OpenGL Notes

Depth testing

The depth buffer is automatically created by the windowing system and stores depth values as 16,24 or 32 bits with the default in most systems being 24 bits.

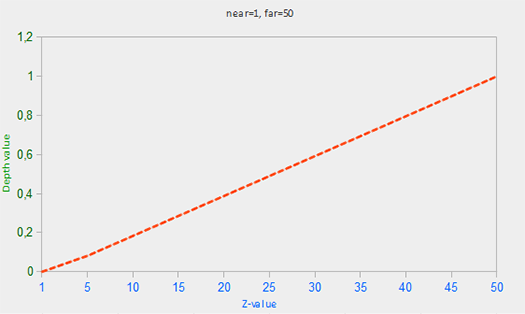
Done in screen space.

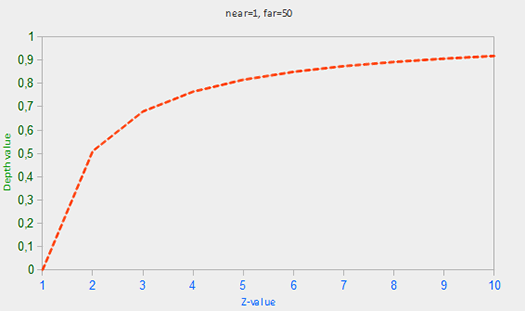


The Screen space coordinates relate directly to the Opengl glViewport function and can be accessed via GLSL’s built-in gl\_FragCoords variable in the fragment shader.

Early depth testing allows the depth test to run before the fragment shader. Due to fragment shading being expensive, so they should be avoided when possible. If a fragment shader would write to its depth value, early depth testing is impossible, OpenGl cant figure out the depth value beforehand.

Default depth value precision is very linear



In practice a linear depth buffer would never be used as it would give enormous precision to objects that are far away. A non Linear depth buffer would be way more suitable

It would give great precision when z is close and much less when its not

Cubemaps

A texture that contains 6 individual 2D textures that each form

one side of a cube.

Because a cubemap consists of 6 textures, one for each face, we have to call glTexImage2D six times with their parameters set to values similar to the previous tutorials. This time however, we have to set the texture *target* parameter to a specific face of the cubemap, basically telling OpenGL which side of the cubemap we're creating a texture for. This means we have to call glTexImage2D once for each face of the cubemap.

Since we have 6 faces OpenGL provides us with 6 special texture targets specifically for targeting a face of the cubemap:

|  |  |
| --- | --- |
| **Texture target** | **Orientation** |
| GL\_TEXTURE\_CUBE\_MAP\_POSITIVE\_X | Right |
| GL\_TEXTURE\_CUBE\_MAP\_NEGATIVE\_X | Left |
| GL\_TEXTURE\_CUBE\_MAP\_POSITIVE\_Y | Top |
| GL\_TEXTURE\_CUBE\_MAP\_NEGATIVE\_Y | Bottom |
| GL\_TEXTURE\_CUBE\_MAP\_POSITIVE\_Z | Back |
| GL\_TEXTURE\_CUBE\_MAP\_NEGATIVE\_Z | Front |