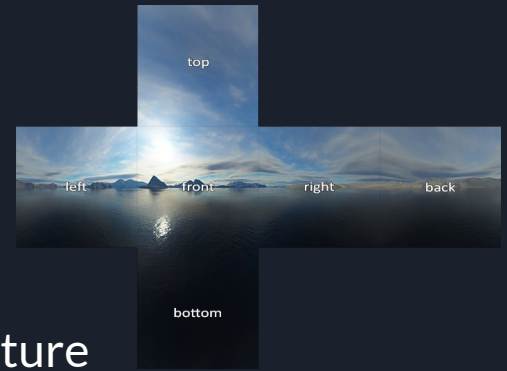




Discussion 6

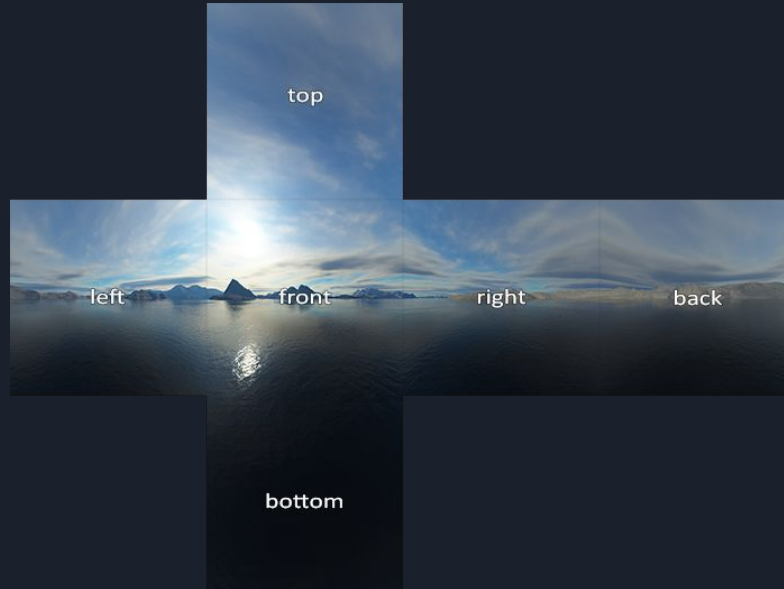
Sky box



- Textures and sky boxes in Part 1: covered in lecture on Nov 5th.
- TODO: A new set of shaders for sky boxes are needed.
- TODO: Cube can be modified and used for skybox.
- Tutorial link:
<https://learnopengl.com/Advanced-OpenGL/Cubemaps>

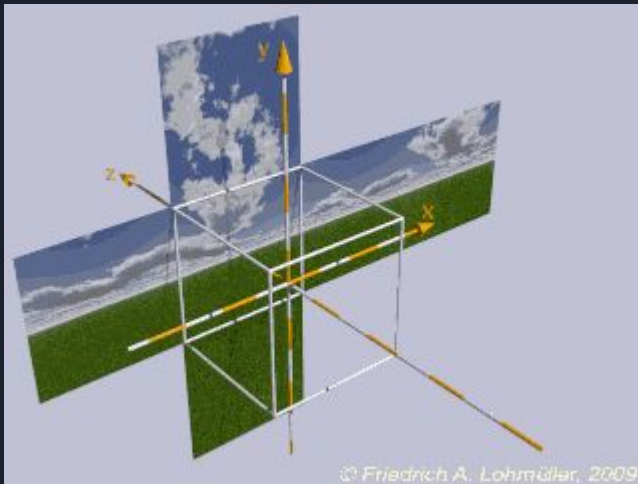
Sky box

- Select your skybox:
 - http://www.f-lohmueller.de/pov_tut/skyboxer/skyboxer_3.htm
 - <http://www.custommapmakers.org/skyboxes.php>
 - Create your own high resolution box textures.
- Make sure the orientations are correct as shown here.



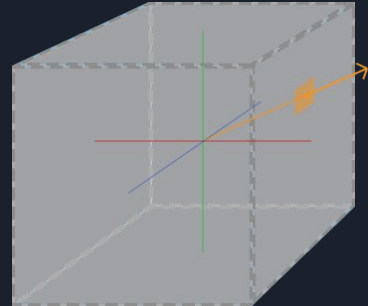
Sky box

- Set up the cube for skybox and the camera to be inside of this cube.



Sky box

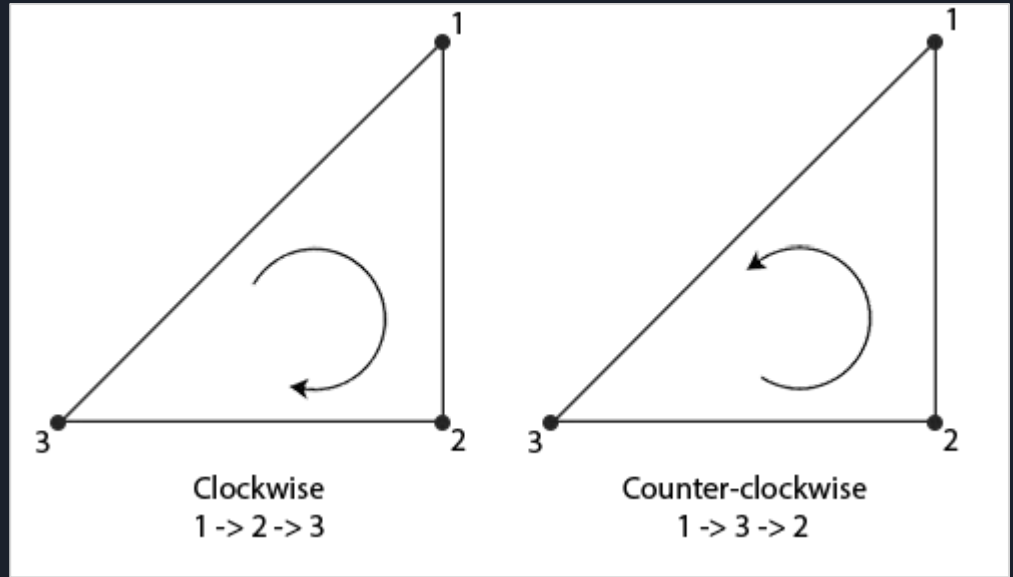
- Coding guide:
 - a. Create a cube object. In Skybox.cpp or Cube.cpp, create **VAO**, VBO and set of vertices just like before.
 - b. Create a simple **shader program** for Skybox,
 - skybox.vert: map input position to texcoords directly.
 - skybox.frag: calculate Fragcolor based on texturecoords using built-in function texture.
 - c. Create a loadCubemap function that setup 6 textures and returns a **texture ID**.
 - d. In the render loop, choose to use the **shader program** from b. , bind vertex array to the **VAO** of skybox from a. , and bind GL_TEXTURE_CUBE_MAP to the **texture ID** created in c.



How to render skybox with front face culling

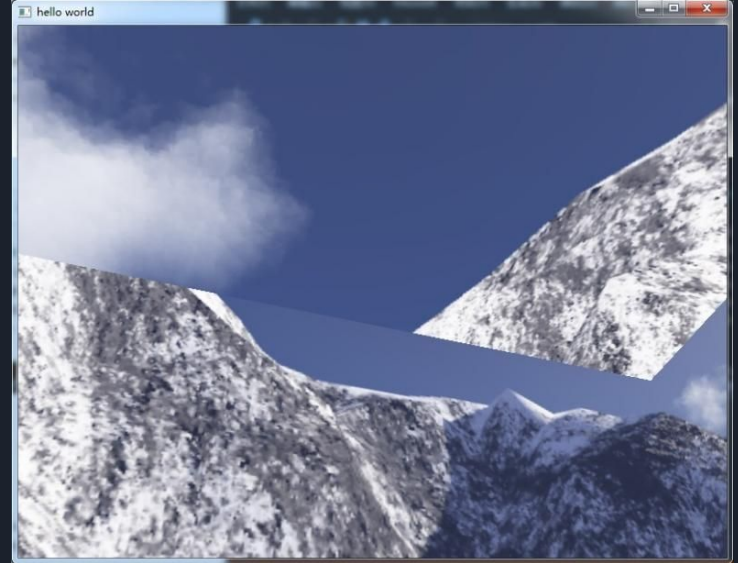
Cube uses counter-clockwise triangles. Here are 2 options to display the inside of the cube as skybox:

1. `glEnable(GL_CULL_FACE);`
`glCullFace(GL_FRONT);`
2. **Telling GL it is defined clockwise:**
`glEnable(GL_CULL_FACE);`
`glCullFace(GL_BACK);`
`glFrontFace(GL_CW);`



Common mistakes

- Texture is in low resolution.
- Visible edge discrepancy.
- Incorrect face culling.



Tutorial: <https://learnopengl.com/Advanced-OpenGL/Face-culling>



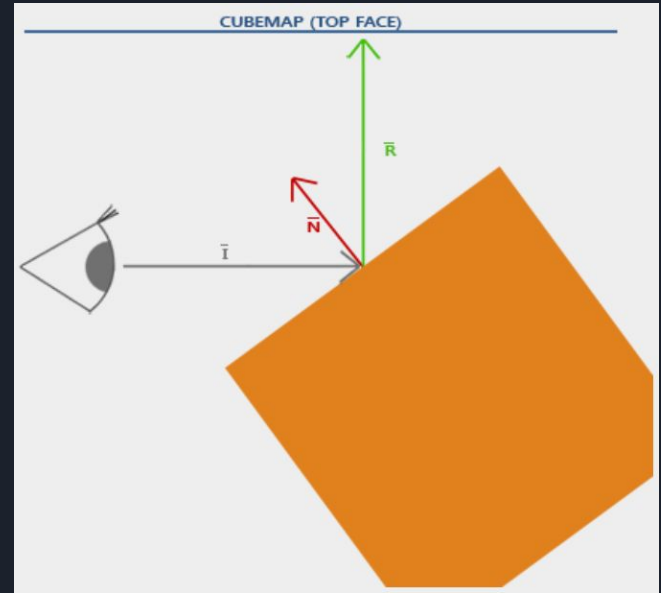
Sphere and Environment Mapping

- Environment mapping in Part 2: covered in lecture on Nov 5th
- Metal ball reflection effect.
- TODO: change the shader.vert and shader.frag to environment mapping. Lighting code is no longer required here.
- Tutorial link:
<https://learnopengl.com/Advanced-OpenGL/Cubemaps>

Sphere and Environment Mapping

- R: reflection vector
- N: normal
- I: view direction
- Calculate reflection vector using GLSL built-in function `reflect()`

```
vec3 I = normalize(Position - cameraPos);  
vec3 R = reflect(I, normalize(Normal));  
FragColor = vec4(texture(skybox, R).rgb, 1.0);
```





Track

- Parametric curves in Parts 3, 4 and 5: covered in lecture on Nov 7th.
- TODO: Add another set of shaders for curves.
- TODO: Add another set of classes for curves.