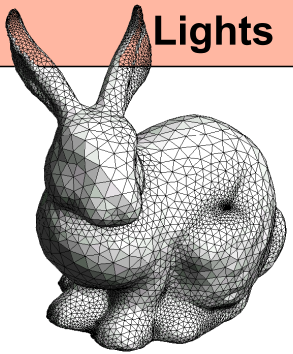
Rendering

Scene Description

3D objects
Coordinate Frame
Camera(s)
Materials
Lights

?

2D Image



OpenGL Rendering Pipeline

(with some details abstracted away)

CPU

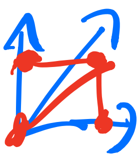
GPU

Javascript, three.js

v_1 v_2 v_3 ...

$\begin{bmatrix}0\\0\\0\end{bmatrix} \begin{bmatrix}1\\0\\0\end{bmatrix} \ldots$ → **vertex shader** (GLSL program) → vertex device coordinates → **rasterization** ("fixed function") → **fragment shader** (GLSL program) → pixel colour → **Framebuffer**

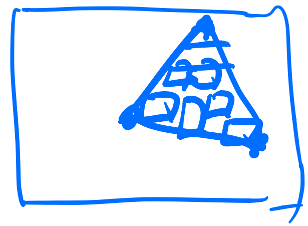
vertex buffer



translating, rotating, scales, projection

$v' = Mv$

for each pixel in triangle
- interpolate attributes
- call fragment shader



lighting

$I = \vec{N} \cdot \vec{L}$

Linear Algebra Review

vectors

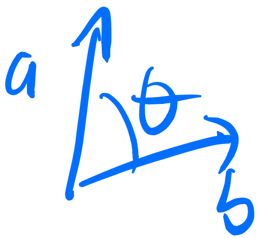
$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \quad \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

column vectors by default.

$$a^T = [a_1 \; a_2 \; a_3]$$

dot product

$$a \cdot b = a^T b = [a_1 \; a_2 \; a_3] \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$



$$= a_1 \cdot b_1 + a_2 b_2 + a_3 b_3$$

$$= |a| |b| \cos(\theta)$$

max when aligned

$= 0$ when $a \perp b$

min when opp directions

Math Review

matrix-vector multiplication

$v' = Mr$

(a) as dot products with the rows

$$\begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} \begin{bmatrix} v \end{bmatrix} = \begin{bmatrix} v \cdot r_1 \\ v \cdot r_2 \\ v \cdot r_3 \end{bmatrix}$$

(b) as weighted combinations of the columns

$$\begin{bmatrix} c_1 & c_2 & c_3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = v_1 \begin{bmatrix} c_1 \end{bmatrix} + v_2 \begin{bmatrix} c_2 \end{bmatrix} + v_3 \begin{bmatrix} c_3 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \underbrace{\begin{bmatrix} i & j & k \end{bmatrix}}_{} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

world coords

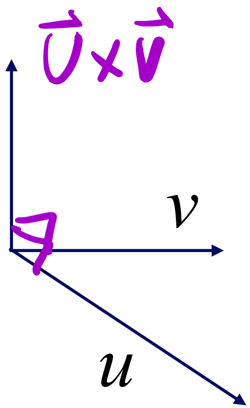
local coords

in world

(local basis vectors, expressed ... coords

Math Review

Cross Product



usually use these.

Right Handed Coordinate System

(curl fingers from u to v; thumb points to u x v)

$$\vec{a} \times \vec{b} = \det \begin{bmatrix} i & j & k \\ a_x & a_y & a_z \\ b_x & b_y & b_z \end{bmatrix} = \begin{bmatrix} a_y b_z - a_z b_y \\ a_z b_x - a_x b_z \\ a_x b_y - a_y b_x \end{bmatrix}$$

$$\vec{a} \times \vec{b} \overset{!}{=} -\vec{b} \times \vec{a}$$

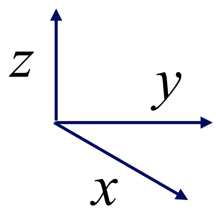
$$|a \times b| = |a||b||\sin\theta|$$

$$\vec{a} \times \vec{a} = \vec{0}$$

Math Review

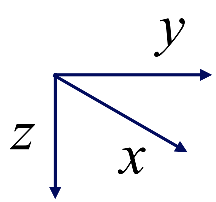
Coordinate Systems

Right-handed Coordinate System → nearly always use these



using right-hand rule

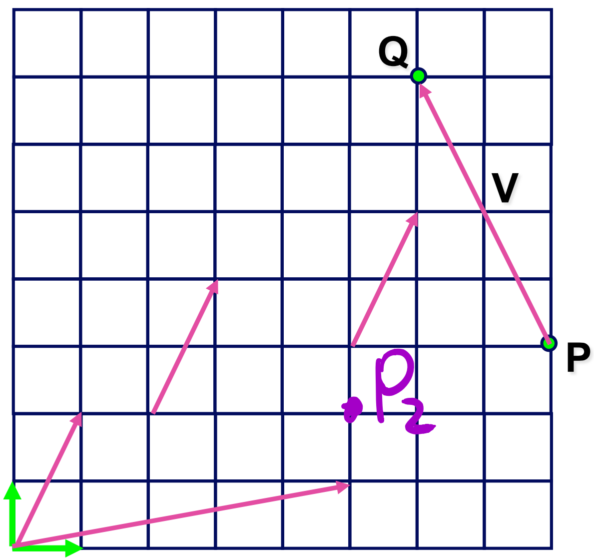
Left-handed Coordinate System → used for device coords (image)



using left-hand rule

Math Review

Points and Vectors



vectors

vector space

vectors are invariant under translation

$V + V' = V''$

$V - V' = V'''$

affine space:

points + vectors

allows vector-to-point addition

$P + V = Q$

$Q - P = V$

✓

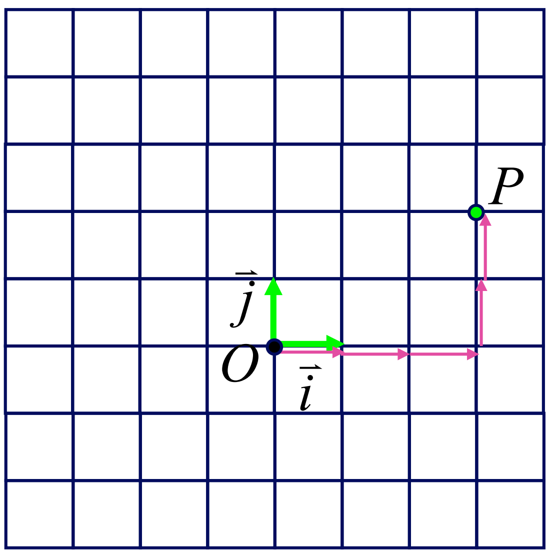
$P + P_2 = ?$

$0.6P = ?$

not meaningful

Math Review

Coordinate System vs Frame



vector space

coordinate system: basis vectors

frame: origin + basis vectors

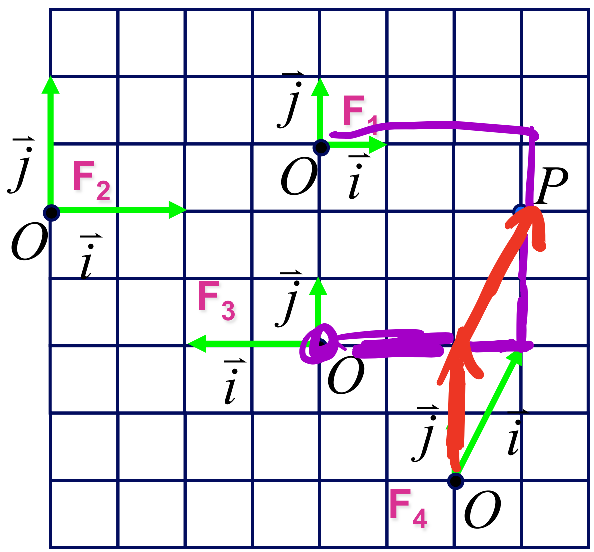
affine space

$$P = O + x\vec{i} + y\vec{j} + z\vec{k}$$

origin

Math Review

Working with Frames



$$P = O + x\vec{i} + y\vec{j}$$

F_1 (3, -1)

F_2 (3.5, 0)

F_3 (-1.5, 2)

F_4 (1, 2)

Many Coordinate Frames in a Scene

(and using transformation matrices to move between them)