The Skybox

Skybox

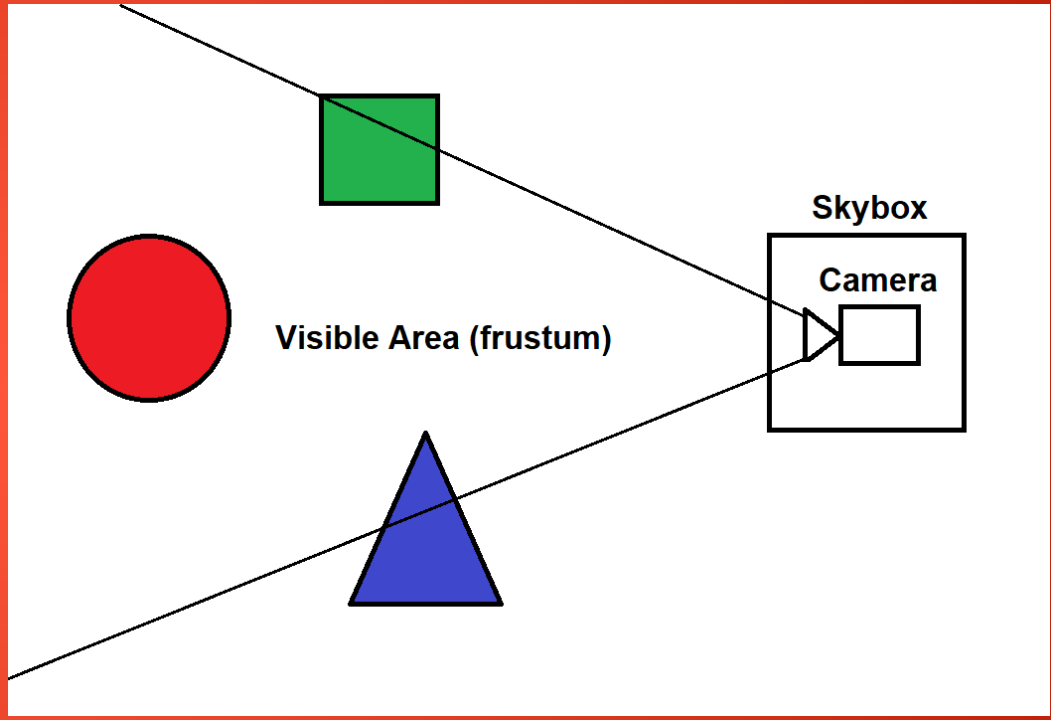
- A cube mesh creating the illusion of a larger world.
- Use cubemaps to texture the cube.
- Requires its own shader to run before the main shader.
- World is drawn “on top of” the skybox.
- Could create a huge skybox around the whole scene…
- But then we’d have to use the same View Matrix far plane everywhere.
- Solution: Disable depthmask.

Skybox - Cubemap

- Creating cubemap works much like before.
- Bind 6 textures to each face, use RGB or RGBA values instead.
- glTexImage2D(GL_TEXTURE_CUBE_MAP_POSITIVE_X + i, 0, GL_RGB, width, height, 0, GL_RGB, GL_UNSIGNED_BYTE, data);
- Use a separate shader for drawing Skybox and pass in a 1x1x1 cube.
- Essentially we are creating a small box around the camera.

Skybox - Cubemap

- Skybox is a small cube around the camera.
- How can we see objects in the scene?
- Disable the depthmask for the skybox render!
- OpenGL treats Skybox depth as being at furthest possible distance…
- So it renders as if BEHIND the objects!



Skybox - Shader

- Pass gl_Position with projection and view matrix, but WITHOUT model matrix: We want the skybox to remain static around the camera.
- TexCoords = aPos;
- Texture Coordinates can use fragment position. Since the skybox does not move, it is effectively at the origin… so any fragment position vector is also its direction from the origin.
- Fragment Shader takes skybox as a samplerCube…
- colour = texture(skybox, TexCoords);
- Remember: samplerCubes use a vector to find a texel, so we use TexCoords vector from Vertex Shader.

Skybox - Drawing

- Drawing the skybox works like any other draw.
  - 1. Initialise shader to use.
  - 2. Bind VAO of skybox cube.
  - 3. Bind texture of skybox.
  - 4. Draw skybox.
- However, another important step is needed.
- Call glDepthMask(GL_FALSE); before drawing to disable the Depth Mask.
- Important: Make sure to re-enable the Depth Mask afterwards!
- glDepthMask(GL_TRUE);

Skybox - Finalising

- When drawing the skybox, need to treat camera as if it is at the origin (where the skybox is treated as being).
- Therefore, View Matrix passed to skybox shader needs to remove any translations it has (rotations are to remain so we can look around our skybox).
- Recall: Translation happens in the 4th column of a matrix…
- So we can just remove the column by converting to a mat3, then convert back to a mat4.
- glm::mat4(glm::mat3(viewMatrix));
- Another important note: If you call glClear before rendering the scene itself… make sure you also do it BEFORE rendering the skybox!
- glClear clears the current Framebuffer. If you clear after the skybox, you will also clear the skybox itself!

Summary

- Skybox uses 6 textures around the camera to create illusion of a larger world.
- A skybox is usually just a small cube around the camera.
- Disabling the depthmask before drawing a skybox enables the scene to be drawn as if in front of the skybox.
- Need to remove translation of View Matrix for Skybox call to keep camera in skybox.
- Make sure to call glClear BEFORE the Skybox, not after it.

See you next video!