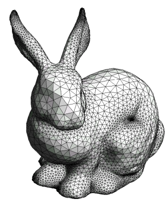
Rendering

Scene

3D objects
Coordinate Frame
Camera(s)
Materials
Lights



?

Framebuffer

final image

OpenGL Rendering Pipeline

(with some details abstracted away)

vertices

$V = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$

homogeneous coordinates

Javascript, three.js

GLSL program

vertex shader

$V' = MV$

4x4 matrix

vertex in device coords

"fixed function"

rasterization

(w, h)

fragments

GLSL program

fragment shader

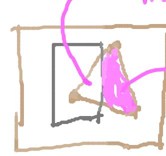
lighting, texture

1000 x 1000

Framebuffer

pixels

hidden fragments



visible fragments

Linear Algebra Review

vectors

$$a = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \quad \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad \begin{bmatrix} x \\ y \\ z \\ h \end{bmatrix}$$

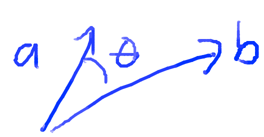
by default, assumption
<u>column</u> vectors

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

Math Review

dot product

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



$$= a_1 b_1 + a_2 b_2 + a_3 b_3$$
$$= |a||b| \cos\theta$$
$$= 0 \text{ when } a \perp b$$

max when a aligned with b
min when a is opp dir. of b

Math Review

matrix-vector multiplication

(a)as dot products with the rows

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

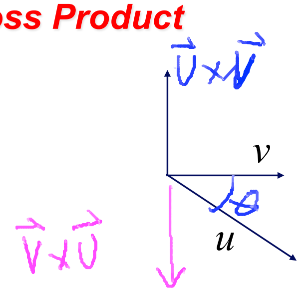
(b) as weighted combinations of the columns

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

$$\begin{bmatrix} i & j & k \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = x\vec{i} + y\vec{j} + z\vec{k}$$

Math Review

Cross Product



Right Handed Coordinate System

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

$\vec{U} \times \vec{V}$ is $\perp$ to $\vec{U}$ and $\vec{V}$

$$|U \times V| = |U|\,|V| \sin\theta$$

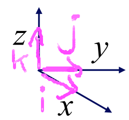
$$\vec{U} \times \vec{V} = -\vec{V} \times \vec{U}$$

Math Review

Coordinate Systems

Right-handed Coordinate System

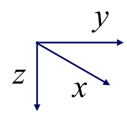
(typically used in CG)



using right-hand rule

$\vec{i} \times \vec{j} = \vec{k}$

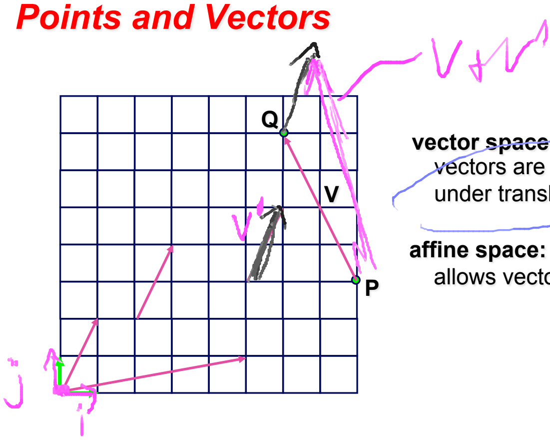
Left-handed Coordinate System



using left-hand rule

Math Review

Points and Vectors



vector space
vectors are invariant under translation

affine space:
allows vector-to-point addition

point + vector = poin

$P + V = Q$

$V = Q - P$

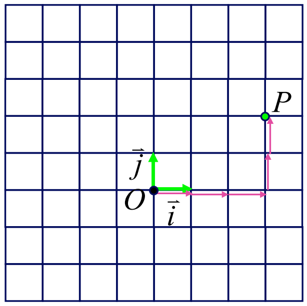
$V + V' = V$

$P + Q =$

point point

Math Review

Coordinate System vs Frame

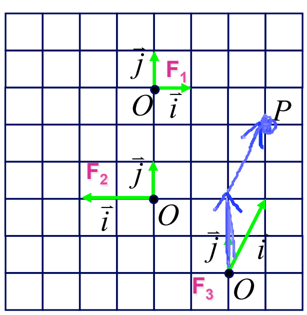


coordinate system: defined by set of basis ~~positions~~ velocities

frame: defined by origin and a set of ~~velocities~~ positions

Math Review

Working with Frames



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

F_1 $\quad P_1(3,-1)$

F_2 $\quad P_2(-1.5, 2) = O_2 - 1.5\, i_2 +$

F_3 $\quad P_3(1, 2) \quad = O_3 + 1 i_3 + 2$

Transformations

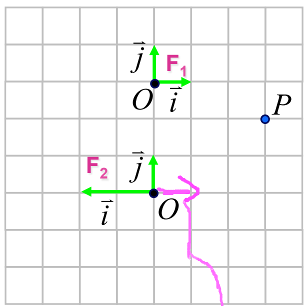
Transformations as a change of frame

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

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}_1 + x_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix}_1 + y_1 \begin{bmatrix} 0 \\ 1 \end{bmatrix}_1$$

Goal: $P_2 = M P$

$(-1.5, 2)$ $(3$



$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix}_2 = 1\begin{bmatrix} 0 \\ 3 \end{bmatrix}_2 + x_1 \begin{bmatrix} -0.5 \\ 0 \end{bmatrix}_2 + y_1 \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} -0.5 & 0 & 0 \\ 0 & 1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

check: