Principles of Computer Game Design and Implementation

Lecture 8

We already knew

- Basic Vectors
- Translation
- Movement
- Code for rotation

Examples on Rotation

```
private Vector3f axis = new Vector3f(1, 0, 0);
// private Vector3f axis = new Vector3f(0, 1, 0);
// private Vector3f axis = new Vector3f(0, 0, 1);
private Quaternion quat = new Quaternion();

 public void simpleUpdate(float tpf) {
     quat.fromAngleAxis(tpf, axis);
       //quat.fromAngleAxis(2 * FastMath.PI, axis);
       //quat.fromAngleAxis(FastMath.PI, axis);
       //quat.fromAngleAxis(0.3f * FastMath.PI, axis);
```

Examples on Rotation

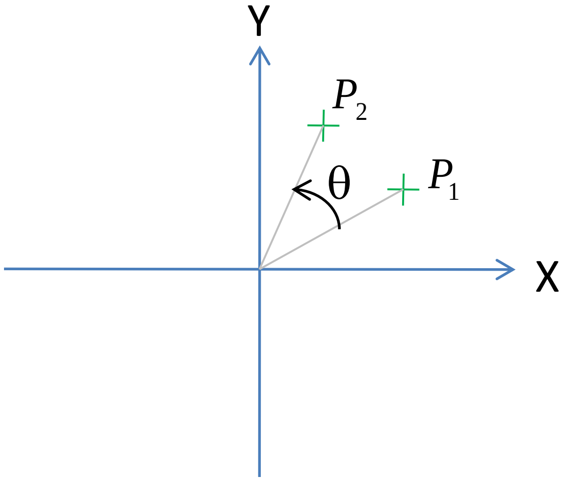
```
public void simpleUpdate(float tpf) {
    b.rotate(0.3f * FastMath.PI, 0 ,0);
    // b.rotate(0, 0.3f * FastMath.PI, 0);
    // b.rotate(0.3f * FastMath.PI, 0 ,0);
  }
```

Outline for Today

- Math for Rotation

Rotation

- Translation is easy, rotation is harder



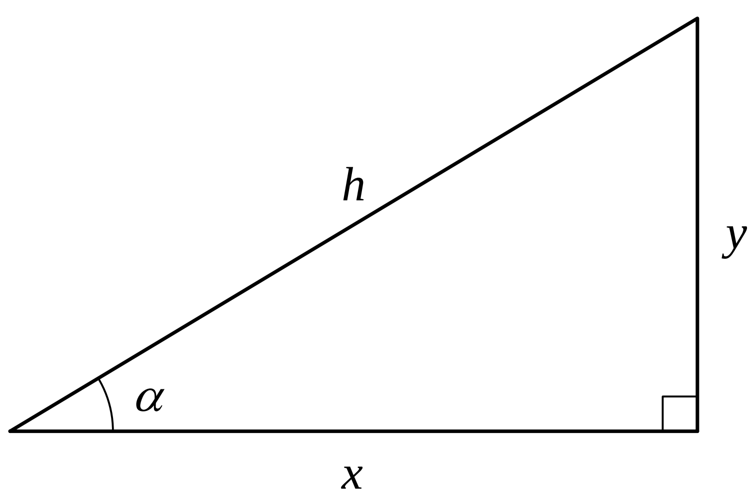
<u>Rotation by angle θ around origin</u>

P_1 rotated to P_2

Positive rotation is counter-clockwise

Applied Trigonometry

- Trigonometric functions
  - Defined using right triangle



$$\sin\alpha = \frac{y}{h}$$

$$\cos\alpha = \frac{x}{h}$$

$$\tan\alpha = \frac{y}{x} = \frac{\sin\alpha}{\cos\alpha}$$

Applied Trigonometry

- Angles measured in radians

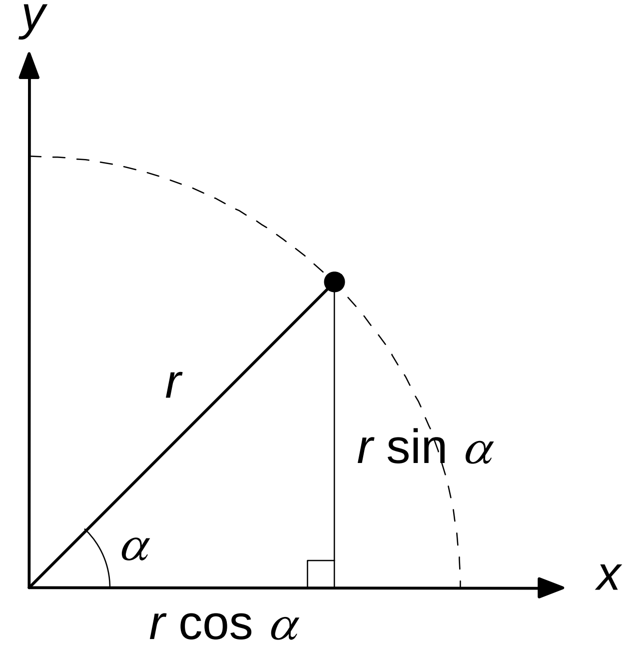
$$radians = \frac{\pi}{180}(degrees)$$

$$degrees = \frac{180}{\pi}(radians)$$

- Full circle contains 2π radians

Applied Trigonometry

- Sine and cosine used to "decompose" a point into horizontal and vertical components



Basic Trigonometric Identities (1)

$$\sin^2\alpha + \cos^2\alpha = 1$$

$$\sin(-\alpha) = -\sin\alpha$$

$$\cos(-\alpha) = \cos\alpha$$

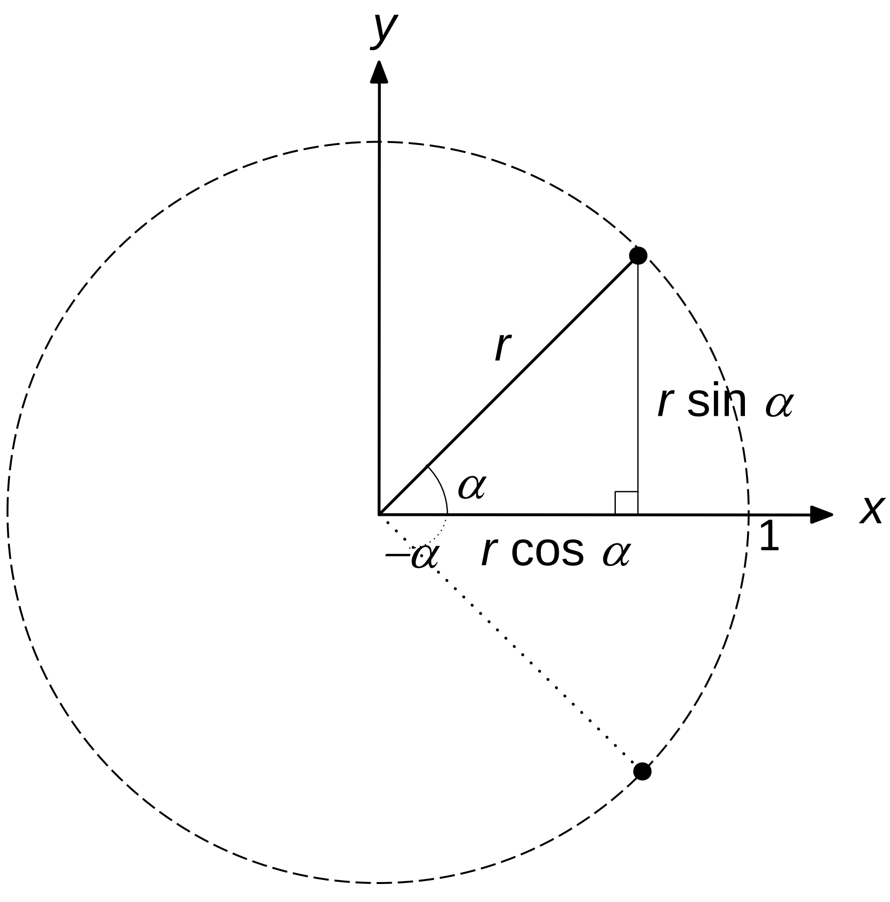
$$\tan(-\alpha) = -\tan\alpha$$

$$\cos\alpha = \sin(\alpha + \pi/2)$$

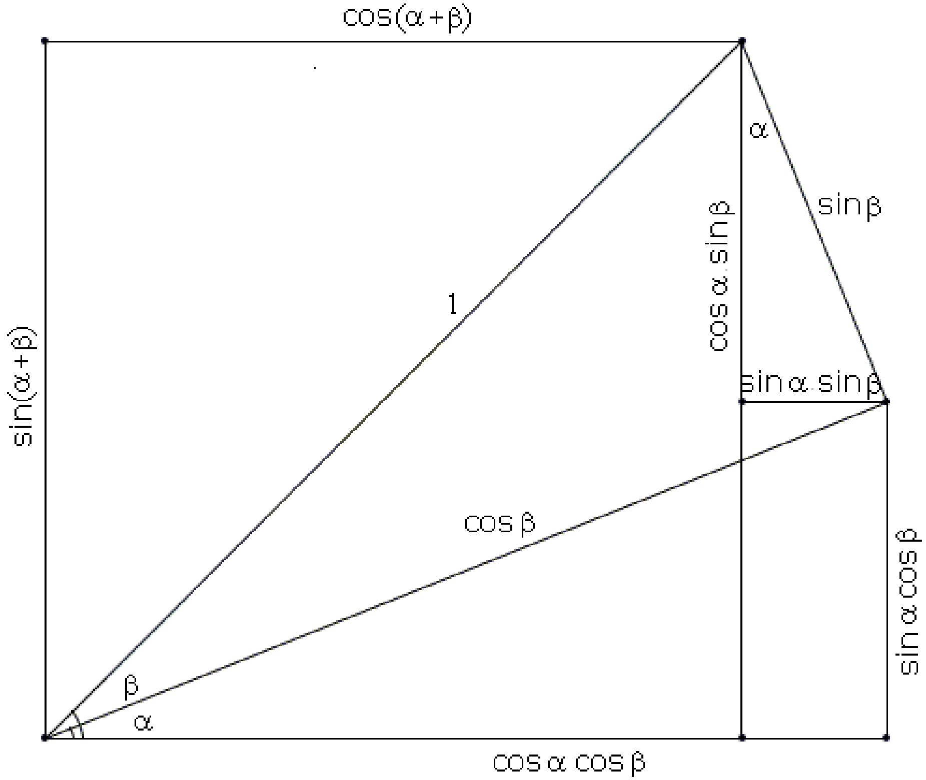
$$\sin\alpha = \cos(\alpha - \pi/2)$$

$$\cos\alpha = -\sin(\alpha - \pi/2)$$

$$\sin\alpha = -\cos(\alpha + \pi/2)$$



Basic Trigonometric Identities (2)



$$\sin(\alpha + \beta) = \cos\alpha \sin\beta + \sin\alpha \cos\beta$$

$$\cos(\alpha + \beta) = \cos\alpha \cos\beta - \sin\alpha \sin\beta$$

http://www.cut-the-knot.org/triangle/SinCosAddition.gif

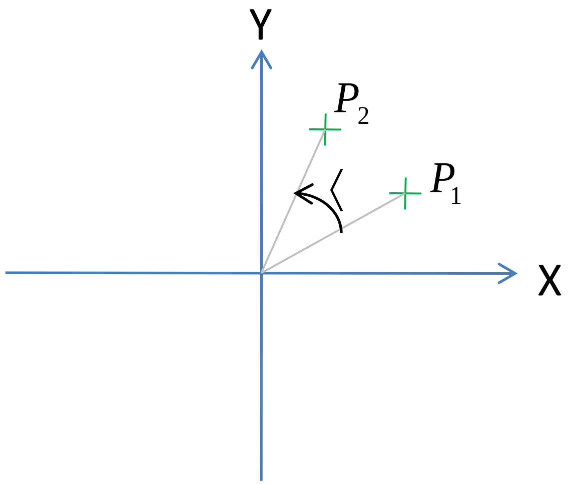
Applied Trigonometry

- Inverse trigonometric functions
  - Return angle for which sin, cos, or tan function produces a particular value
  - If $\sin \alpha = z$, then $\alpha = \sin^{-1} z$
  - If $\cos \alpha = z$, then $\alpha = \cos^{-1} z$
  - If $\tan \alpha = z$, then $\alpha = \tan^{-1} z$

Maths in jME

- FastMath package
  - FastMath.PI = π
  - FastMath.sin(float x)
  - FastMath.cos(float x)
  - FastMath.tan(float x)
  - FastMath.asin(float x) $= \sin^{-1}(x)$
  - FastMath.acos(float x) $= \cos^{-1}(x)$
  - FastMath.atan(float x) $= \tan^{-1}(x)$
  - …

Rotation in 2D

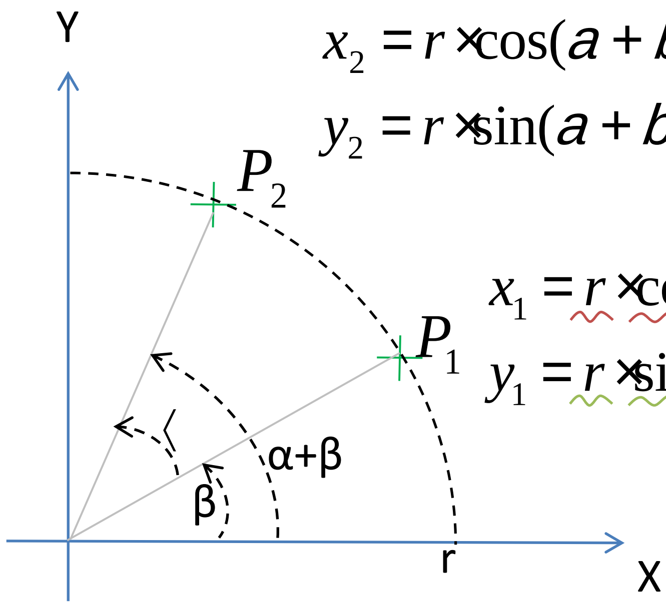


<u>Rotation by angle 〈 around origin</u>

P_1 rotated to P_2

Positive rotation is counter-clockwise

Rotation in 2D



$$x_2 = r \times \cos(a + b)$$

$$y_2 = r \times \sin(a + b)$$

$$x_1 = r \times \cos b$$

$$y_1 = r \times \sin b$$

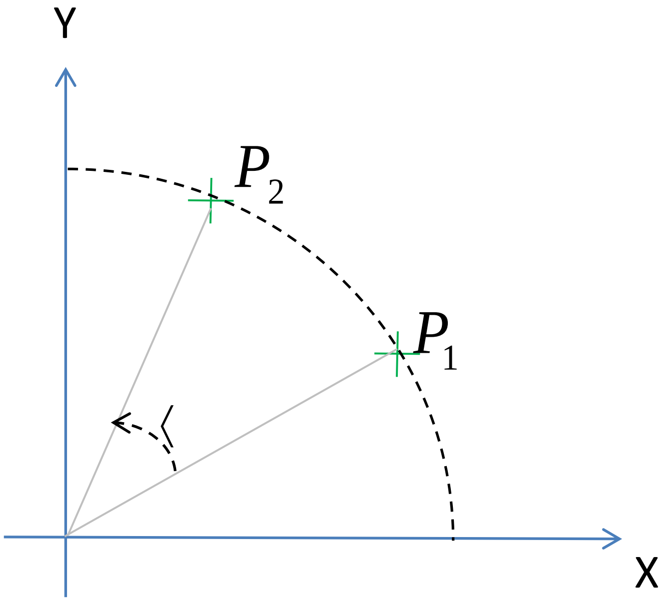
$$\sin(a + b) = \cos a \sin b + \sin a \cos b$$

$$\cos(a + b) = \cos a \cos b - \sin a \sin b$$

$$x_2 = r \times \cos(a + b) = r \times \cos a \cos b - r \times \sin a \sin b = x_1 \times \cos a - y_1 \times \sin a$$

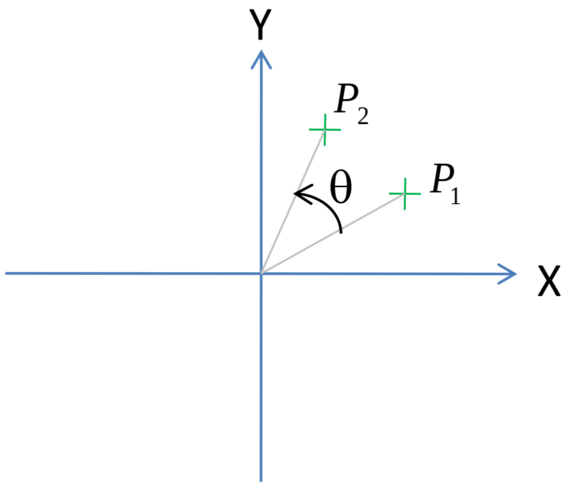
$$y_2 = r \times \sin(a + b) = r \times \cos a \sin b + r \times \sin a \cos b = y_1 \times \cos a + x_1 \times \sin a$$

Rotation in 2D



$$(x_2, y_2) = (x_1 \cos a - y_1 \sin a, x_1 \sin a + y_1 \cos a)$$

Example



$\underline{\theta = 27°}$

$P_1 = (3,2)$

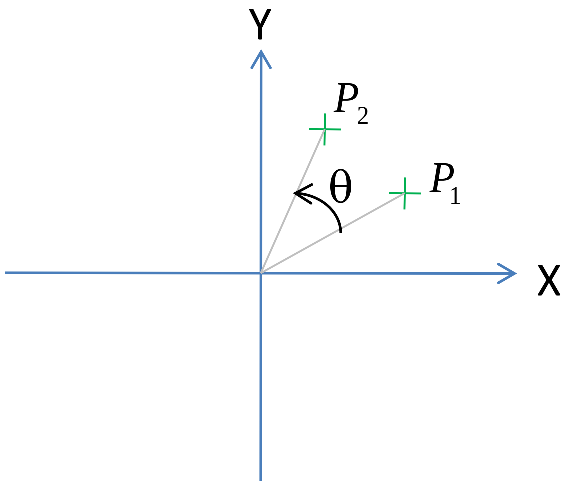
$\sin(27°) \approx 0.454$

$\cos(27°) \approx 0.891$

$$x_2 = 3\cos(27°) - 2\sin(27°) \approx 1.765$$

$$y_2 = 3\sin(27°) + 2\cos(27°) \approx 3.144$$

Rotation in 2D: Linear Form



<u>Rotation by angle θ around origin</u>

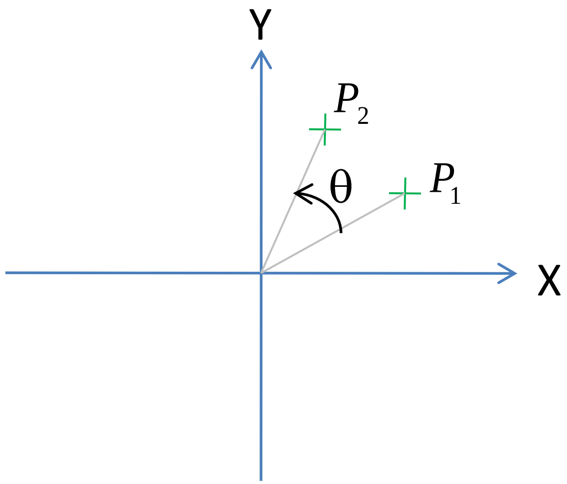
P_1 rotated to P_2

Positive rotation is counter-clockwise

$$x_2 = x_1 \cos q - y_1 \sin q$$

$$y_2 = x_1 \sin q + y_1 \cos q$$

Rotation in 2D: Matrix Form



<u>Rotation by angle θ around origin</u>

P_1 rotated to P_2

Positive rotation is counter-clockwise

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

Same thing expressed differently

Matrices

- A matrix is a rectangular array of numbers arranged as rows and columns
  - A matrix having *n* rows and *m* columns is an $n \times m$ matrix
  - At the right, **M** is a 2×3 matrix

$$\mathbf{M} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

  - If $n = m$, the matrix is a square matrix

Matrix Elements

- The entry of a matrix **M** in the *i*-th row and *j*-th column is denoted M_{ij}
- For example,

$$\mathbf{M} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$M_{11} = 1 \quad M_{21} = 3$$

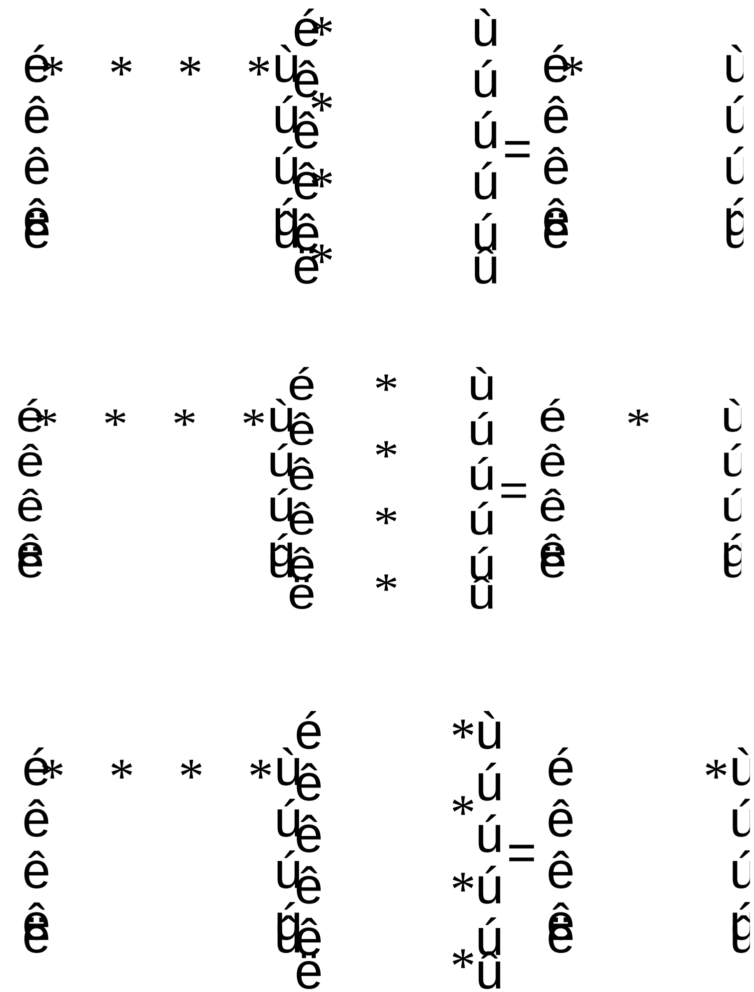
$$M_{12} = 2 \quad M_{22} = 4$$

Matrix Multiplication

Product of two matrices **A** and **B**

- Number of columns of **A** must equal number of rows of **B**
- Entries of the product are given by

$$(\mathbf{AB})_{ij} = \sum_{k=1}^{m} A_{ik} B_{kj}$$



Example

$$\mathbf{M} = \begin{bmatrix} 2 & 3 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} -2 & 1 \\ 4 & -5 \end{bmatrix} = \begin{bmatrix} 8 & -13 \\ -6 & 6 \end{bmatrix}$$

$$\begin{aligned} M_{11} &= 2\cdot(-2)+3\cdot 4 &&= 8 \\ M_{12} &= 2\cdot 1+3\cdot(-5) &&= -13 \\ M_{21} &= 1\cdot(-2)+(-1)\cdot 4 &&= -6 \\ M_{22} &= 1\cdot 1+(-1)\cdot(-5) &&= 6 \end{aligned}$$

Vector as Matrix

- A vector **V** = (x,y,z) can be represented as a 1x3 matrix

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

- A *vertical vector*

2D Rotation Matrix

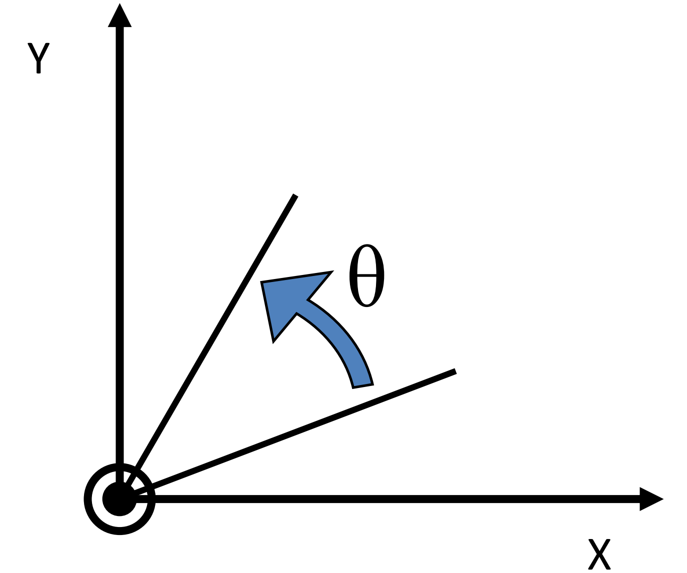
$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} \cos\theta \cdot x_1 - \sin\theta \cdot y_1 \\ \sin\theta \cdot x_1 + \cos\theta \cdot y_1 \end{bmatrix}$$

That is,

$$(x_2, y_2) = (x_1 \cos\theta - y_1 \sin\theta, x_1 \sin\theta + y_1 \cos\theta)$$

Rotation around Z

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} \cos q & -\sin q & 0 \\ \sin q & \cos q & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}$$



- What about an arbitrary matrix?

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = M \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}$$

Some Useful Transformations

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}$$

Identity matrix (no change)

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} c & 0 & 0 \\ 0 & c & 0 \\ 0 & 0 & c \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}$$

Scale

Combination of Transformations

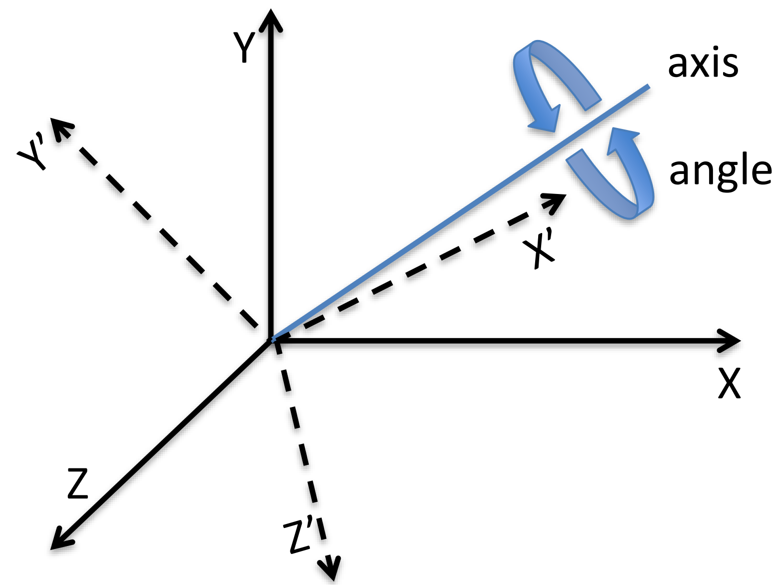
$$\begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} = M' \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} \qquad \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = M \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}$$

$$\begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} = \left(M \times M'\right) \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix}$$

Quaternions and Rotation Matrices

- Quaternions we looked at previously can generate rotation matrices
- Get a good book on linear algebra if you want to know more

Quaternion from 3 Vectors



- q.fromAngleAxis(angle, axis) : (**x**,**y**,**z**) -> (**x**1,**y**1,**z**1)
- `q.fromAxes(`**x**`1,`**y**`1,`**z**`1) –`
  `"inverse"`