Assignment 2 - Transmittance

Due: 27th Oct, 2016

Objectives

PART 1:

- (a) Implement a shader for an object that is both transmissive AND reflective;
  - Transmission may entail any of the effects discussed in the last lecture but most likely refraction
  - You must include a Fresnel component in your shader
  - You should also include the chromatic dispersion effect
  - You should also include an environment texture , cube map or sphere map in your scene which affects the appearance of your object

PART 2 (Discussed next week):

- Add bump mapping to your scene
  - This may be to your object above and/or to additional objects in the scene

SECONDARY OBJECTIVES:

- Implement a scene with some rotating objects using the above shaders.
  - Try to make the scene it as photorealistic as possible.
  - Try to add some variation in models, scene, shader to make your demo slightly unique.

Submission Details

- You must demo this in the lab on 27th of October at 5 pm (however you are strongly encouraged to complete the refraction bit next week)
- You must also submit by on 28th October, by email
  - a short (less than 5 minutes) video of your demo
  - A zip file including source code and shader code for your program (Source code only do not include executable)
  - In the submission add a short description of your scene and mention any external libraries, 3rd party source code you may have used (max 1 paragraph)
- You should work on your own. You may use and refer to external code but should reference it (see above) and in code comments
- You should use GLSL

Assessment

- This two week assignment is worth ~16% of the module
  - And is further broken down as follows:

Reflection	~10%
Refraction	~5%
Fresnel (for ratio of reflectance to transmittance)	~5%
Chromatic Dispersion	~10%
Environment texture or cube map	~10%
Normal map (discussed next week)	30%
Complexity of implementation/scene & any additional work taken on e.g. personalizing the scene, trying other approximations of Fresnel, attempting something other than refraction etc.	~25%
Video	~5%

Helper Notes

Refraction and Reflection

- You should refer to notes and read the relevant section 14.1 of the OpenGL Shading Language Book
  - https://www.scss.tcd.ie/Michael.Manzke/CS7055/GLSL/GLSL-3rdEd-refraction.pdf
- We are going to leave reflection for you to figure out but there are tonnes of examples to find (including in the orange book), and it is a minor variation on refraction (there is a GLSL function **reflect** that may be useful)

Fresnel Effect

- The full fresnel equation is fairly complex and there are various different approximations being used.
- Details are provided in the lecture notes

$$R_F(\theta) \approx R_F(0) + (1 - (\mathbf{h}.\mathbf{n}))^5 \times (1 - R_F(0)) \quad \text{and} \quad T_F = 1 - R_F$$

- Some alternative examples include:
  - http://en.wikibooks.org/wiki/GLSL_Programming/Unity/Specular_Highlights_at_Silhouettes
  - For this lab all are equivalent as long as the overall effect is preserved (i.e. incidence angle dependency of reflectance/refraction ratio, chromatic dispersion)

Environment Texture / Cube Map

- Most marks will be awarded for this element if you have anything (even just 2d) like an environment being reflected/refracted off your object. Some details on cube map set up follows.
- This requires some application stage OpenGL setup.
- For details See chapter 9 of the OpenGL Superbible
  - Excerpt and sample code from the Superbible 4th edition is available at the following link (you only need to read p 357-362)
  - http://www.scss.tcd.ie/Michael.Manzke/cs7055/Lab2
- In the GLSL shader, it should then be as simple as:

```
//fragment shader
uniform samplerCube CubeMap ;
varying vec3 R; // refracted vector
void main () { gl_FragColor = textureCube ( CubeMap , R); }
```

Environment Textures

- Refraction and reflection tend to distort the texture, magnifying it in areas (i.e. texels are mapped onto more pixels than intended)
- Thus, good environment textures should ideally be reasonably high-res and detailed in colour range. This is where the use of HDR (High-Dynamic-Range) images comes in for environment maps and light probes. N.B. HDR images however comes in special non-conventional formats
- Some sample cubemap textures
  - http://www.pauldebevec.com/Probes/ (look at the very bottom for some LDR tifs which should be the easiest to use)
  - http://www.codemonsters.de/home/content.php?show=cubemaps
  - http://www.humus.name/index.php?page=Textures (some very high res)
- Some large polygonal models (if you do not have a loader, feel free to use glutSolidTeapot, glutSolidTorus, etc..):
  - http://www.cc.gatech.edu/projects/large_models/

References

- OpenGL Shading Language “Orange Book” 2nd Edition (Not hosted here) :
  - http://wiki.labomedia.org/images/1/10/Orange_Book_-_OpenGL_Shading_Language_2nd_Edition.pdf
- OpenGL Superbible 4th Edition. (Not hosted here)
  - http://www.doc.ic.ac.uk/~af909/Addison.Wesley.OpenGL.SuperBible.4th.Edition.Jun.2007.pdf
- A [Third-party] Cubemap Tutorial:
  - http://www.keithlantz.net/2011/10/rendering-a-skybox-using-a-cube-map-with-opengl-and-glsl/

Submission from last year

http://www.youtube.com/watch?v=d0ZBMl4hhpw