CS 300 |

Assignment 3| Dynamic Reflection and Refraction with Cube Mapping

Files (submit folder) due
Week 14, Friday 23:59:59

Description

In this assignment, you will implement the following functionality:

1. Construct the 6 texture maps for cube mapping algorithm **[Grade: 15%]**
    1. The 6 textures are to be generated **at runtime every frame**. Use the FBO functionality to construct an **FBO with 6 render-to-texture targets/attachments**.
    2. To construct one side of the cube map:
        - Position the camera at the center of the reflecting/refracting object.
        - Aim the camera along one of the world space axis (+X, -X, etc.)
        - Set the camera field of view to 90 and aspect ratio to 1.
        - Render the scene to texture map using Frame Buffer Objects
    3. Repeat step 2 for all 6 faces of the cube map

2. Implement the environment mapping algorithm in the fragment shader **[Grade: 20%]**
    1. Calculate the reflection/refraction of the **view vector** with respect to the fragment normal.
    2. Use the transformed reflection/refraction vector to choose the cube map side and to calculate the texture coordinate. You have already implemented this in cube mapping!
    3. Sample the correct texture map and use the color as the fragment color.
    4. **Note:**
        - **You must NOT use OpenGL reflect/refract function, i.e. write your own.**
        - **You must NOT use built-in OpenGL cube mapping functionality. Do not use GL_TEXTURE_CUBE_MAP. Use 6 separate GL_TEXTURE_2D objects instead.**
    5. Fresnel terms:
        - Split refraction computation into multiple bins based on refraction dispersion proportional to the wavelength of light. Use 3 bins (R, G, B) for three wavelengths of light. (Refer slides for shader details.)

3. Scene setup **[Grade: 30%]**:
    1. Same setup as previous assignment with the following modification:
        - The main object is rendered with environment map on it. The other objects only need to be solid colors.
        - Remove the bottom plane
        - Create sky box:
            - Sky box is always centered at the camera position and does NOT rotate with the camera.
            - Sky box is rendered as cube with its sides facing inward.
            - Render it first with z-buffer turned off (disable both depth test and write) when rendering it.
            - Turn z-buffer back on (enable both depth test and write) after rendering the sky box.
        - Render the 6 side of the cube maps on screen in this order: left (-x), right (+x), bottom (-y), top (+y), back (-z), and front (+z).

- Note: Use GL_CLAMP_TO_EDGE texture parameter to avoid black pixels along the seams of the skybox.

4. In addition to inputs in assignment 3, have user input to cycle between the following **[Grade 20%]**:
   1. **Environment Mapping**
      - Only reflection
      - Only refraction
      - Combination of both using the Fresnel approximation (Schlick's method)
   2. **Phong Shading + Environment Mapping**
      - Combine any of the above options with the object's Final Color as per the Phong Model
      - User provides the interpolation factor to 'mix' (hint ☺) the two colors.
5. User Interactivity **[Grade 5%]**:
   1. Use user input for the object refractive index and assume that the camera is in air (refractive index = 1.0). User values can be in the range [0, 100]. **Very high values of refractive index should show total internal reflection**.
   2. Experiment with the values below to implement different materials
      - Air – 1.000293
      - Hydrogen – 1.000132
      - Water – 1.333
      - Olive Oil – 1.47
      - Ice (solidified water) – 1.31
      - Quartz – 1.46
      - Diamond – 2.42
      - Acrylic / plexiglas / Lucite – 1.49

6. **Miscellaneous Issues [Grade 10% - Zero on assignment if not satisfied]**
   1. Failure to provide a submission that has issues not listed in the rubric above will result in zero for the entire assignment.
   2. This could include (but is not limited to) issues with compilation, building or execution, scene setup, rendering, environment map creation and usage that are not explicitly stated in points 1 through 6 above.

Assignment Submission Guideline

Please refer to the syllabus for assignment submission guideline.

Bonus points (20%)

Implement layered rendering using geometry shaders to output the dynamic environment map directly to 6 render-targets in one pass.

GRADING SHEET

Name of student: _____ **Total points obtained :** _____

Implementation Point	Grade	Points obtained	Comments
Sky box rendering	15%		
Correct rendering of the sky box when generating the environment map.	7		
Correct rendering of the sky box when applying the environment map.	8		
Scene setup & Environment Map generation	30%		
Correct camera setup to render the 6 sides	5		
Correct copy (or render) to texture maps	5		
Correct rendering on environment map on screen	5		
Correct rendering of spheres rotating around the object into the dynamic render targets	5		
Correct rendering of skybox and scene with proper use of z-buffer operations	5		
Correct rendering of textures at the seams of the skybox	5		
Environment Map application in shaders	20%		
Correct loading of the cube map in the shader	5		
Correct calculation of the reflection and refraction vectors	5		
Correct cube side is chosen for sampling	5		
Correct calculation of the texture coordinates for sampling textures	5		
User Interactivity – Rendering	25%		
Reflection only	5		
Refraction only	5		
Fresnel combination of reflection and refraction for 3 wavelengths (R,G,B)	5		
Combination with object emissive color	5		
Change refractive index of object interactively	5		
Miscellaneous issues	10%	Automatic zero on rest of the assignment	
TOTAL	**100%**		