Summer 2020
CS 300 | Advanced Computer Graphics I
Assignment 4 | Reflection and Refraction with Cube Mapping

Description

For this programming assignment you need to implement reflection and refraction system using cube mapping:

1. Cube Map

Like the shadow mapping assignment, this project will also need two different passes in order to implement reflection and refraction. Each pass will require rendering the whole scene from different views:

1. Construct the 6 texture maps for cube mapping algorithm. The 6 textures are to be generated at runtime every frame.
    - 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, +y, -y, +z, -z).
        - Set the camera field of view to 90 and aspect ratio to 1.
        - Render the scene directly to a color texture map via a *framebuffer objects* (**FBOs**).
        - Render the 6 side of the cube maps on screen in this order: left (-x), right (+x), bottom (-y), top (+y), back (-z), front (+z).
        - In order to sample a cube map the input is a 3D coordinate, so wrapping policy needs to be specified for the 3 coordinates (S,T,R).

2. Shader programs

1. **Vertex Shader Cube Map Rendering**
    - Transform the vertex position.
    - Transfer vertex position in model space to fragment shader.

2. **Fragment Shader Cube Map Rendering**
    - Use position in model space to sample the cube map.

3. **Vertex Shader Environment Cube Map Usage**
    - Transform the vertex and the normal to the corresponding space.

4. **Fragment Shader Environment Cube Map Usage**
    - Calculate the reflection/refraction of the view vector (in world space) with respect to the fragment normal (in world space).
    - Use the transformed reflection/refraction vector to choose the cube map side and to calculate the texture coordinate.
    - Sample the environment map and use the color as the fragment color.

3. Scene and Setup

- Create a skybox that is always centered on the camera. Use the cube created on assignment 0 but avoid culling back faces (so we see faces from the inside). Mesh and textures for the sky box will be provided. Disable depth buffer and culling when rendering this object and render it in first place, enable it for every other object.
- Render one shape with size 20x20x20 at a time (from the shape library), arrows keys will rotate that shape.
- Repeat the center shape with two adjacent objects that move up and down along the Y axis. Positions of those two shapes will be (-20,0,0) and (20,0,0) and scale will be 5x5x5. Adjacent objects should be moving in up/down in a sinusoidal manner and they should be reflected on the top and bottom faces of the environment map.
- Adjacent objects should show the color texture implemented on assignment 0.
- Central shape should reflect/refract the environment, including the adjacent objects. Use refractive index 1.33 for the object.
- Lighting/Normal mapping/Shadow mapping is **NOT** required for this assignment and it should be disabled.

4. Input Handling

- Move the camera around in always looking at the object.
  - W: Move up.
  - S: Move down.
  - A: Move left.
  - D: Move right.
  - E: Further from object.
  - Q: Closer to object.

- Select shape to be rendered through the number keys.
  - Numbers 1 to 5: Change the shape to be rendered
    - 1: Plane
    - 2: Cube
    - 3: Cone
    - 4: Cylinder
    - 5: Sphere
  - +: Increase the shape subdivisions
  - -: Decrease the shape subdivisions

- O: Toggle to pause/start adjacent object animation.
- N: Toggle normal rendering
- F: Toggle face/averaged normal
- M: Toggle wireframe mode on/off
- T: Cycle between the different modes (texture/ reflective/ refractive)

- Object rotation for center shape.
  - Arrows Up/Down: Rotate the shape along Y-axis
  - Arrows Right/Left: Rotate the shape along X-axis

Assignment Submission

Please refer to the syllabus for assignment submission guideline. Failure to the submission guidelines correctly might cause you to lose point.

Grading Rubrics

The following is a rough guideline on how your assignment will be graded and the weight of each part.

- Sky box rendering 15%
- Environment map generation 35%
- Environment map application 40%
- Scene and presentation 10%