10:18:06 From Jacob Eckroth(He/him/his) to Everyone: is that why when we rotate the whole scene it looks a little strange? [when doing cube-mapping]

Yes. Use the object-transformation part of glman to rotate/scale/translate just the object, like we did today.

10:21:08 From Leathem, Selma to Everyone: I thought I saw that project4 is due next week and not this one. Is that correct?

That is correct. It is due February 9. Don’t ignore Project #4, but you don’t need to kill yourself on it this week.

10:22:12 From Koning, Jonathan Scott to Everyone: could you add a second camera view to fake a reflection? [inter-reflection]

Yes, although it becomes sort of unwieldy. You would then have to generate 6 image views from the new camera.

10:24:23 From Leathem, Selma to Everyone: if zero is returned to the mix function from refraction does it then use reflection?

When refraction angles do not work (i.e., when the sine of an angle tries to be > 1. Or < -1.), the built-in refract( ) function returns vec3(0.,0.,0.). In the example you saw, the code just intercepts it at that point and calls the reflect( ) function instead:

```cpp
vec3 refractVector = refract( ??? );
vec4 refractColor;
if( all( equal( refractVector, vec3(0.,0.,0.) ) ) )
{
    refractColor = reflectColor;
}
else
{
    refractColor = texture( ??? );
    refractColor = mix( refractColor, WHITE, 0.30 );
}
```

The built-in functions equal( ) and all( ) are there to help you do if-tests on vec’s. In this case, equal( ) will return a bvec3 (3-element bool) and all( ) will return a scalar bool.

10:43:22 From Todankar, Diksha Pritam to Everyone: what are some good values for uka, ukd, uks? we did use a slider for proj3 so we could play with it....but when we are not using a slider what values would work?

The biggest thing is to make sure those 3 values sum to no more than 1.0, otherwise the object is giving off more energy than it is taking in. After that, it is largely up to personal taste.
10:52:25 From Todankar, Diksha Pritam to Everyone: I am guessing when they were converting old grayscale movies to color, they just increased the saturation on it....

No, they artificially colorized it by hand. They couldn’t just increase the saturation because they never recorded the RGB in the first place. That’s why it often (always?) looks so bad. Really, B&W movies should just stay that way.

10:52:56 From Leathem, Selma to Everyone: how do they do technicolor?

I am not familiar with all the details, but I know it was an early way to film directly in color using a variety of “hacks” to get around the fact that color negative film for cameras didn’t exist yet. I know they used something called the “three-strip film” method where there were 3 strips of B&W film running through the movie camera at the same time, each with an R, G, and B filter in front of them. That way, they could capture grayscale images of each color and could recombine them later. I think that the Oz scenes in *The Wizard of Oz* were done that way. Does anyone know more?

10:55:28 From Todankar, Diksha Pritam to Everyone: the movies looked highly saturated though....which is why I guessed

I believe that was a characteristic of the TechniColor process itself, not a deliberate goal...

10:57:03 From Koning, Jonathan Scott to Everyone: You mean Kansas isn't just grey?

The massive sunflower fields in summer are amazing! But, 424 miles is a long way to drive.

11:00:09 From Todankar, Diksha Pritam to Everyone: can we use glm with glman

No reason to. GLM is something that your C/C++ program uses. glman is there to spare you from writing a C/C++ program. glman could have used GLM *internally*, but it didn’t.

11:01:27 From Leathem, Selma to Everyone: Do the shaders use GLM?

No, because GLM is made for your C/C++ program to use. Your shaders can use the *matrices* (etc.) that GLM produces, but your shaders cannot tell if a 4x4 matrix came from the deprecated OpenGL calls (e.g., glRotatef) or from GLM.

11:02:11 From Leathem, Selma to Everyone: Is vec3 glm?

GLM has a typedef called vec3 that is the same concept as GLSL uses when it uses a vec3. The GLM-people deliberately made GLM typedefs look like the GLSL variable types you already know. But, other than looking and behaving the same, there is no automatic connection between GLM vec3’s and GLSL vec3’s, although using the glslprogram C++ class, you can pass a uniform GLM vec3 into a GLSL vec3.

11:03:36 From Leathem, Selma to Everyone: does shaders use glsl?

Yes, the language that you have been using since Project #1 is called GLSL (GL Shading Language). There are other (similar) shading languages, such as HLSL for Direct3D. GLSL was made to go with OpenGL.

11:12:00 From Neiger, Kevin Daniel to Everyone: Is GLM going to be on the midterm?

No.