컴퓨터그래픽스

과제 1:

당구공 애니메이션

201903868 컴퓨터전자시스템공학과

황가은

<소스코드>

```
#include <GL/glut.h>
#include <GL/glu.h>
#include <GL/gl.h>
#include <math.h>
#include <time.h>

GLfloat radius = 25;
GLfloat sugu_x, sugu_y, sugu_x_dir, sugu_y_dir;
GLfloat red_x, red_y, red_x_dir, red_y_dir;
GLfloat dx, dy = 25;
GLint check, crash, count_sugu, count_red = 0;
GLfloat reduce;

GLfloat six_c = cos(60 * 3.141592 / 180);
GLfloat six_s = sin(60 * 3.141592 / 180);
GLfloat thr_c = cos(30 * 3.141592 / 180);
GLfloat thr_s = sin(30 * 3.141592 / 180);

void begin_vertex()
{
    sugu_x = ((rand() % 600) - 300);
    sugu_y = -1 * (rand() % 400);
    red_x = ((rand() % 600) - 300);
    red_y = ((rand() % 400));
}
```

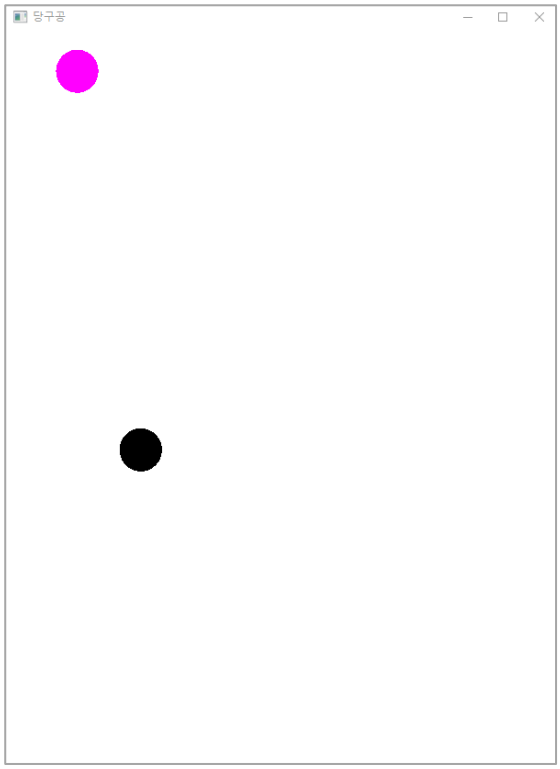
```
void begin_value()
{
    if (sugu_x > red_x)
    {
        sugu_x_dir = -1 * sqrt(pow(sugu_x - red_x, 2)) / 25;
    }
    else
    {
        sugu_x_dir = sqrt(pow(sugu_x - red_x, 2)) / 25;
    }
    sugu_y_dir = sqrt(pow(sugu_y - red_y, 2)) / 25;

    reduce = 1.8;
}

void Timer(int Value) {
    glutPostRedisplay();
    glutTimerFunc(10, Timer, 1);
}

void sugu_circle(float cx, float cy, float r)
{
    float x1, y1, x2, y2;
    float angle;
    double radius = r;


    x1 = cx;
    y1 = cy;
    glColor3f(0.0, 0.0, 0.0);

    glBegin(GL_TRIANGLE_FAN);
    glVertex2f(x1, y1);

    for (angle = 0.01f; angle <= 2 * 3.14f + 0.2; angle += 0.1)
    {
        x2 = x1 + sin(angle) * radius;
        y2 = y1 + cos(angle) * radius;
        glVertex2f(x2, y2);
    }
    glEnd();
}
void red_circle(float cx, float cy, float r) {
    float x1, y1, x2, y2;
    float angle;
    double radius = r;


    x1 = cx;
    y1 = cy;
    glColor3f(1.0, 0.0, 1.0);

    glBegin(GL_TRIANGLE_FAN);
    glVertex2f(x1, y1);
```

```
    for (angle = 0.01f; angle <= 2 * 3.14f + 0.2; angle += 0.1)
    {
        x2 = x1 + sin(angle) * radius;
        y2 = y1 + cos(angle) * radius;
        glVertex2f(x2, y2);
    }
    glEnd();
}


void Display()
{
    if (check == 0)
    {
        glViewport(0, 0, 600, 800);
        glClear(GL_COLOR_BUFFER_BIT);

        sugu_circle(sugu_x, sugu_y, radius);
        red_circle(red_x, red_y, radius);

        glutSwapBuffers();
    }

    if (check == 1)
    {
        glClear(GL_COLOR_BUFFER_BIT);
        sugu_circle(sugu_x + sugu_x_dir, sugu_y + sugu_y_dir, radius);
        red_circle(red_x + red_x_dir, red_y + red_y_dir, radius);

        if (sqrt(pow(sugu_x - red_x, 2) + pow(sugu_y - red_y, 2)) <= 2 * radius)
        {
            crash++;
            if (crash < 2)
            {
                sugu_x_dir = (0.5 * sugu_x - (-1) * (sqrt(3) / 2) * sugu_y) / 25;
                sugu_y_dir = ((-1) * (sqrt(3) / 2) * sugu_x + 0.5 * sugu_y) / 25;

                red_x_dir = ((sqrt(3) / 2) * red_x - 0.5 * red_y) / 25;
                red_y_dir = (0.5 * red_x + (sqrt(3) / 2) * red_y) / 25;
                crash++;
            }
            else if (crash == 2)
            {
                sugu_x_dir = 0;
                sugu_y_dir = 0;

                red_x_dir = 0;
                red_y_dir = 0;
            }
        }
        sugu_x += sugu_x_dir;
        sugu_y += sugu_y_dir;

        red_x += red_x_dir;
        red_y += red_y_dir;
```

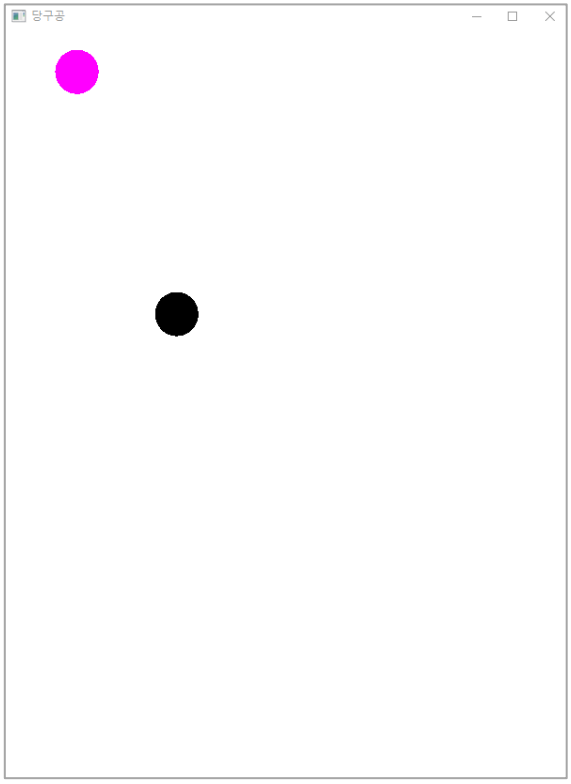
```
        if ((sugu_x_dir < 0 && sugu_x <= -300) || (sugu_x_dir > 0 && sugu_x >= 300))
        {
            sugu_x_dir *= -1;
            sugu_x_dir, sugu_y_dir /= reduce;
            count_sugu++;
        }
        if ((sugu_y_dir < 0 && sugu_y <= -400) || (sugu_y_dir > 0 && sugu_y >= 400))
        {
            sugu_y_dir *= -1;
            sugu_x_dir, sugu_y_dir /= reduce;
            count_sugu++;
        }
        if ((red_x_dir < 0 && red_x <= -300) || (red_x_dir > 0 && red_x >= 300))
        {
            red_x_dir *= -1;
            red_x_dir, red_y_dir /= reduce;
            count_red++;
        }
        if ((red_y_dir < 0 && red_y <= -400) || (red_y_dir > 0 && red_y >= 400))
        {
            red_y_dir *= -1;
            red_x_dir, red_y_dir /= reduce;
            count_red++;
        }

        if (count_sugu >= 2)
        {
            sugu_x_dir /= 1.2;
            sugu_y_dir /= 1.2;
        }
        if (count_red >= 3)
        {
            red_x_dir /= 1.2;
            red_y_dir /= 1.2;
        }


        glutSwapBuffers();
    }


}

void Key(unsigned char key, int x, int y)
{
    if (key == 'q' || key == 'Q')
    {
        check = 1;
    }
    else if (key == 'w' || key == 'W')
    {
        sugu_y += 5;
        check = 0;
        begin_value();  //위치를 이동시키고 나서도 적구에 올바르게 맞을 수 있도록 값을 재조정.
```

```
        glutPostRedisplay();
    }
    else if (key == 's' || key == 'S')
    {
        sugu_y += -5;
        check = 0;
        begin_value();
        glutPostRedisplay();
    }
    else if (key == 'a' || key == 'A')
    {
        sugu_x += -5;
        check = 0;
        begin_value();
        glutPostRedisplay();
    }
    else if (key == 'd' || key == 'D')
    {
        sugu_x += 5;
        check = 0;
        begin_value();
        glutPostRedisplay();
    }
    else if (key == 27)
    {
        exit(0);
    }

}

int main(int argc, char** argv) {
    srand((unsigned)time(NULL));
    begin_vertex();
    begin_value();

    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE);

    glutInitWindowSize(600,800);
    glutCreateWindow("당구공");

    glClearColor(1.0, 1.0, 1.0, 1.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-325.0, 325.0, -425.0, 425.0, -1.0, 1.0);
    glutDisplayFunc(Display);
    glutKeyboardFunc(Key);
    glutTimerFunc(40, Timer, 1);
    glutMainLoop();

    return 0;
}
```

<화면 DUMP>



-> 수구 위치 이동



<소감>

다른 과목의 과제에도 쫓기다 보니 이 과제를 하는 데에 시간적으로 어려움이 있었습니다. 또한 실습시간에 배운 것보다 훨씬 어려운 수준의 과제라서 더 힘들었던 것 같습니다. 지금까지 CMD 창에서 봐왔던 것은 숫자 아니면 문자였는데 처음으로 움직이는 물체를 만들어봐서 참신했습니다.