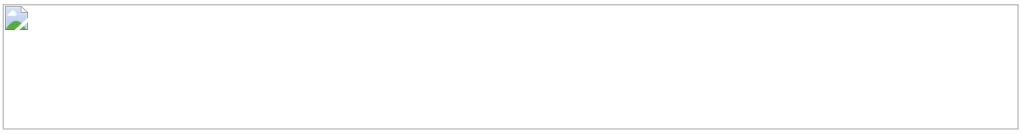
openGL SubTexturing



I have image data and i want to get a sub image of that to use as an opengl texture.

```
glGenTextures(1, &m_name);
glGetIntegerv(GL_TEXTURE_BINDING_2D, &oldName);
glBindTexture(GL_TEXTURE_2D, m_name);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA, m_width, m_height, 0, GL_RGBA, GL_UNSIGNED_BYTE,
m_data);
```

How can i get a sub image of that image loaded as a texture. I think it has something to do with using glTexSubImage2D, but i have no clue how to use it to create a new texture that i can load. Calling:

```
glTexSubImage2D(GL_TEXTURE_2D, 0, xOffset, yOffset, xWidth, yHeight, GL_RGBA,
GL_UNSIGNED_BYTE, m_data);
```

doe nothing that i can see, and calling glCopyTexSubImage2D just takes part of my framebuffer. Thanks

c++ opengl

asked Oct 15 '08 at 16:56

DavidG
1,013 1 15 26

1 Answer

Edit: Use glPixelStorei. You use it to set `GL_UNPACK_ROW_LENGTH` to the width (in pixels) of the entire image. Then you call glTexImage2D (or whatever), passing it a pointer to the first pixel of the subimage and the width and height of the subimage.

Don't forget to restore `GL_UNPACK_ROW_LENGTH` to 0 when you're finished with it.

Ie:

```
glPixelStorei( GL_UNPACK_ROW_LENGTH, img_width );
char *subimg = (char*)m_data + (sub_x + sub_y*img_width)*4;
glTexImage2D( GL_TEXTURE_2D, 0, GL_RGBA, sub_width, sub_height, 0, GL_RGBA,
GL_UNSIGNED_BYTE, subimg );
glPixelStorei( GL_UNPACK_ROW_LENGTH, 0 );
```

Or, if you're allergic to pointer maths:

```
glPixelStorei( GL_UNPACK_ROW_LENGTH, img_width );
glPixelStorei( GL_UNPACK_SKIP_PIXELS, sub_x );
glPixelStorei( GL_UNPACK_SKIP_ROWS, sub_y );

glTexImage2D( GL_TEXTURE_2D, 0, GL_RGBA, sub_width, sub_height, 0, GL_RGBA,
GL_UNSIGNED_BYTE, m_data );

glPixelStorei( GL_UNPACK_ROW_LENGTH, 0 );
glPixelStorei( GL_UNPACK_SKIP_PIXELS, 0 );
glPixelStorei( GL_UNPACK_SKIP_ROWS, 0 );
```

Edit2: For the sake of completeness, I should point out that if you're using OpenGL-ES then you don't get `GL_UNPACK_ROW_LENGTH`. In which case, you could either (a) extract the subimage into a new buffer yourself, or (b)...

```
glTexImage2D( GL_TEXTURE_2D, 0, GL_RGBA, sub_width, sub_height, 0, GL_RGBA,
GL_UNSIGNED_BYTES, NULL );

for( int y = 0; y < sub_height; y++ )
{
    char *row = m_data + ((y + sub_y)*img_width + sub_x) * 4;
    glTexSubImage2D( GL_TEXTURE_2D, 0, 0, y, sub_width, 1, GL_RGBA, GL_UNSIGNED_BYTE, row
);
}
```

edited Oct 15 '08 at 18:56 answered Oct 15 '08 at 17:05

Mike F

Basically i have an image (as raw data), and i want to use part of that image as a texture. I know how to load the entire image as a texture, but don't know how to use just a bit of it. – DavidG Oct 15 '08 at 17:37

Poster answered this. If you have an image in memory, it is stored somehow. If you used glTexImage2D(...GL_RGBA, GL_UNSIGNED_BYTE) to load, then there's a byte for R, G, B and alpha value. The images are stored with pixel (0, 0) starts at position 0. Pixel x at row y starts at image[width*y+x][0] – mstrobl Oct 15 '08 at 18:15

since i am using openGL ES, i ended up doing the first option u gave, that is extracting subimage into a new buffer. sadly on the device it no longer works, so i'm going to rework the texture so that i don't need to solve the problem. – DavidG Oct 15 '08 at 19:01