CS 450/550 -- Fall Quarter 2025

Project #1

50 Points

Due: October 6

Draw Something Cool in 3D!

This page was last updated: July 6, 2025

Introduction

This project requires you to draw something fun and cool in 3D. *It must be your own creation.*

Just as an inspiration, here is what the CS 450/550 Class of 2022 did with this assignment, but I know you can do better.

Learning Objective:

When you are done with this assignment, you will understand how to generate and manipulate 3D graphics objects using OpenGL. At this point, you will realize that you can basically draw *anything* just by repeatedly applying what you did here.

Instructions

1. You can use pre-canned 3D objects, such as the ones that GLUT provides, or the OSU functions that we provide, or OBJ files, but *in addition* to your own, not instead of. You must create your own geometry.
2. You can explicitly list your own 3D coordinates, or you can use equations and define the 3D shapes procedurally.
3. You must have at least 100 x-y-z coordinates. These can be divided up among multiple of *your* objects.
4. The scene must have 3D thickness, nothing that is completely 2D planar.
5. You must use at least 5 different colors.
6. The 3D rotation and scaling from the sample program must still be working.

Getting Started

Go to the Class Resources Page and scroll down to **Downloadable Files**. Then download one of the SampleWindows.zip, SampleLinux.tar, or SampleMac.tar files (by right-clicking on them). This will produce a folder full of *all the other files you need.* You do not need to go hunt the internet for any other files. Use the ones that have been given to you.

Start by getting the sample program to work. If you are on Windows, double-click on the .sln file. If you are on Linux or Mac, type **make**

Then start modifying the function *InitLists( )* to draw something of your own design.

No, using a GLUT object or an OSU object or an OBJ file will not count.

Using Random Numbers in Computer Graphics:

In your sample code, you have a function called **Ranf( )**. Use it to select a random number between a low value and a high value. For example:

```
float red = Ranf( 0.f, 1.f );
```

The way random numbers work, you will get the same sequence every time you run your program. If you want to get a different sequence every time, seed the random numbers with the time of day. To do this, call the function:

```
TimeofDaySeed( );
```

as the first thing that your main program does.

How can you have fun with random numbers? Well, for example, you could pick a random color like this:

```
glColor3f( Ranf(0.f,1.f) , Ranf(0.f, 1.f), Ranf(0.f,1.f) );
```

or you could draw a random collection of triangles like this:

```
int numTriangles = 33;
glBegin( GL_TRIANGLES );
for( int t = 0; t < numTriangles; t++ )
{
        glColor3f( Ranf(0.f,1.f) , Ranf(0.f, 1.f), Ranf(0.f,1.f) );
        for( int v = 0; v < 3; v++ )
        {
                glVertex3f(  Ranf(-1.f,1.f) , Ranf(-1.f, 1.f), Ranf(-1.f,1.f) );
        }
}
glEnd( );
```

If you do this, it is just for fun. **It doesn't count as one of your Project #1 objects!**

Turn-in:

Use Canvas to turn in your:

1. Your .cpp file
2. A short PDF report containing:
    - Your name
    - Your email address
    - Project number and title
    - A description of what you did to get the display you got

- A cool-looking screen shot from your program
- The link to the Kaltura video demonstrating that your project does what the requirements ask for. If you can, we'd appreciate it if you'd narrate your video so that you can tell us what it is doing.

3. To see how to turn these files in to Canvas, go to our Project Notes noteset, and go the the slide labeled **How to Turn In a Project on Canvas.**

4. **Be sure that your video's permissions are set to *unlisted.***.
   The best place to set this is on the OSU Media Server.

5. A good way to test your video's permissions is to ask a friend to try to open the same video link that you are giving us.

6. The video doesn't have to be made with Kaltura. Any similar tool will do.

Grading:

Feature	Points
At least 100 vertices	20
At least 5 colors	20
3D rotation and scaling	10
Potential Total	**50**