12:00:08 From Nawaz,Usman: So the testing file you gave on your website are they supposed to show something in 3-D when we run it?

Yes, a colorful cube. Use the left mouse button to rotate. Use the middle mouse button to scale.

12:04:10 From Kocabagli, Sarp: If anyone is using macOS, How did you run the sample code?
12:04:23 From Jacob Eckroth(He/him/his): this might be helpful: https://piazza.com/class/kcdnnyg6a3z655?cid=12

Thanks for posting this!

12:09:12 From Conklin, Cade Thomas: What language will primarily be used?

C/C++. I call it C++ and we use a .cpp source file, but we use very few C++-isms. OpenGL was designed as a C-callable API only.

12:19:26 From Conklin, Cade Thomas: Can Z go in a negative direction?

Yes, the “right hand coordinate system” just controls what the axes look like, i.e., what direction Z goes. You can specify x, y, and z coordinates + or – the limits of IEEE 32-bit floating point. That range is approximately $10^{-38}$ to $10^{37}$.

12:23:15 From Land, Hunter: Do GPUs only use float32 rather than anything like doubles?

There is a glVertex3d( ) function that takes x, y, and z as doubles. Whether or not it actually uses them as doubles depends on your graphics hardware.

12:23:36 From Sterrett, Matthew B: Math with doubles is much slower, so floats are mostly used.
12:24:30 From Guidos, Job M: most GPUs can compute doubles. its like 32 times slower on most cards
12:24:55 From Sterrett, Matthew B: I just looked it up, and the RTX 2080ti can do 13.45 terraflops when using floats, but only 420.2 gigaflops when using doubles

Not surprising.

12:29:21 From Land, Hunter: Why wouldn't GL_POLYGON use triangle strips under the hood?

The configuration of the vertices may not allow it. Under the hood, I would guess that a GL_POLYGON just gets cracked up into separate triangles.
12:29:51    From Kocabagli, Sarp : Do the polygons have to be convex?

There is no rule enforcing this (i.e., there are no Convex Police out there to arrest you), but OpenGL doesn’t guarantee you will get the right polygon appearance if you don’t do it.

12:31:27    From Nieh, Shu-Kan : is it possible to create an object by several different Topologies?
For example:
gl_begin(1st topology)
...
gl_end()
gl_begin(2nd topology)
...
gl_end()

Yes, that will work perfectly. That’s how you are expected to do it.

12:39:20    From Pannapat Chanpaisaeng : So a star is a non-convex polygon…?
Correct.

12:39:26    From Mahmoud, Ibrahim : can we use vertex buffer objects and arrays?
Yes.

12:43:42    From Lee, Vincent Nicholas : from a game standpoint will this class get into why the gpu can only be one frame behind the current frame having its logic run on the cpu

I don’t know for sure, but I think this is an API limitation, not a GPU limitation. OpenGL, for instance, can only do double-buffering, while Vulkan can do up to 8-way buffering.

12:44:12    From Nguyen, Vinh : how small can we make a color value without it being 0?

(1./255.) I would guess.

12:44:36    From Zhao, Liang : If I know the color, how do I know the RGB value? I don’t want try all value

Go to our Resources web page and download ColorPicker.exe
Put it in the Debug folder of your sample project so that when you run it, it can get at the same .dll files,
12:52:06 From Jacob Eckroth(He/him/his) : what range of numbers can we input into glScalef? Is it still 0.0-1.0, and if so what does that mean for the scaling?

Whatever numbers you want! You can scale an object by 1000000. if you want. You can also scale by 0.00001 -- it’s just math. (Both of those examples would be pretty bad ideas)

12:56:44 From Lloyd, Doug : how do you designate which things are getting transformed if you have multiple glBegins? does it apply to all?

It applies to everything that gets drawn after the transformation has been set. The act of drawing something takes every x, y, z from a glVertex3f( ) call and transforms it by the current transformation. And, OpenGL doesn’t care if you set a transformation, draw some things, change that transformation, draw some more things, and so on.

13:04:37 From Guidos, Job M : if you don't want to think of it like matrices, you can imagine pitch, roll, and yaw of an airplane. if you pitch down and roll right, you end up in a different orientation than if you roll right then pitch down.

Good analogy! But really, just think of it like “some transformations have been defined – I don’t care how they get stored – just use them”.

13:04:41 From Alamgir, Jawad : what does translate do?

Moves the object by adding numbers to the x, y, and z coordinates.

13:10:55 From Pannapat Chanpaisaeng : what’s a cockpit window?

The horizontal window in front of where the pilots are sitting.

13:26:53 From Golletz, Jackson Jerome : Is there a significant difference between HSL and HSV? I've seen both of them used in different applications

They are roughly the same thing. Hue and Saturation are exactly the same thing. How Lightness and Value fit in are slightly different.

13:32:41 From Toby Parrish : Essentially light is additive color and materials are subtractive. A screen emits light while paint and other materials absorb light

Well put!

13:35:07 From Grant Haines : Does the rasterizer do anything with antialiasing, or is that applied later?.

The rasterizer can do antialiasing, but you need to setup OpenGL to do that. In general, OpenGL’s defaults are “whatever runs fastest”. That means, by default, antialiasing is turned off.
13:39:16 From Lloyd, Doug: is most of our work in project 1 going to be in changing the glVertex3f calls under glBegin? just want to make sure the scope of project 1 doesn't seem like much

Yes, that’s it. Project #1 is a thinly-disguised excuse to get you to compile and run OpenGL. That’s new and foreign enough without building a more sophisticated assignment on top of it. Project #2 will be more challenging. Although, feel free to make Project #1 as wild and crazy as you’d like.

13:40:03 From He, Zhiyi: Could we adjust the sample code as the submission of assign1? Or we should create a brand new one?

Yes, please, please, please just modify the sample code! I don’t want anyone re-typing all that boilerplate. Your time is more valuable than that.

13:48:33 From Zhao, Liang: Why is OpenGL referred to as a large state machine? What’s that mean?

It means that every time you draw something, a whole bunch of parameters (the state) get applied to the thing being drawn: color, transformation, surface normal, texture coordinates, line width, etc.

13:51:37 From Nguyen, Quan M: can we draw a bunch of cubes with different colors?

Yes. If it was me, I would create multiple display lists with a differently-colored cube in each one.

13:52:07 From Zhao, Liang: If I don’t like GLUT, can I use GLFW? GLFW maybe looks more popular.

Yes, using GLFW would be fine. I gravitate to GLUT for this class because of the built-in 3D objects. Who wouldn’t like to make a scene with a teapot? 😊

13:52:14 From Pannapat Chanpaisaeng: any tutorial on how to make the submission video?

Yes: [http://cs.oregonstate.edu/~mjb/cs557/Handouts/kaltura.1pp.pdf](http://cs.oregonstate.edu/~mjb/cs557/Handouts/kaltura.1pp.pdf)

13:52:46 From Mills, Arthur M: I haven't run the sample.cpp on the OSU server, but after you type make how do you run the program?

If, by “OSU Server”, you mean flip – don’t. flip does not have a graphics card in it.

13:54:05 From Land, Hunter: Is our shape for project 1 going to be a wireframe or solid-faced object?

Either one will be fine. Unless you already know how to do lighting, the wireframe will be more intelligible.
13:57:04 From Campa, Victor Daniel: would it be recommended to use GLUT, GLEW if we want to follow along with you?

GLEW serves a different purpose than GLUT and GLFW do, so you want to use that no matter what. After that, you can pick GLUT or GLFW. I have a noteset on GLUT. I didn't do one on GLFW, although you can get some information from the Vulkan notes: http://cs.oregonstate.edu/~mjb/vulkan/Handouts/GLFW.1pp.pdf

13:58:03 From Yuan, Tianle: is the glutMainLoop() the same as the while-loop in GLFW?

Yes.

14:00:22 From Lloyd, Doug: what does glNewList and glEndList do again?

The first begins a display list. From then on, all drawing calls go into GPU memory instead of to the screen. The second ends drawing to the GPU memory. From then on, drawing calls go to the screen.

14:03:49 From Hastings, Arman Keegan: Oh, so that other class goes over GLSL?

Yes, CS 457/557. We cover some GLSL in this class, but in 457/557, we do it for 10 weeks.

14:20:31 From justiner: My cat just snuck up and sneezed on me mid-question.

Ewwwwww…

14:21:46 From Hernandez, Fransisco Javier: you have office hours every day except Saturday right?

That’s correct.

14:36:39 From Zhao, Liang: Use this for the Mac

```c
#include "glew.h"
#include <OpenGL/gl.h>
#include <OpenGL/glu.h>
#include "glut.h"
```

Thanks for posting this!