CPSC 314 Homework 5

Term: Jan 2009, Instructor: Wolfgang Heidrich, heidrich@cs.ubc.ca, http://www.ugrad.cs.ubc.ca/~cs314

This problem sheet deals with the depth buffer and blending. Solutions will be discussed in the labs in the week of March 2-6.

1 Depth Buffer Resolution

As we have seen in the discussion of perspective matrices, the z mapping of the standard OpenGL perspective matrix is (after perspective division):

$$z' = \frac{f+n}{f-n} + \frac{2fn}{f-n} \cdot \frac{1}{z}.$$

a) confirm that this formula maps points at $z = -n$ (the near plane) to a depth value of -1, and points at $z = -f$ (the far plane) to 1.

$\frac{f+n}{f-n} + \frac{2fn}{f-n} \cdot \frac{1}{-n} = \frac{f+n-2f}{f-n} = -1$

$\frac{f+n}{f-n} + \frac{2fn}{f-n} \cdot \frac{1}{-f} = \frac{f+n-2n}{f-n} = 1$

b) for storing the depth value into a z-buffer, the depth z' after the perspective divide is linearly mapped to the integer range $0 \ldots 2^N - 1$, where N is the number of bits in the depth buffer. Derive the formula for this mapping.

$z'' = a z' + b$ with $a(-1) + b = 0 \Rightarrow a = b$

$a + b = 2^N - 1$

$\Rightarrow a = b = \frac{2^N - 1}{2}$

c) Assume a 16 bit z-buffer, a near plane at z=10cm, and a far plane at z=1km. What is the world-space depth value $z_{\Delta n}$ that results in a z-buffer value of 1? What is the world-space depth value $z_{\Delta f}$ that maps to a z-buffer value of $65534 (= 2^{16} - 2)$? Note that $z_{\Delta n} - n$ and $f - z_{\Delta f}$ are indicators of the numerical precision of the z-buffer close to the near and the far plane, respectively.

See next page

First: determine combined mapping:

$$z'' = \frac{2^N-1}{z}\left(\frac{f+n}{f-n} + \frac{2fn}{f-n}\cdot\frac{1}{z}\right) + \frac{2^N-1}{2}$$

$$= (2^N-1)\frac{fn}{f-n}\cdot\frac{1}{z} + \frac{2^N-1}{2}\left(\underbrace{\frac{f+n}{f-n}+1}_{=\frac{2f}{f-n}}\right)$$

$$= (2^N-1)\frac{fn}{f-n}\cdot\frac{1}{z} + (2^N-1)\frac{f}{f-n}$$

(easy to verify: $z=-n \Rightarrow z''=0$, $z=-f \Rightarrow z''=2^N-1$)

now: invert this mapping (solve for z):

$$z = \frac{(2^N-1)\frac{fn}{f-n}}{z'' - (2^N-1)\frac{f}{f-n}}$$

finally: plug in numbers (I work in units of meters):

$$z_{\Delta n} = \frac{65535\cdot\frac{1000\cdot 0.1}{1000-0.1}}{1-65535\cdot\frac{1000}{1000-0.1}} = -0.100001\,52$$

i.e. next to the near plane we have a resolution of 1 µm!!

$$z_{\Delta f} = \frac{65535\cdot\frac{1000\cdot 0.1}{1000-0.1}}{65534-65535\cdot\frac{1000}{1000-0.1}} = -867.6$$

i.e. next to the far plane, the resolution is only less than 100 m!!

2 Blending

Assume you want to model objects made out of colored glass. Assume we have a red glass that leaves through 75% of the red light, but only 25% of each of the green and blue components, and a blue glass that transmits 80% of blue and 10% each of red and green. Assume neither glass reflects light on its own.

a) write down the blending equations that describe the red glass.

$dst_{red} = 0.25 \cdot dst_{red} = src_{red} \cdot dst_{red}$

$dst_{green} = 0.25 \cdot dst_{green}$

$dst_{blue} = 0.25 \cdot dst_{blue}$

b) what are the OpenGL blending function calls are required to specify this material? What RGBA value do you need to assign to the vertices of this glass object?

glBlendFunc(GL_ZERO, GL_SRC_COLOR)

or: glBlendFunc(GL_DST_COLOR, GL_ZERO)

RGBA = (0.75, 0.25, 0.25, ?) (alpha doesn't matter here)

c) show that the blending equation from part a) is *order independent*. That is, if you have objects, one with the red material and one with the blue material, and both objects partially occlude each other, it does not matter in which order you draw the objects.

Channels are just multiplied. Multiplication commutes.

d) the property from part c) does not hold for general blending equations. Show a counterexample, and demonstrate how different orders can result in different color values.

See "over" operator discussed in class