

ШИНЖЛЭХ УХААН ТЕХНОЛОГИЙН ИХ СУРГУУЛЬ  
МЭДЭЭЛЭЛ, ХОЛБООНЫ ТЕХНОЛОГИЙН СУРГУУЛЬ

# ЛАБОРАТОРИЙН АЖИЛ

F.CS209  
ЛАБОРАТОРИЙН АЖЛЫН ТАЙЛАН

Гүйцэтгэсэн:

О.Нэр /Оюутны код/

Шалгасан:

Ч.Цэнд-Аюуш /Доктор (Ph.D)/

УЛААНБААТАР ХОТ  
2019 ОН

## Лаб: Фрактал

Фрактал: Тодорхой процессыг (давалт – iteration) давтан хийснээр үүссэн геометрийн дүрс.

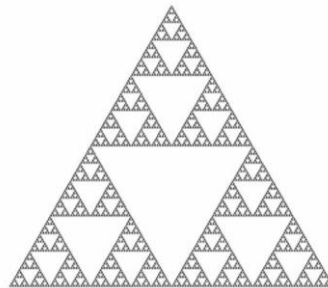
Фракталууд нь масштабын аль ч түвшинд нарийн төвөгтэй байдал нь өөрчилдөггүй. Энэ нь бид хичнээн томруулсан ч гэсэн хэлбэр нь адилхан харагдаж байна гэсэн үг юм.

Фрактал байгуулахын тулд бид "давалт" хэмээх хэрэгслийг ашигладаг. Энэ нь анхны дүрс (ихэвчлэн энгийн геометрийн хэлбэр) дээр алхамыг хэрэгжүүлж, үр дүнг нь авч, алхамыг дахин гүйцэтгэнэ гэсэн үг юм. Фракталын хувьд бид энэ процессыг хязгааргүй олон удаа давтдаг (энэ нь хэзээ ч дуусдаггүй).

### Сиерпинскийн гурвалжин (Sierpinski Triangle)

Өнөөдөр бид Сиерпинскийн гурвалжин хэмээх фрактал байгуулна. Фракталыг байгуулахад дараах алхмуудыг хэрэгжүүлнэ.

1. Адил талт гурвалжин зур
2. Гурван талын дунд цэгүүдийг холбоно уу
3. Үүссэн доошоо доошоо гурвалжинг арилгаж (сүүдэрлэж эсвэл хайчилж ав)
4. Дунд гурвалжинг хасна
5. Үлдсэн жижиг гурвалжнуудын хувьд 2-3-р алхамыг давтана
6. 4-р алхамыг хязгааргүй олон удаа давт!



### Сиерпинскийн гурвалжины талбай

Эхний тэгш өнцөгт гурвалжны талбайг  $1 \text{ нэгж квадрат}^2$  гэж үзье.



1. Эхний давалтын дараа Сиерпинскийн гурвалжины талбай утга?



Анхны дүрсийн  $\frac{1}{4}$  устгагдсан ба гурвалжины талбай  $1 \times \frac{3}{4} = \frac{3}{4}$  нэгж квадрат.

2. 2,3 дахь давталтын дараа гурвалжны тоо болон талбайн утга?
3. Гэх мэтчилэн ....n дэх давталтын утгыг хэрхэн тооцоолох вэ?
4. Програмчлах: Доорхи кодыг ашиглан Serpynsky Pyramid байгуулна.

```
#include <iostream>
#include <glut.h>

using namespace std;

// functsiin 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);

// erguuleh ontsog
GLfloat angle = 45;
// davhraga tuvshin
int level = 1;
// irmeg zurah esvel budah: false-budah, true- only irmeg( P товч)
bool state = false;
// pyramid
GLfloat pyramid[5][3] =
{
    { -1.0, -1.0, 1.0 }, // A
    { 1.0, -1.0, 1.0 }, // B
    { 1.0, -1.0, -1.0 }, // C
    { -1.0, -1.0, -1.0 }, // D
    { 0.0, 1.0, 0.0 } // S
};

int main(int argc, char **argv)
{
    //initization
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
    // creating window
    glutInitWindowSize(700, 700);
    glutCreateWindow("Serpinski Pyramid Zurah");

    MyInit();
    glutDisplayFunc(Display);
    glutReshapeFunc(Reshape);
    glutKeyboardFunc(Keyboard);
    glutIdleFunc(Rotate);

    glutMainLoop();
}
```

```

        return 0;
    }

void MyInit()
{
    // black background
    glClearColor(0.0, 0.0, 0.0, 1.0);
    // разрешить тест глубины
    glEnable(GL_DEPTH_TEST);
    // perspective saiiruulah
    glHint(GL_PERSPECTIVE_CORRECTION_HINT, GL_NICEST);
}

void Display()
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    // translated into (ecran ruu) 4 point
    glTranslatef(0.0, 0.0, -4.0);
    // Y tenhlegiin daguu erguuleh
    glRotatef(angle, 0, 1, 0);
    // level > 0 uyd pyriamidaa huvaaj ogno, dahin zurna
    DividePyramid(pyramid[0], pyramid[1], pyramid[2], pyramid[3], pyramid[4], level);
    //display-g davtah (ergeltiin ontsgiiг bnga solihguin tuld zurah functsiig dahin duudah dohio
    ogno)
    glutPostRedisplay();
    glutSwapBuffers();
}

void Reshape(int w, int h)
{
    glViewport(0, 0, w, h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    // tsonhnii perspektiv
    gluPerspective(45.0, (GLdouble)w / (GLdouble)h, 0.1, 200.0);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}

void DividePyramid(GLfloat *a, GLfloat *b, GLfloat *c, GLfloat *d, GLfloat *s, int level)
{
    // irmegiin goloon koordinat
    GLfloat ab[3], bc[3], ac[3], ad[3], bs[3], as[3], cs[3], ds[3];
    if (level == 0)
    {
        DrawPyramid(a, b, c, d, s);
    }
    else
    {
        for (int j = 0; j < 3; j++)
        {
            ab[j] = (a[j] + b[j]) / 2.0; // A ba B dundaj
            bc[j] = (b[j] + c[j]) / 2.0; // B,C
            ac[j] = (a[j] + c[j]) / 2.0; // A ba C (kvadratiin tov)
            ad[j] = (a[j] + d[j]) / 2.0; // A ba D
        }
    }
}

```

```

        cd[j] = (c[j] + d[j]) / 2.0; // C ба D
        bs[j] = (b[j] + s[j]) / 2.0; // B ба S
        as[j] = (a[j] + s[j]) / 2.0; // A ба S
        cs[j] = (c[j] + s[j]) / 2.0; // C ба S
        ds[j] = (d[j] + s[j]) / 2.0; // D ба S-ийн дундаж
    }

    // recursive дуудалт (5n жигж pyramidiin huvid)
    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)
{
    // будах
    if (!state)
    {
        glBegin(GL_TRIANGLES);
        // purple ongo
        glColor3f(0.52, 0.44, 1.0);
        DrawTriangle(a, b, s);
        // yellow
        glColor3f(1.0, 0.84, 0.0);
        DrawTriangle(b, c, s);
        // purple
        glColor3f(0.52, 0.44, 1.0);
        DrawTriangle(c, d, s);
        // yellow
        glColor3f(1.0, 0.84, 0.0);
        DrawTriangle(a, d, s);
        glEnd();

        glBegin(GL_QUADS);
        // yellow
        glColor3f(1.0, 0.84, 0.0);
        DrawQuoad(a, b, c, d);
        glEnd();
    }
    // esreg tohioldold zovhon irmegee зурна
    else
    {
        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)
{
    glVertex3fv(a);
    glVertex3fv(b);

```

```

        glVertex3fv(c);
    }
void DrawQuoad(GLfloat *a, GLfloat *b, GLfloat *c, GLfloat *d)
{
    glVertex3fv(a);
    glVertex3fv(b);
    glVertex3fv(c);
    glVertex3fv(d);
}
void DrawLineTriangle(GLfloat *a, GLfloat *b, GLfloat *c)
{
    glVertex3fv(a);
    glVertex3fv(b);
    glVertex3fv(b);
    glVertex3fv(c);
    glVertex3fv(a);
    glVertex3fv(c);
}
void DrawLineQuoad(GLfloat *a, GLfloat *b, GLfloat *c, GLfloat *d)
{
    glVertex3fv(a);
    glVertex3fv(b);
    glVertex3fv(b);
    glVertex3fv(c);
    glVertex3fv(c);
    glVertex3fv(d);
    glVertex3fv(a);
    glVertex3fv(d);
}
void Rotate()
{
    angle += 0.05;
    if (angle > 360) angle = 0;
}
void Keyboard(unsigned char key, int x, int y) {
    switch (key)
    {
        case '+':
            if (level < 10) level += 1;
            Display();
            break;
        case '-':
            if (level > 0) level -= 1;
            Display();
            break;
        case 'p':
            state = !state;
            Display();
            break;
    }
}

```

5. Биет бүтээл гаргах (Бүтээлийнхээ зургийг тайланд хавсаргах)

Тайлан хүлээн авах сүүлийн хугацаа: 2020 оны 10 сарын 27 цаг: 23:00

Тайлан авах хэлбэр: Файлаар (Moodle)