Convention of faces in OpenGL cubemapping

Asked 8 years, 5 months ago Active 8 years, 5 months ago Viewed 5k times

17

10

What is the convention OpenGL follows for cubemaps?

I followed this convention (found on a website) and used the correspondent GLenum to specify the 6 faces `GL_TEXTURE_CUBE_MAP_POSITIVE_X_EXT` but I always get wrong Y, so I have to invert Positive Y with Negative Y face. Why?

```
          ________
         |        |
         | pos y  |
         |        |
 ________|________|_________________
|        |        |        |        |
| neg x  | pos z  |  pos x |  neg z |
|        |        |        |        |
|________|________|________|________|
         |        |
         |        |
         | neg y  |
         |________|
```

opengl 3d geometry glsl

Share Improve this question Follow

asked Jul 27 '12 at 10:04

linello

7,184 11 53 95

2 Answers

Active Oldest Votes

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email Sign up with Google Sign up with GitHub Sign up with Facebook ✕

19

> but I always get wrong Y, so I have to invert Positive Y with Negative Y face. Why?

Ah, yes, this is one of the most odd things about Cube Maps. Rest assured, you're not the only one to fall for it. You see:

Cube Maps have been specified to follow the RenderMan specification (for whatever reason), and RenderMan assumes the images' origin being in the upper left, contrary to the usual OpenGL behaviour of having the image origin in the lower left. That's why things get swapped in the Y direction. It totally breaks with the usual OpenGL semantics and doesn't make sense at all. But now we're stuck with it.

Take note that upper left, vs. lower left are defined in the context of identity transformation from model space to NDC space

Share Improve this answer Follow

answered Jul 27 '12 at 15:06
datenwolf
147k 11 162 267

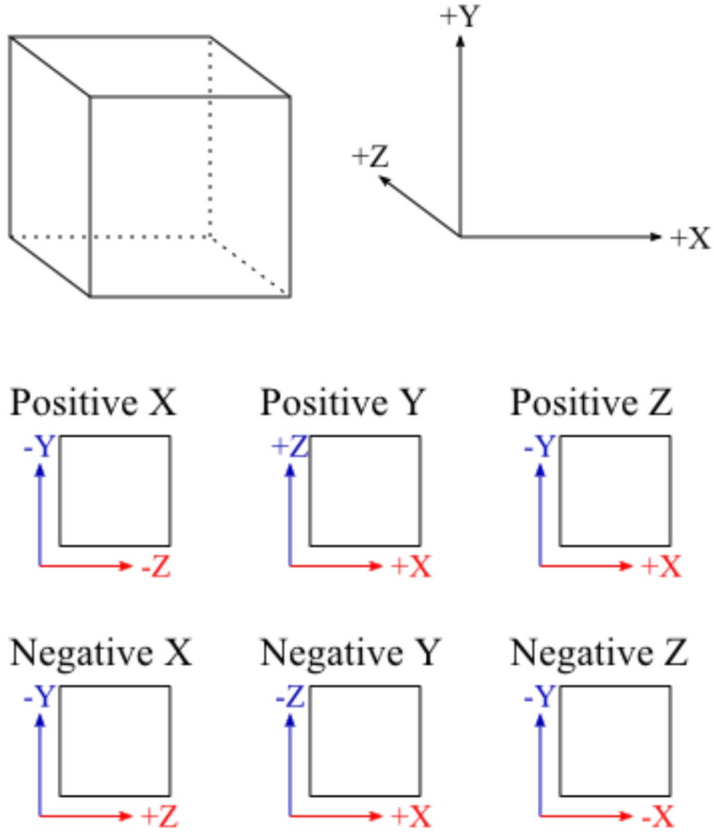
1 Ok, so it's not my fault! This is only a weird behaviour of OpenGL! Thanks! – linello Jul 30 '12 at 9:09

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email Sign up with Google Sign up with GitHub Sign up with Facebook ✕

16

Here is a convenient diagram showing how the axes work in OpenGL cubemaps:



Share Improve this answer Follow

answered Jul 27 '12 at 19:20

Nicol Bolas

364k 48 613 809

This image clarifies everything now! Thanks! – linello Jul 30 '12 at 9:10

1 Doesn't -z in OpenGL normally point forward instead of +z? Is this different for cubemaps? Also, if the bottom left of the Positive X texture is the bottom, furthest, right hand corner of the cube, isn't -Y pointing in the wrong direction compared to your axes? – user673679 Nov 10 '14 at 16:24

"*Doesn't -z in OpenGL normally point forward instead of +z?*" Not in cubemaps. The spec is quite clear on this. "*if the bottom left of the Positive X texture is the bottom, furthest, right hand corner of the cube*" It isn't. The bottom left of the +X face maps to the *top*, right, far corner of the cube. Remember: OpenGL textures tend to be upside-down, due to the way that GL uses a bottom-left coordinate system for its faces. – Nicol Bolas Mar 13 '16 at 14:55

Does the above image show cube map faces from inside (cube map center) or outside ? – Irbis Sep 30 '19 at 8:13

@Irbis: Cubemaps compute the texel from a directional texture coordinate as if the direction were inside of the cube. So the faces represent the inside of the cube, as does the above diagram. – Nicol Bolas Sep 30 '19 at 13:42

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email Sign up with Google Sign up with GitHub Sign up with Facebook