Graphics algorithms

Perspective matrix for horizontal fov $f \in (0, \pi)$, aspect ratio $a = \frac{w}{h}$, near plane $n > 0$ (assuming x is right, y is up, z is forward) is given by

$$\begin{bmatrix} \frac{1}{\tan(f/2)} & 0 & 0 & 0 \\ 0 & \frac{a}{\tan(f/2)} & 0 & 0 \\ 0 & 0 & 0 & n \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

describing the function mapping a view space coordinate (x, y, z) to clip space:

$$(x, y, z, 1) \mapsto \left(\frac{x}{\tan(f/2)}, \frac{ay}{\tan(f/2)}, n, z\right)$$

According to rasterizer desc, in clip space, we clip to $-w \leq x \leq w, -w \leq y \leq w, 0 \leq z \leq w, w > 0$. In terms of view-space coordinates, this is equivalent to clipping to

$$-z \tan(f/2) \leq x \leq z \tan(f/2)$$

$$-\frac{z \tan(f/2)}{a} \leq y \leq \frac{z \tan(f/2)}{a}$$

$$0 \leq n \leq z$$

$$z > 0 \text{ (although this is already implied since n } > 0)$$

Why can we clip in clip space? Because clipping is finding intersections with clipping planes using lines in clip space which correspond to lines in view space: let p, q be two points in view space (padded with 1), and let t make $(1-t)(Pp) + t(Pq)$ satisfy the the clip space inequalities. This is equal to $P((1-t)p + tq)$

Depth clip simply disables view $0 \leq n \leq z$, but $z > 0$ is still enforced. Note that if depth clip is disabled, then depth is **clamped to the viewport** before any depth testing takes place (see docs).

NDC is then given (in terms of view space) by

$$\left(\frac{x}{z \tan(f/2)}, \frac{ay}{z \tan(f/2)}, \frac{n}{z}\right)$$

Now let $p_1, p_2, p_3 \in \mathbb{R}^3$ be the vertices of a triangle in view space. Let $p = ap_1 + bp_2 + cp_3$ where $a, b, c \geq 0, a + b + c = 1$ be a point inside the triangle. Let f_1, f_2, f_3 be attribute values at the vertices.

Evidently, the interpolated attribute value at p is given by $F = af_1 + bf_2 + cf_3$. But this is also equal to the value given by the rasterization formula (check rasterization.ipynb and the vulkan spec.

Orthographic projection

Now, let $a, b, c \in \mathbb{R}$. Ortho projection is given by

$$\begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

describing the function

$$(x, y, z, 1) \mapsto (ax, by, cz, 1)$$

so in view space, we clip to

$$-1 \leq ax \leq 1$$

$$-1 \leq by \leq 1$$

$$0 \leq cz \leq 1 \text{ (but this one can be disabled with no depth clip)}$$

Screen space ambient occlusion

The technique is to use the depth buffer as an approximation: if a point is inside geometry (note: watertight meshes must be used, i.e. ones with a clear inside and outside), then depth of that point is > final depth value (nearest geometry), but the converse is not true.

To compute the number of additional pixels needed to ensure proper screen edge rendering, divide the ssao radius by the side length of a single pixel at the near plane, a.k.a. $\frac{n \tan(f/2)}{\text{screen width }/2}$ (take ceiling of that).

Deferred rendering

you should store the final color instead of uvs so that the correct mip level is selected.

casting and conversion: https://learn.microsoft.com/en-us/windows/win32/direct3d9/casting-and-conversion