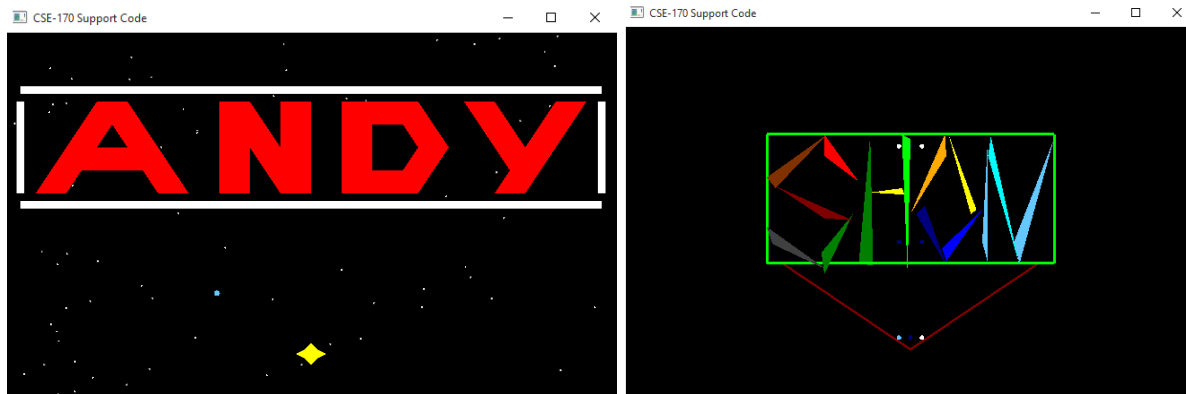


# Programming Assignment #1

In this assignment you will draw your name with OpenGL.

Below are two examples from previous years:



## First follow these steps:

1) Download PA1\_support\_code.zip from CatCourses. It includes a Microsoft Visual Studio 2022 solution, along with the freeglut, glm, and glew libraries we will be using. The four folders should be in the **same folder**, as the support code solution is linked with relative paths.

2) Double-clicking on BasicOpenGLProject.sln will open Visual Studio. The support code is configured for x64 and Release build, so set those settings up top if they are not already (to the left of "Local Windows Debugger"). At that point, if you then press F5 or Ctrl+F5 you should see the project compile and run without any issues. However, a variety of problems may occur: be sure to have the latest drivers for your graphics card, and if you are running a different version of Visual Studio you may need to retarget the project. Your TA will be helping with these steps. Go to lab as that is the best way to get help with this! *(Note that the included version of glew will only work on Release build. If you really want to run the project in Debug build, you can try copying glew32.dll to your system32 folder, and that should let Visual Studio find it.)*

3) When you are able to run the application, you will see three rgb lines along the x, y, and z axes. Clicking and dragging will rotate the scene and you can use the mouse wheel to "zoom" in and out, which should help you to visualize what you create.

The support code includes files to help with shader creation, loading, and use, and two basic shaders. There is already code to create the model, view, and projection matrices and send them to the perspective shader. It also shows how to create the necessary buffers and arrays of vertex information to render the lines for the x, y, and z axes, which serves as a useful example for your own objects.

Take some time to familiarize yourself with the support code. Read over the comments and try to understand all of the parts of the program. If there is anything you do not understand, refer to the lecture slides, any of the other OpenGL resources suggested, or ask your TA for help.

4) When you are ready, implement the requirements below.

### **Requirements:**

**Requirement 1 (40%): Draw at least the first 4 letters of your name using triangles.** You might want to first draw the letters on a piece of paper and write down the coordinates of each point that you will need to use to decompose the letters of your name in triangles. The letters can be simplified but they have to be legible. Although you will be viewing them in 3D, they do not necessarily have to vary in all 3 dimensions. For example, you can set  $z=0$  and only vary the  $x$  and  $y$  coordinates.

Follow what is already implemented in the support code. The support code already shows how to organize the coordinates and colors of vertices into buffers and how to make OpenGL calls to send the data to the GPU and for drawing the triangles encoded in the buffers. You will need to do the same, with a new VAO and VBOs for your object. (Although the letters may be disjoint, think of them as one "object".)

Note: For the built-in OpenGL "back-face culling" to work properly, make sure the order of the vertices in each triangle is always counter-clockwise, when seen from the front of the letters.

**Requirement 2 (40%): Interactive shape parameterization.** Use the callback functions to allow the user to control your letters in some way, via parameters. Include at least 3 parameters, with at least one using the keyboard and at least one using the mouse. For example, parameters to control the size of the letters, change the color or orientation, click and drag a letter to move it around, etc. Be creative.

Consider that you do not need to generate a new VAO and VBOs for your object when it changes, but merely send the new data to the GPU.

**Requirement 3 (10%): User friendliness.** Add functionality to make it possible for the user to press a certain key (for example, 'h' or '?') to print in the console a list of all the keys that control your application.

**Requirement 4 (10%): Overall quality.** Everything counts here: if requirements are well implemented, creativity, source code organization, etc. There is no need to do anything complex, just make sure your project looks good and you will get the full points here!

Notes:

- Many details will become clearer with the explanations and interactions that you will have during the labs. Plan to attend the labs and participate. Be sure to ask questions whenever you feel stuck somewhere.

### **Submission:**

Please follow the instructions in parules.txt (uploaded to CatCourses). In particular: please do not include any third-party support code and do not forget to Clean Solution before preparing your project for submission! Also, check for hidden folders (such as .vs) which can sometimes balloon to hundreds of megabytes!