Subject: Re: UppGL
Posted by nlneilson on Sat, 21 May 2011 08:11:11 GMT
View Forum Message <> Reply to Message

Once the lowest resolution tile size (lztd) has been chosen subsequent level sizes can be found by:

n = LOD; // from 0 to 17 with lztd = 180 deg
lztd DIVIDED BY 2**n // maybe your reference to pow2 ???

0 : 180.0
1 : 90.0
2 : 45.0
3 : 22.5
4 : 11.25
5 : 5.625
6 : 2.8125
7 : 1.40625
8 : 0.703125
9 : 0.3515625
10 : 0.17578125
11 : 0.087890625
12 : 0.0439453125
13 : 0.02197265625
14 : 0.010986328125
15 : 0.0054931640625
16 : 0.00274658203125
17 : 0.001373291015625

with an lztd of 1.0 which I chose for the TAC charts
0 : 1.0
1 : 0.5
2 : 0.25
3 : 0.125
4 : 0.0625
5 : 0.03125
6 : 0.015625
7 : 0.0078125
8 : 0.00390625
9 : 0.001953125
Note that with an lztd of 180.0 and the levels shown the number of decimal places is reaching the limits for a double.
With an lztd of 1.0 for the TAC charts only 4 levels (0 to 3) were needed to show the grain of the paper the FAA (NACO) scanned the images from.

NASA's WWJ cache structure is like this:

...\Earth\NASA\Marble\3\30\30_15.dds
the actual file name, 30_15.dds, the first integer relates to latitude and the second longitude.

I wrote a python app that takes two arguments, the lztd and file name (including the path) which gives the location (lat and lon) of the bottom left of that tile (file) with the reference at -90 lat and -180 lon.

The same concept is used for high resolution images like X-Rays.
In that case the lowest level of resolution would just be the integer 1 for level 0 (zero) or a size in mm, inch or ?? if scaled.

Maybe I can help if UppGL gets into tiling images.