

Extra credit offer for Project 3

1 Description

Add an item 'toggle environment' to your subdivision program. When it is selected, it should use an image from a ppm file 'input.ppm' as a spherical environment map for the currently displayed object (either the original mesh or subdivided one). When it's selected again, your code should show the flat shaded subdivided surface again. Try the provided jpeg images as the environment maps. Convert them to ppm format using gimp. To obtain full credit you should:

1. Keep the environment fixed when the object rotates. The sky should always be up!
2. Optimize your code: use display lists (with correct memory management) and don't waste time calling `glTexImage()` when just drawing a frame.
3. Of course, get the correct image!

Here is how this part is going to be graded:

1. Image correct except for possibly rotation - 10pts
2. Environment fixed - 10pts
3. Optimizations and stability of the code - 10pts.

2 OpenGL commands

Relevant commands:

`glTexImage()`, `glMatrixMode()` (with `GL_TEXTURE` argument – to 'rotate' the texture coordinates as the object rotates), `glTexCoord*()`. Possibly also `glTexGen()`, if you want to use automatic texture generation.

Before you use the texture (before `glBegin()`), you need to enable texturing (`glEnable(GL_TEXTURE_2D)`), set the way the texture relates to the lighting (this is performed by `glTexEnvf` command; for an RGB texture (like ours), use it with arguments `GL_TEXTURE_ENV`, `GL_TEXTURE_ENV_MODE`, `GL_REPLACE` if you want to disable lighting and just apply the texture; change the last argument to `GL_MODULATE` if you want the lighting to be combined with the texture map (this way you will see both highlights and the reflection of the environment on your object)). Also, make the magnification/minification filters just do linear interpolation by calling

```
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
```

This will cause OpenGL to use your texture no matter what the size of the target object is. A more elegant way of doing this would be through *mipmapping*, i.e. providing different resolutions of the texture you wish to use.