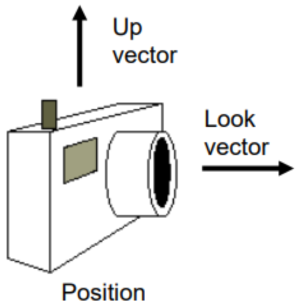
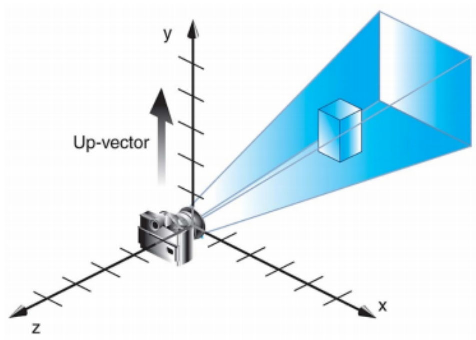
1. Consider a 3D space using right-handed coordinates. A camera is located at $\mathbf{p}$ and looking at a point $\mathbf{p} + \mathbf{d}$ ($\mathbf{d}$ is a unit vector), with its reference up vector being $\mathbf{u}$. (notice that the reference up vector is not true up vector.)

   (a) Write down the unit vectors $\vec{x}$, $\vec{y}$ and $\vec{z}$ of the camera coordinate axes in the world space. Note that we are using conventions from *OpenGL*: $x+$ is pointing to the right; $y+$ is pointing to the up, and $z+$ is pointing to the inverse view direction.

   (b) Write down the view transformation matrix for this camera, which transforms a point from the world space to the camera's local space.

   You can write the answer as cross product of vectors and product of matrices





Answer:

(a) The unit vector dir $\vec{d}$ is equal to -$\vec{k}'$,so $\vec{k}' = -\vec{d}$
the $\vec{i}' = \vec{d} \times \vec{u}$, the $\vec{j}' = (\vec{d} \times \vec{u}) \times \vec{d}$

(b) The view matrix is calculated using glm::lookAt. This is because moving an object in the scene is the same as applying the inverse movement to the camera. To create the view matrix, we need to apply the inverse of both the camera's rotation and translation matrices. This way, if the camera moves or rotates, it will look like the objects in the scene are moving in the opposite direction.

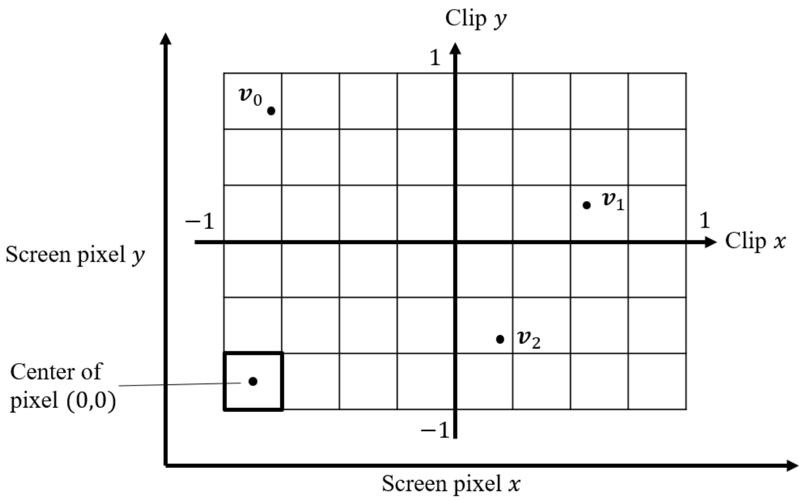
$$\begin{pmatrix} i'_x & i'_y & i'_z & 0 \\ j'_x & j'_y & j'_z & 0 \\ k'_x & k'_y & k'_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -p_x \\ 0 & 1 & 0 & -p_y \\ 0 & 0 & 1 & -p_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

2. Consider the rasterization of a triangle. After applying the model-view-projection matrix, the triangle vertices are $\boldsymbol{v}_0\,(-0.8, 0.8)$, $\boldsymbol{v}_1\,(0.6, 0.2)$ and $\boldsymbol{v}_2\,(0.2, -0.6)$ in clipping space (where x and y has a normalized range [-1, 1]).

   Suppose the screen has a resolution of 8 (horizontal) by 6 (vertical) pixels. Pixel $(0, 0)$ locates at the bottom-left corner of the screen.

   Decide whether the center of pixel $(3, 2)$ **lies within the given triangle**, that is, whether this pixel will be shaded for the triangle. **Write down your calculation process.**

   Hint: The sketch below is inaccurate and is only used for reference.



Answer:

Center of pixel (3, 2) in frustum space: (-1/8, -1/6).

Equation of the line between (-0.8, 0.8) and (0.2, -0.6): $-1.4x - y - 0.32 = 0$

Substitute in (0, 0) yield -0.32

Substitute in (-1/8, -1/6) yield 0.02167...

Obviously (0, 0) is within the triangle range

Thus the pixel center is not in the triangle range.