* Title
* Agenda
* Rotation in 2D
* Rotation in 2D
* Final 2D Positions with cos and sin about the axis on the unit circle
* Matrices and complex number rotations
* Resulting Equations from Matrices and Complex Numbers
* XYZ Axis
* 3D Rotation Matrices
* Overall Rotation Matrix
* Resulting Equations
* Throwback to 2D Rotation Equations
* Back To Resulting Equations
* Question
  + Why aren’t they spinning
* Throwback to independent rotation matrices
* What if we rotate by 90deg on Y axis
* Multiply Matrices: Gimbal Locked Rotation Matrix
* Throwback to 3D Rotation Matrix
* Gimbal Locked Rotation Matrix
* Title
* Agenda
* Section Title
  + Rotations in 2D with Matrices
* 2D Rotation Matrix
* Matrix-Vector Matrix Multiplication
* Resulting Equations
* Section Title
  + Rotations in 3D with Matrices
* 3D Rotation Matrices
* 3D Rotation Matrix
* Matrix-Vector Rotation Multiplication
* Resulting Equations
* Statement
  + That’s only one way to rotate in space
* Section Title
  + Rotations in 2D with Complex Numbers
* Complex numbers with i
* Complex number multiplication
* Resulting Equation
* Compare 2D Rotation Equations with Matrices and Complex Numbers
* Intro to Quaternions
* What is a Quaternion
  + A Quaternion is a complex number with an additional 2 imaginary number parts
* Imaginary Numbers
  + Give definitions for i,j,k
  + Give definition that ijk = -1
* Give basic rules for Quaternion Arithmetic
  + Quaternion Arithmetic works very similarly to Vector Arithmetic
    - Quaternions can dot and cross with other quaternions
* Define a Quaternion rotation
* Show the equation for Rotating a point with a Quaternion rotation
* Show an interesting identity used in the equation for the quaternion rotation
  + Because *u* will always be orthogonal to *v*
* Show an interesting identity used in the equation for the quaternion rotation
  + - Because a Quaternion dotted with itself is the square of it’s modulus (in this case 1)
  + By Lagrange’s Triple Product Formula
* Question
  + How do we find what Quaternion Rotation we need to apply
* Define Quaternion Rotation Combination