

# ШИНЖЛЭХ УХААН ТЕХНОЛОГИЙН ИХ СУРГУУЛЬ

Мэдээлэл, холбооны технологийн сургууль



## БИЕ ДААЛТЫН АЖЛЫН ТАЙЛАН 1

Компьютерийн график (F.CS209)

2022-2023 хичээлийн жилийн намрын улирал

Шалгасан багш:

маг. Х.Хулан

Гүйцэтгэсэн:

Багийн дугаар: 2

Багийн гишүүд:

B210910019 А. Эрхэмбаяр  
B210910068 Б.Балгансүрэн  
B210910057 Г.Пүрэвбаатар  
B210910023 Н.Төрбадрах

Лаб.цаг:2-2

Улаанбаатар хот

2022 он

```

#include <iostream>
#include <GL\freeglut.h>
using namespace std;
// функцийн prototype
void MyInit();
void Display();
void Rotate();
void Reshape(int w, int h);
void DividePyramid(GLfloat* a, GLfloat* b, GLfloat* c, GLfloat* d, GLfloat* s, int level);
void DrawPyramid(GLfloat* a, GLfloat* b, GLfloat* c, GLfloat* d, GLfloat* s);
void DrawTriangle(GLfloat* a, GLfloat* b, GLfloat* c);
void DrawQuoad(GLfloat* a, GLfloat* b, GLfloat* c, GLfloat* d);
void DrawLineTriangle(GLfloat* a, GLfloat* b, GLfloat* c);
void DrawLineQuoad(GLfloat* a, GLfloat* b, GLfloat* c, GLfloat* d);
void Keyboard(unsigned char key, int x, int y);
GLfloat angle = 45;
bool rot = true;
int level = 1; //хуваалтын зэрэг
bool state = false; //өнгөтэй үгүйг заах хувьсагч
GLfloat pyramid[5][3] = // матриц зарласан
{
{ -1.0, -1.0, 1.0 },
{ 1.0, -1.0, 1.0 },
{ 1.0, -1.0, -1.0 },
{ -1.0, -1.0, -1.0 },
{ 0.0, 1.0, 0.0 }
};
int main(int argc, char** argv)
{
    //initization
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH); // дэлгэцэнд RGB өнгө,
    DOUBLE төрөл ашиглуулна.
    glutInitWindowSize(700, 700); // цонхны хэмжээ тохируулна.
    glutCreateWindow("Serpinski Pyramid Zurah");
    MyInit();
    glutDisplayFunc(Display); // display функцийг байгуулах функцыг дуудах
    glutReshapeFunc(Reshape); // цонхыг хаахгүйгээр дахин зурах
    glutKeyboardFunc(Keyboard); // keyboard eventlist-ажиллуулах функц дуудах
    glutIdleFunc(Rotate); //
    glutMainLoop(); // утга оруултал терминалыг хаахгүй байх
    return 0;
}
void MyInit()
{
    glClearColor(0.0, 0.0, 0.0, 1.0); // өнгө тодорхойлох
    glEnable(GL_DEPTH_TEST); //гүний буффер эхлүүлэх
    glHint(GL_PERSPECTIVE_CORRECTION_HINT, GL_NICEST); // *glHint – хэрэгжилтэд зориулсан
    зөвлөмжийг зааж өгдөг ба
    GL_PERSPECTIVE_CORRECTION_HINT-
    Өнгөний чанар, координатын бүтэц, манангийн координатын интерполяцийн чанарыг заана.
    GL_NICEST-Хамгийн зөв буюу хамгийн чанартай сонголтыг сонгох хэрэгтэй
    */
}
void Display() //пирамидыг байгуулах функц
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT); // өнгөний буффер цэвэрлэж
    ашиглахад бэлэн болгох
    glMatrixMode(GL_MODELVIEW); //Дараах матрицын үйлдлүүдийг загвар харах матрицын стект
    хэрэглэнэ.
    glLoadIdentity(); //одоогийн матрицыг өөрчлөх
    glTranslatef(0.0, 0.0, -4.0); // Z тэнхлэгийн дагуу -4 өөр зөөх
    glRotatef(angle, 0, 1, 0); // angle=45 өнцгөөр эргүүлэх
    DividePyramid(pyramid[0], pyramid[1], pyramid[2], pyramid[3], pyramid[4],
    level); //пирамидыг хувааж байгуулах
    glutPostRedisplay(); //одоогийн цонхыг дахин харуулах
    glutSwapBuffers(); //давхар буфертэй бол одоогийн цонхны буферыг солино.
}
void Reshape(int w, int h)
{
    glViewport(0, 0, w, h); //функц нь харах хэсгийг тохируулдаг.
    glMatrixMode(GL_PROJECTION); //аль матриц нь ерөнхий матриц болохыг зааж өгнө

```

```

    glLoadIdentity();
    gluPerspective(45.0, (GLdouble)w / (GLdouble)h, 0.1, 200.0); //харагдацыг өөрчлөх
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}
void DividePyramid(GLfloat* a, GLfloat* b, GLfloat* c, GLfloat* d, GLfloat* s, int level)
//пирамидын 5н орой байгаа. level-хуваалтын зэрэг
{
    GLfloat ab[3], bc[3], ac[3], ad[3], cd[3], bs[3], as[3], cs[3], ds[3];
    if (level == 0)
    {
        DrawPyramid(a, b, c, d, s); //level=0 буюу ерөөсөө хуваагаагүй байгаа үед
бүгд ижил
    }
    else // хэрэв үгүй бол
    {
        for (int j = 0; j < 3; j++)
        {
            ab[j] = (a[j] + b[j]) / 2.0; // голын цэгийг олох
            bc[j] = (b[j] + c[j]) / 2.0;
            ac[j] = (a[j] + c[j]) / 2.0;
            ad[j] = (a[j] + d[j]) / 2.0;
            cd[j] = (c[j] + d[j]) / 2.0;
            bs[j] = (b[j] + s[j]) / 2.0;
            as[j] = (a[j] + s[j]) / 2.0;
            cs[j] = (c[j] + s[j]) / 2.0;
            ds[j] = (d[j] + s[j]) / 2.0;
        }
        DividePyramid(a, ab, ac, ad, as, level - 1);
        DividePyramid(ab, b, bc, ac, bs, level - 1);
        DividePyramid(ac, bc, c, cd, cs, level - 1);
        DividePyramid(ad, ac, cd, d, ds, level - 1);
        DividePyramid(as, bs, cs, ds, s, level - 1);
    }
}
void DrawPyramid(GLfloat* a, GLfloat* b, GLfloat* c, GLfloat* d, GLfloat* s) //пирамидыг
зурах
{
    if (!state) //өнгөтэй бол
    {
        glBegin(GL_TRIANGLES);
        glColor3f(0.52, 0.44, 1.0);
        DrawTriangle(a, b, s);
        glColor3f(1.0, 0.84, 0.0);
        DrawTriangle(b, c, s);
        glColor3f(0.52, 0.44, 1.0);
        DrawTriangle(c, d, s);
        glColor3f(1.0, 0.84, 0.0);
        DrawTriangle(a, d, s);
        glEnd();
        glBegin(GL_QUADS);
        glColor3f(1.0, 0.84, 0.0);
        DrawQuoad(a, b, c, d);
        glEnd();
    }
    else //өнгөгүй бол
    {
        glBegin(GL_LINES);
        // blue
        glColor3f(0.0, 0.0, 1.0);
        DrawLineTriangle(a, b, s);
        DrawLineTriangle(c, d, s);
        DrawLineQuoad(a, b, c, d);
        glEnd();
    }
}
void DrawTriangle(GLfloat* a, GLfloat* b, GLfloat* c) //гурвалжин зурах
{
    glVertex3fv(a);
    glVertex3fv(b);
    glVertex3fv(c);
}

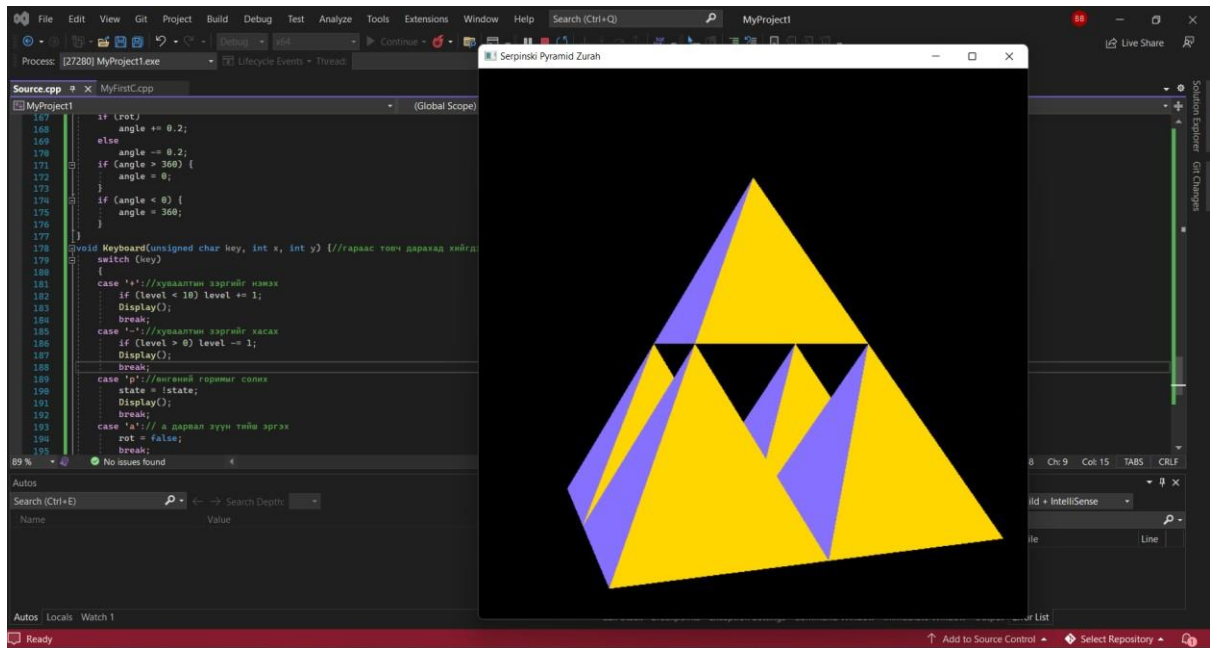
```

```

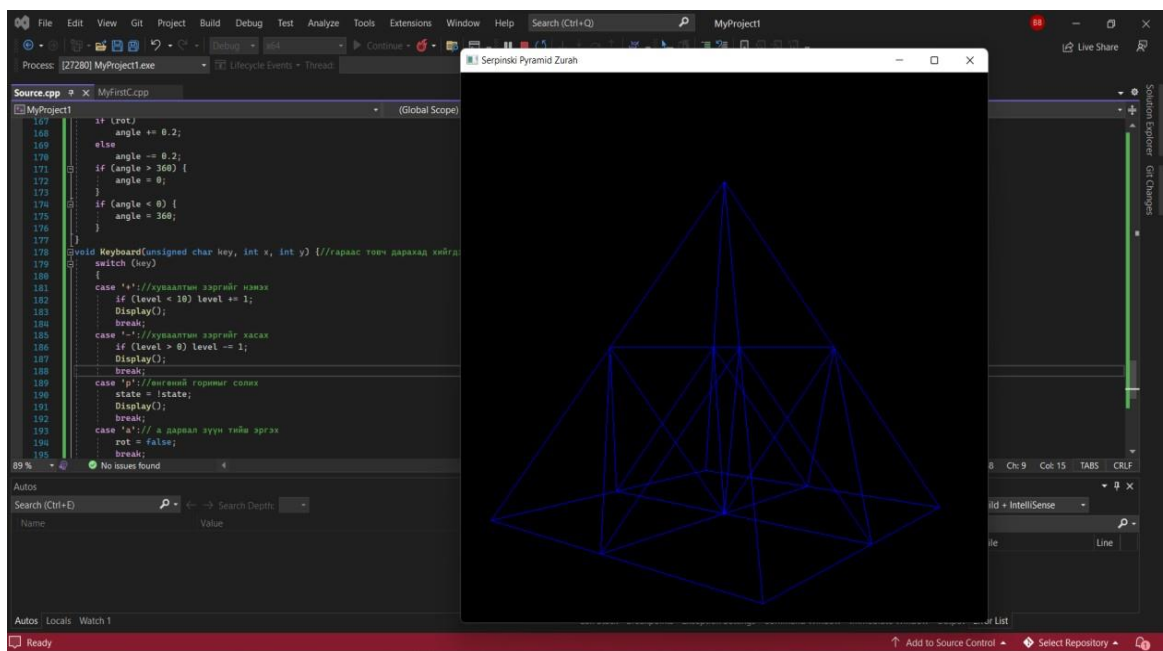
void DrawQuoad(GLfloat* a, GLfloat* b, GLfloat* c, GLfloat* d)//суурийн квадрат зурах
{
    glVertex3fv(a);
    glVertex3fv(b);
    glVertex3fv(c);
    glVertex3fv(d);
}
void DrawLineTriangle(GLfloat* a, GLfloat* b, GLfloat* c)
{
    glVertex3fv(a); //linestrip-ээр зурж байна
    glVertex3fv(b); // a-b
    glVertex3fv(b);
    glVertex3fv(c); //b-c
    glVertex3fv(a);
    glVertex3fv(c); //a-c pvv 4 удаа shuluun tataj байнаa
}
void DrawLineQuoad(GLfloat* a, GLfloat* b, GLfloat* c, GLfloat* d)
{
    glVertex3fv(a);
    glVertex3fv(b); //a-b
    glVertex3fv(b);
    glVertex3fv(c); //b-c
    glVertex3fv(c);
    glVertex3fv(d); //c-d
    glVertex3fv(a);
    glVertex3fv(d); //a-d
}
void Rotate()//эргүүлэх
{
    if (rot)
        angle += 0.2;
    else
        angle -= 0.2;
    if (angle > 360) {
        angle = 0;
    }
    if (angle < 0) {
        angle = 360;
    }
}
void Keyboard(unsigned char key, int x, int y) { //гараас товч дарахад хийгдэх үйлдлүүд
    switch (key)
    {
        case '+': //хуваалтын зэргийг нэмэх
            if (level < 10) level += 1;
            Display();
            break;
        case '-': //хуваалтын зэргийг хасах
            if (level > 0) level -= 1;
            Display();
            break;
        case 'p': //өнгөний горимыг солих
            state = !state;
            Display();
            break;
        case 'a': // a дарвал зүүн тийш эргэх
            rot = false;
            break;
        case 'd': // d дарвал зүүн тийш эргэх
            rot = true;
            break;
    }
}

```

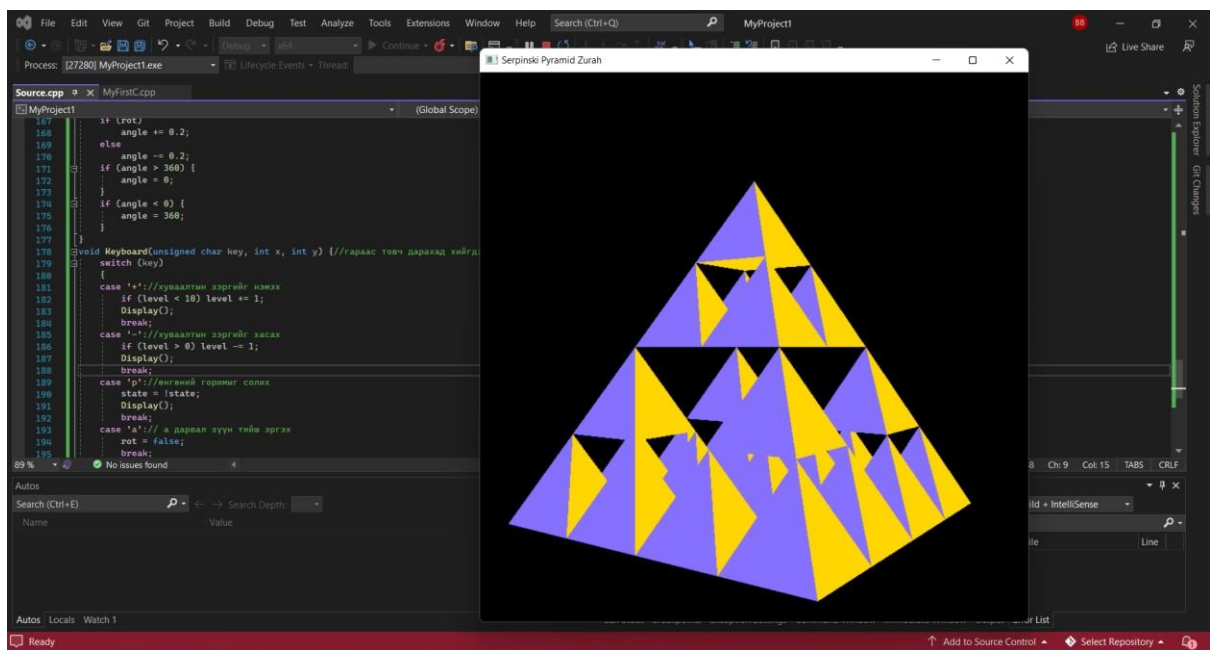
## Үр дүн харуулсан хэсэг



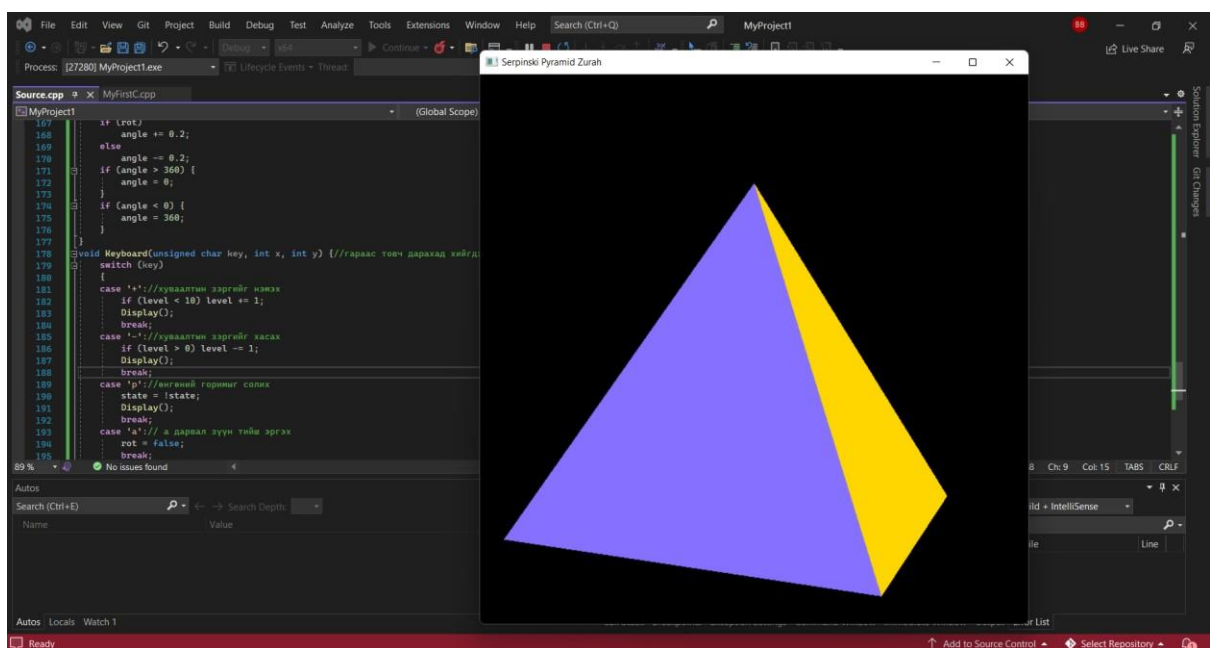
Зураг 1.1 Кодыг ажиллуулахад анхны



Зураг 1.2 Р товч дарахад буюу зураасан горимд шилжсэн үр дүн



Зураг 1.3 + тэмдэг дарахад



Зураг 1.4 - тэмдэг дарахад

## Нэмэлт ажил

Бие даалтын ажлаар нэмэлтээр кодонд оруулсан зүйл бол Серпинскийн гурвалжныг аль зүг рүү эргэхийг заах үйлдэл юм. Тодруулбал **a** дээр дарахад зүүн тийш, **d** дээр дарахад баруун тийш эргэнэ. Доорх линкээр орж үзнэ үү.

<https://www.youtube.com/watch?v=wLv0NrsVnRU>