VECTORS:

$e = [e_0, e_1, e_2]$: Vector which represents the eye (where the camera is located).

$t = [t_0, t_1, t_2]$: Vector which defines the direction in which the eye is looking and the length between the eye and the projection plane.

$p = [p_0, p_1, p_2]$: Vector which represents the position of a point in space.

THE PROJECTION PLANE:

It is a plane which is located at the position $e + t$ and its normal vector is t. Its general form can be written as:

$$t_0(x - (e_0 + t_0)) + t_1(y - (e_1 + t_1)) + t_2(z - (e_2 + t_2)) = 0$$

THE EYE-POINT LINE:

In real life light travels linearly as it gets reflected on surfaces. In this case, the light reflected from the point towards the eye can be represented with a line, and it will get projected into the projection plane. It’s vectorial form can be written as:

$$e + a(p - e)$$

Where a is a random scalar.

Projecting a point into the projection plane means finding which a coefficient makes the resulting vector land on the plane. To do this, we have to solve a system of equations using the parametric form of this line:

$$x + 0y + 0z - (p0 - e0)a = e0$$
$$0x + y + 0z - (p1 - e1)a = e1$$
$$0x + 0y + z - (p2 - e2)a = e2$$

EXAMPLE:

$e = [2, 2, 2]$
$t = [-0.42, -0.57, 0.71]$
$p = [5, -3, 7]$

```
octave:6>
eq = [t(1), t(2), t(3), 0, ((e(1)+t(1))*t(1) + (e(2)+t(2))*t(2) + (e(3)+t(3))*t(3)); 1, 0, 0,
(-(p(1) - e(1))), e(1); 0, 1, 0, (-(p(2) - e(2))), e(2); 0, 0, 1, (-(p(3) - e(3))), e(3)]
eq =

  -0.4200  -0.5700   0.7100        0   0.4454
   1.0000        0        0  -3.0000   2.0000
        0   1.0000        0   5.0000   2.0000
        0        0   1.0000  -5.0000   2.0000

octave:7> res = rref(eq)
res =

   1.0000        0        0        0   2.5868
        0   1.0000        0        0   1.0220
        0        0   1.0000        0   2.9780
        0        0        0   1.0000   0.1956
```

The solution is the vector $p' = [2.5868,\ 1.0220,\ 2.9780]$ in which $a = 0.1956$.

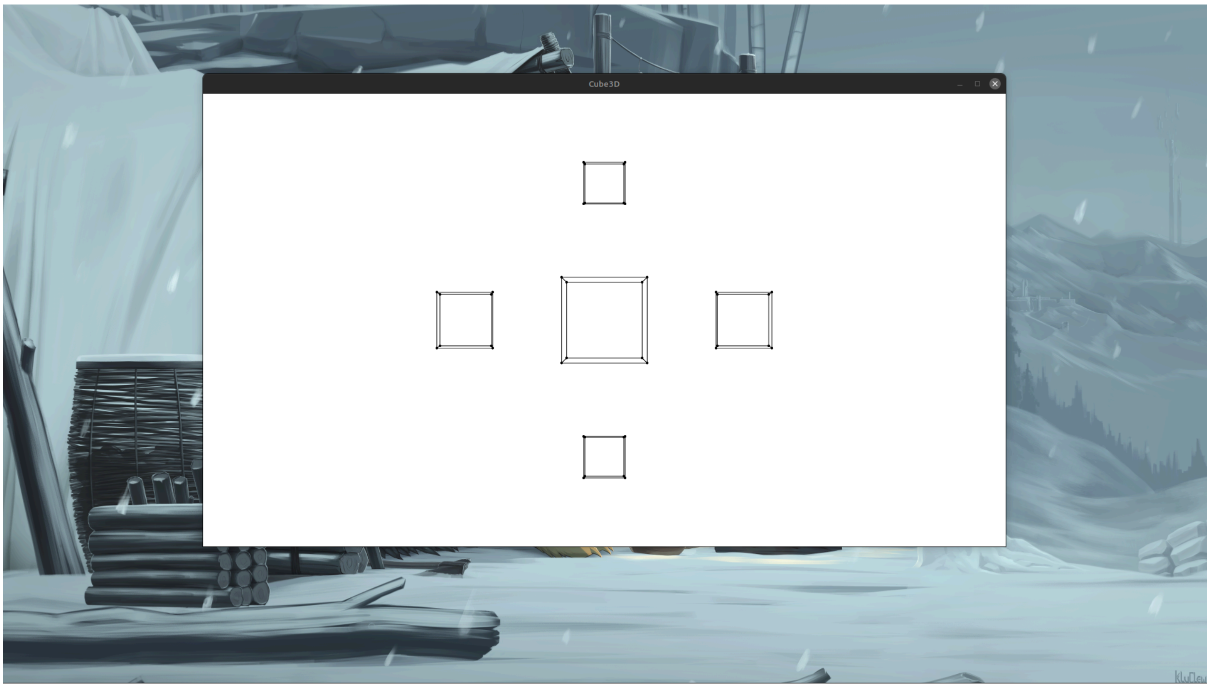
SHOWING IT ON SCREEN:

To show the view plane on screen in a way that the direction you point at is the center of the screen, it is needed to displace the screen view. For example:

- If we are projecting a scene into a screen defined with coordinates x and y, then we have to move the screen view relative to where the vector $[0,\ 0,\ 1]$ lands on the view plane (let's call this projection v) and where the camera is currently looking at.

- In other words, we'd have to find the vector $v - (e + t)$ and move the screen view according to its x and y components.

RESULTS:



CONCLUSION:

This method is kinda tricky and it might be computationally inefficient since you have to solve a system of 4 equations for every point in the scene. It’s easier to just move the scene and let the view stay in place.