More on Rotation

- Rotation matrices can accumulate error
  - Rows/columns can grow/shrink to be other than unit length
  - Rows/columns can skew so that they are no longer mutually orthogonal
More on Rotation

- A rotation can also be represented by an ordered set of 3 rotations about a set of mutually orthogonal axes
  - E.g., roll, pitch, yaw
  - Can result in *gimbal lock*

Gimbal Lock

3 gimbals: 1, 2, 3

- *Gimbals* are concentric mounting rings
  - Each ring pivots on an axis to provide 1 degree of rotational freedom
  - Rotation axes are orthogonal
- *Gimbal lock* occurs when 2 axes align
  - 1 degree of rotational freedom is lost
Gimbal Lock

- Rotation about Y axis

- Rotation about Z axis
Gimbal Lock

- Rotation about X axis
- “Lost” degree of rotational freedom

1 and 3 are aligned! Can’t respond to roll

Experiencing gimbal lock with a 3D tripod head
**Gimbal Lock**

- Experiencing gimbal lock with a 3D tripod head
  - Adjust starting from object (camera)
    - roll, pitch, yaw around axes of parent’s (tripod) coord sys
  - Adjust starting from parent (tripod)
    - yaw, pitch, roll around axes of object’s (camera) coord sys

---

**Demo: The Perils of Roll, Pitch, & Yaw**
**Gimbal Lock**

- A real physical problem in aircraft/spacecraft navigation systems with physical gimbals
- A real conceptual problem in 3D user interfaces
  - There is *always* an ordered set of rotations about the given axes that will produce a desired rotation
    - But, it might not involve an intuitive incremental change from the current set of rotations

**Axis of Rotation and Angle**

- Can represent any rotation as an axis of rotation and an angle
  - Avoids gimbal lock
  - But, matrices, rotations about an ordered set of axes, axis of rotation and angle all have problems interpolating between two rotations
    - Smoothly
    - Along shortest path
Quaternions  W. Hamilton, 1843

- A quaternion is
  - A four-tuple $xi + yj + zk + w$, where $i, j, k$ are imaginary
    
    \[
    i^2 = j^2 = k^2 = ijk = -1 \\
    ij = k \quad ji = -k \\
    jk = i \quad kj = -i \\
    ki = j \quad ik = -j
    \]
  - $q = (v, w) = v_x i + v_y j + v_z k + w$
  - $v$ is a vector, $w$ is a scalar

- A unit quaternion ($|q| = 1$) represents rotation about normalized rotation axis $u$ by angle $\theta$
  - $v = (\sin(\theta/2)) u$
  - $w = \cos(\theta/2)$
  
  \[
  |q| = \sqrt{v_x^2 + v_y^2 + v_z^2 + w^2}
  \]
  - A unit quaternion is a point on a 4D unit sphere
Quaternions  W. Hamilton, 1843

- Converting to a rotation matrix

\[
(v, w) = \begin{bmatrix}
1 - 2y^2 - 2z^2 & 2xy + 2wz & 2xz - 2wy & 0 \\
2xy - 2wz & 1 - 2x^2 - 2z^2 & 2yz + 2wx & 0 \\
2xz + 2wy & 2yz - 2wx & 1 - 2x^2 - 2y^2 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

- Multiplying quaternions

\[q_1 \cdot q_2 = (v_1, w_1) \cdot (v_2, w_2) = (w_1v_2 + w_2v_1 + v_1 \otimes v_2, w_1w_2 - v_1v_2)\]

  - Product of two unit quaternions corresponds to composition of their rotations (associative, but not commutative)

Quaternions  W. Hamilton, 1843

- Spherical Linear Interpolation (SLERP)

  - Smoothly interpolates two quaternions as points on 4D unit sphere
    - Constant angular velocity along great circle between two points on 4D unit sphere

- Support in Unity

  - You do not have to implement the math yourself!
  - https://unity3d.com/learn/tutorials/modules/intermediate/scripting/quaternions