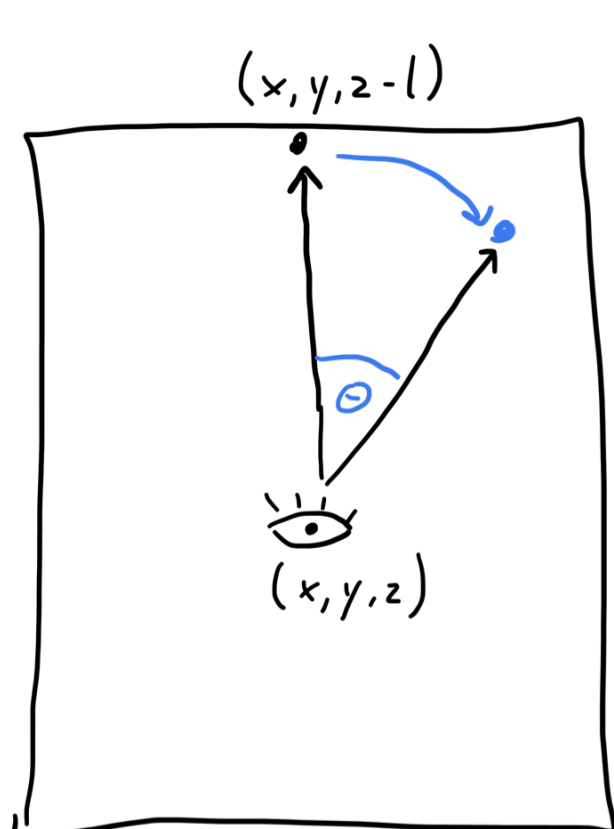


Rotating lookAt



setlookat($\underbrace{x, y, z}_{\text{eye pos}}, \underbrace{a, b, c}_{\text{lookat Pos}}, \underbrace{0, 1, 0}_{\text{up direction}}$)

$$a = \text{Math.cos}(\theta) + x$$

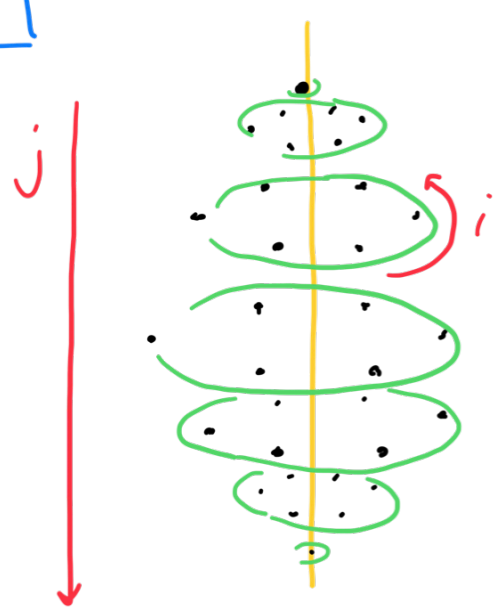
$$b = 0$$

$$c = \text{Math.sin}(\theta) + z$$

Offset by eye position

Global Var adjusted with arrow keys

Sphere



Coordinate Spaces

gl-Position =

= a-Position

Object / Local space - coordinates relative to the object itself

= u_ModelMatrix * a-Position

World Space - coordinates after placing the object in the scene

= u_ViewMatrix * u_ModelMatrix * a-Position

View / Camera space - coordinates relative to the camera

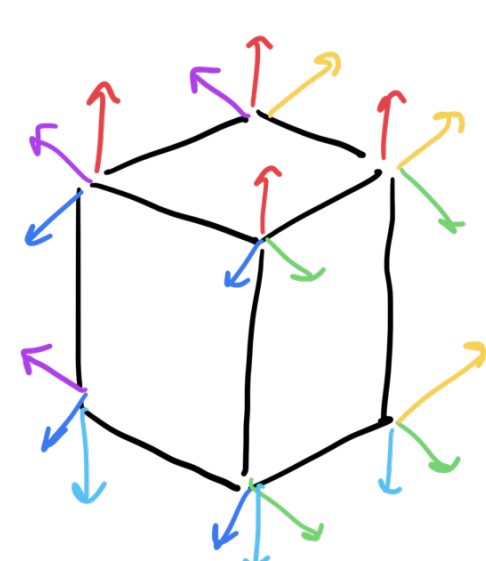
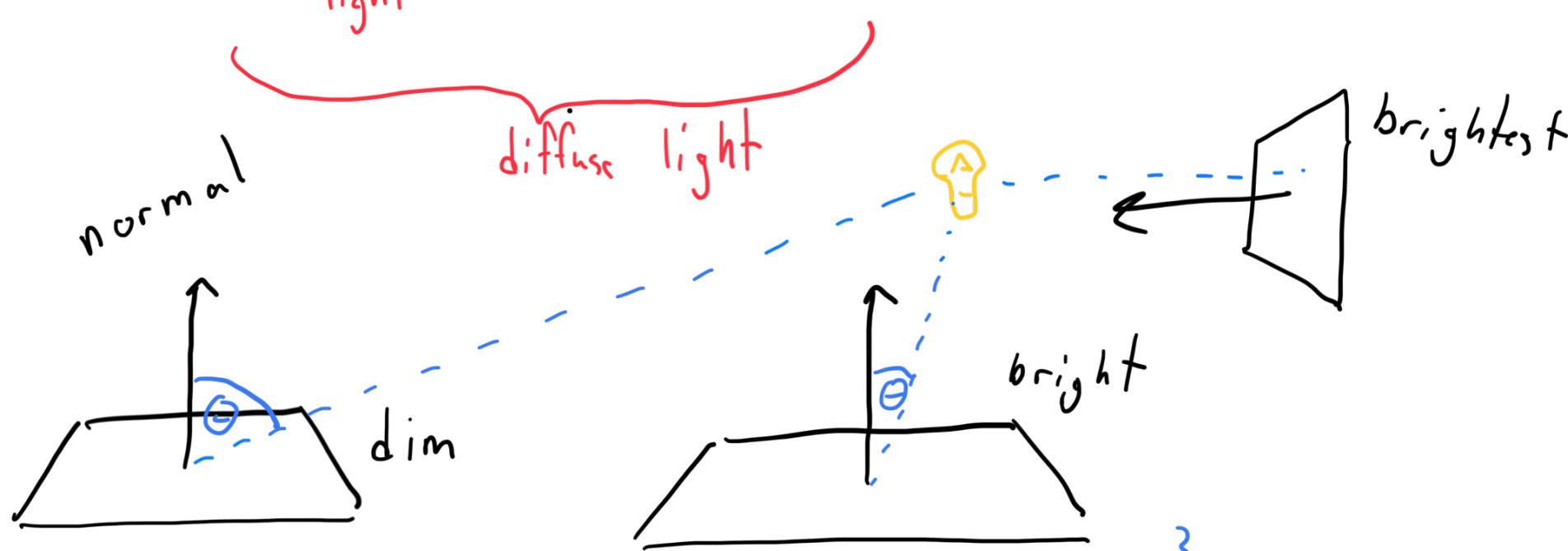
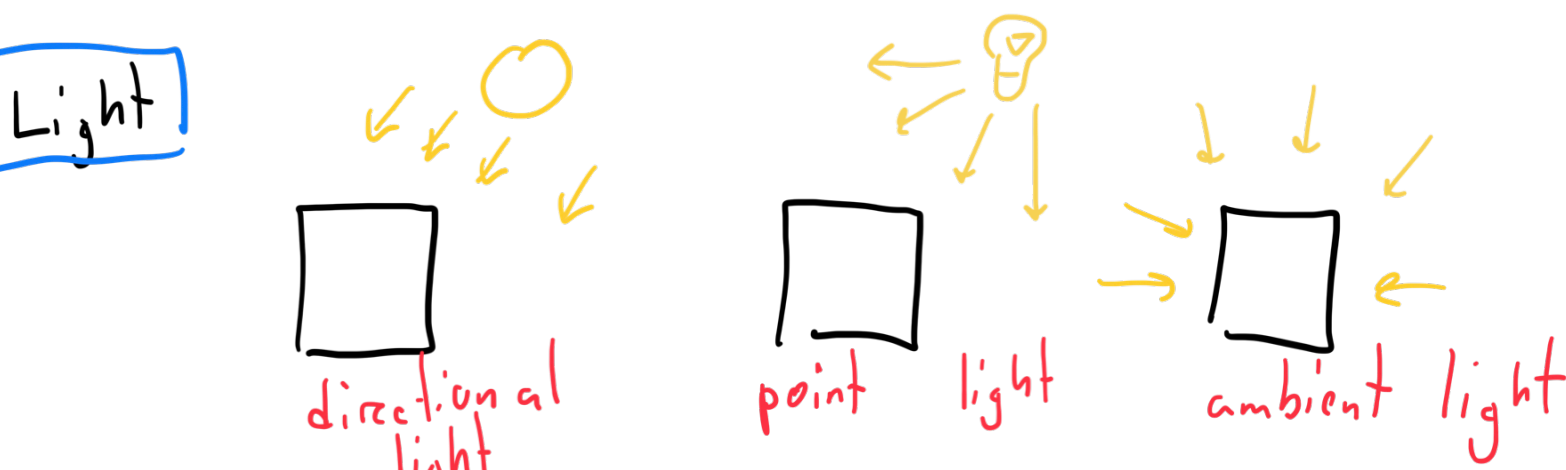
= u_ProjMatrix * u_ViewMatrix * u_ModelMatrix * a-Position

Clip space : coordinates after projection

(Frag shader) v-Position

Screen space : actual pixel coordinate on the screen

Light

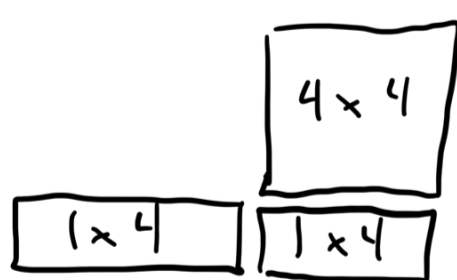


$$\langle \text{surface color by diffuse} \rangle = \langle \text{light color} \rangle \times \langle \text{color of surface} \rangle \times \cos \theta$$

vec3 v_Color.rgb float

($\langle \text{light direction} \rangle \cdot \langle \text{orientation of surface} \rangle$)
dot product

matrix multiplication



dot product

$a = \text{lightDir}$ (1x3) $b = v_Normal$ (3x1)

scalar

$$|a| \cdot |b| \cdot \cos \theta$$

length of a
 $(\sqrt{a_x^2 + a_y^2 + a_z^2})$

To make $|a| = |b| = 1$, we normalize! a & b first!

