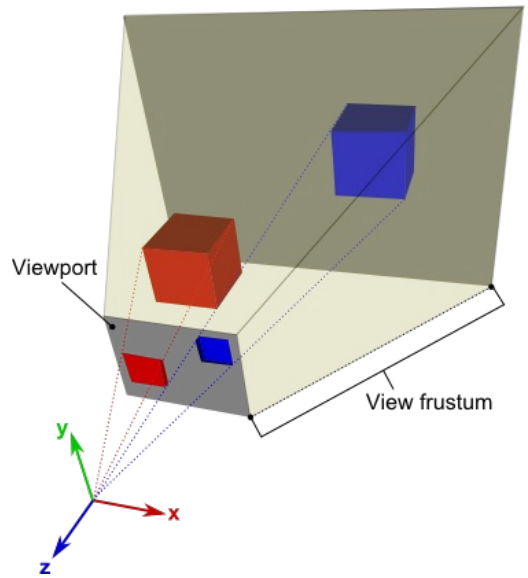
Camera



The camera frustum:

Specified in world coordinates.

E : Eye position

Q : Orientation as a quaternion (type glm::quat)
(encodes both view direction and up vector)
represents the rotation of the X,Y,Z axes
to the world-space view

r_y : frustum's Y versus Z slope

Camera frame in world coordinates

Given **W,H**, the screen's width and height calculate

r_x = r_y * W/H // Beware of unwanted integer division.

Calculate world space vectors **X**, **Y**, and **Z** which align with the screen in world space

vec3 X = r_x * transformVector(Q,Xaxis());
vec3 Y = r_y * transformVector(Q,Yaxis());
vec3 Z = transformVector(Q,Zaxis());

Generate rays for every pixel on screen

(Uses **X**, **Y**, and **Z** from the previous paragraph.)

Screen is **W**, **H** pixels in width and height

for y from 0 to H-1: //Parallelize this loop with an OpenMP #pragma

for x from 0 to W-1:

$d_x = 2(x+1/2)/W - 1$

$d_y = 2(y+1/2)/H - 1$ **// Center of pixel in [-1..1] coordinate system**

Create ray with

origin E, and

direction vector $d_x X + d_y Y - Z$ // Normalize to unit length

Looking ahead to projects 2 and 5

With naive anti-aliasing

Add some randomness (ξ_1 , and ξ_2) to the Ray's position within a pixel

$$d_x = 2(x+\xi_1)/W - 1$$
$$d_y = 2(y+\xi_2)/H - 1$$

This is useful later in the semester when you trace and average multiple rays for every screen pixel.

Depth-of-field (An easy feature for project 5.)

Specify the distance to the in-focus plane: f (in real world coordinate units)

Relative width of disk around eye to spread ray starts: w (0.3=lots, 0.1=little)

Do the pixel calculation in the focal plane (instead of the one-unit-out plane)

$$\hat{P} = E + f\,d_x X + f\,d_y Y + f\,Z$$

Adjust the starting point in a disk around the eye similarly:

$r = w\sqrt{(\xi_1)}$ (Sqrt causes uniform distribution across the disk.)

$$\theta = 2\pi\,\xi_2$$
$$(r_x, r_y) = (r\,\cos\theta, r\,\sin\theta)$$
$$\hat{E} = E + r_x X + r_y Y$$

Use ray

starting at: $E + r_x X + r_y Y$

in direction $\hat{P} - \hat{E} = (f\,d_x - r_x)X + (f\,d_y - r_y)Y + f\,Z$