // chihii ih hes
    glBegin(GL_QUADS);
    glColor3f(0.82, 0.0, 0.0);
    glVertex2i(380, 320);
    glVertex2i(460, 320);
    glVertex2i(400, 450);
    glVertex2i(440, 450);
    glEnd();

    // Toirog
    glBegin(GL_TRIANGLE_FAN);
    glColor3f(0.82, 0.71, 0.55);
    float radius = 30.0f;
    float centerX = 420.0f;
    float centerY = 450.0f + radius;

    for (int i = 0; i < 36; i++) {
        float angle = 2.0 * 3.1415926f * float(i) / 36.0;
        float x = cosf(angle) * radius + centerX;
        float y = sinf(angle) * radius + centerY;
        glVertex2f(x, y);
    }
    glEnd();

    // 2 nud
    glBegin(GL_TRIANGLE_FAN);
    glColor3f(0.0, 0.0, 0.0);
    float rad = 3.0f;
    float cenX = 430.0f;
    float cenY = 480.0f + rad;

    for (int i = 0; i < 36; i++) {
        float a = 2.0 * 3.1415926f * float(i) / 36.0;
        float x = cosf(a) * rad + cenX;
        float y = sinf(a) * rad + cenY;
        glVertex2f(x, y);
    }
    glEnd();
```



```
    glBegin(GL_TRIANGLE_FAN);
    glColor3f(0.0, 0.0, 0.0);
    float cX = 410.0f;
    float cY = 480.0f + rad;

    for (int i = 0; i < 36; i++) {
        float a = 2.0 * 3.1415926f * float(i) / 36.0;
        float x = cosf(a) * rad + cX;
        float y = sinf(a) * rad + cY;
        glVertex2f(x, y);
    }
    glEnd();

    //chihii us
    int numHairs = 16;
    float hairLength = 60.0f;

    for (int i = 0; i < numHairs; i++) {

        float angle = 3.1415926f * float(i) / (numHairs - 1);

        float startX = cosf(angle) * radius + centerX;
        float startY = sinf(angle) * radius + centerY;

        float endX = cosf(angle) * (radius + hairLength) + centerX + hair;
        float endY = sinf(angle) * (radius + hairLength) + centerY;


        glBegin(GL_POLYGON);
        glColor3f(0.0, 0.0, 0.0);
        glVertex2f(startX, startY);
        glVertex2f(startX - 5.0f, startY);
        glVertex2f(endX - 5.0f, endY);
        glVertex2f(endX, endY);
        glEnd();
    }
```



```
    // 2 gar
    glBegin(GL_POLYGON);
    glColor3f(0.82, 0.71, 0.55);
    glVertex2i(410, 440);
    glVertex2i(410, 430);
    glVertex2i(330, 430);
    glVertex2i(330, 420);
    glEnd();

    glBegin(GL_POLYGON);
    glColor3f(0.82, 0.71, 0.55);
    glVertex2i(420, 420);
    glVertex2i(420, 410);
    glVertex2i(330, 420);
    glVertex2i(330, 410);
    glEnd();


    glPopMatrix();
}
//P[i] = (1 - t)A[i] + B[i];

float startUmbrellaX = 330.0f;
float startUmbrellaY = 600.0f;

float endUmbrellaX = 395.0f;
float endUmbrellaY = 550.0f;

float t = 0.0f;
float tweenSpeed = 0.01f;

bool tweening = true;

void drawUmbrella() {

    float currentUmbrellaX = (1 - t) * startUmbrellaX + t * endUmbrellaX;
    float currentUmbrellaY = (1 - t) * startUmbrellaY + t * endUmbrellaY;

    glPushMatrix();
    glBegin(GL_LINES);
    glColor3f(0.0, 1.0, 0.0);
```



```
    glVertex2f(currentUmbrellaX, currentUmbrellaY);
    glVertex2f(startUmbrellaX, 420);
    glEnd();

    glPushMatrix();
    glTranslatef(currentUmbrellaX, currentUmbrellaY, 0);
    glColor3f(0.5, 0.5, 0.5);

    float Radius = 80.0f;
    glBegin(GL_TRIANGLE_FAN);
    glVertex2f(0, 0);
    for (int i = 0; i <= 180; i++) {
        float angle = 3.1415926f * float(i) / 180.0f;
        float x = cosf(angle) * Radius ;
        float y = sinf(angle) * Radius ;
        glVertex2f(x, y);
    }
    glEnd();
    glPopMatrix();
}
void timer(int value) {
    if (tweening) {
        t += tweenSpeed;
        if (t >= 1.0f) {
            t = 1.0f;
            tweening = false;
        }
    } else {
        t -= tweenSpeed;
        if (t <= 0.0f) {
            t = 0.0f;
            tweening = true;
        }
    }
    glutPostRedisplay();
    glutTimerFunc(33, timer, 0);
}
```

```
196 void drawWater() {
197     glColor3f(0.0, 0.4, 1.0);
198
199     glBegin(GL_POLYGON);
200
201     float waveHeight = 15.0f;
202     float waveFrequency = 0.05f;
203     float xStart = 0.0f;
204     float xEnd = 1200.0f;
205     float waterLevel = 100.0f;
206
207     for (float x = xStart; x <= xEnd; x += 5.0f) {
208         float y = waterLevel + sinf((x + water) * waveFrequency) * waveHeight;
209         glVertex2f(x, y);
210     }
211
212     glVertex2f(xEnd, 0.0f);
213     glVertex2f(xStart, 0.0f);
214
215     glEnd();
216 }
217 void display() {
218     glClearColor(1.0f, 1.0f, 1.0f, 1.0f);
219     glClear(GL_COLOR_BUFFER_BIT);
220     drawUmbrella();
221     drawBoat();
222     drawWater();
223     glutSwapBuffers();
224 }
225
226 void reshape(GLsizei w, GLsizei h) {
227     glViewport(0, 0, w, h);
228     glMatrixMode(GL_PROJECTION);
229     glLoadIdentity();
230     gluOrtho2D(0.0, 1200.0, 0.0, 800.0);
231     glMatrixMode(GL_MODELVIEW);
232     glLoadIdentity();
233 }
```

```
235 int main(int argc, char** argv) {
236     glutInit(&argc, argv);
237     glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
238     glutInitWindowSize(1200, 600);
239     glutCreateWindow("Girl on the boat");
240     glutDisplayFunc(display);
241     glutReshapeFunc(reshape);
242
243     glutTimerFunc(0, timer, 0);
244
245     glutMainLoop();
246     return 0;
247 }
248
```

Дүгнэлт:

drawBoat() функцээр завь болон охины зургийг зурсан, drawUmbrella() функцээр Tweening аргыг хэрэгжүүлсэн ба энд шүхрийн байрлал өөрчлөгдөж байгаа, drawWater() функц нь усны гадаргууг дүрсэлсэн. Хэрэгжүүлсэн tweening аргаа glutTimerFunc() ашиглан нэгж хугацаанд 33 фрэйм солигддог байхаар хийлээ.

Үр дүн: