ESCOLA DE NOVES TECNOLOGIES INTERACTIVES

centre adscrit a: UNIVERSITAT DE BARCELONA

Animation Foundations

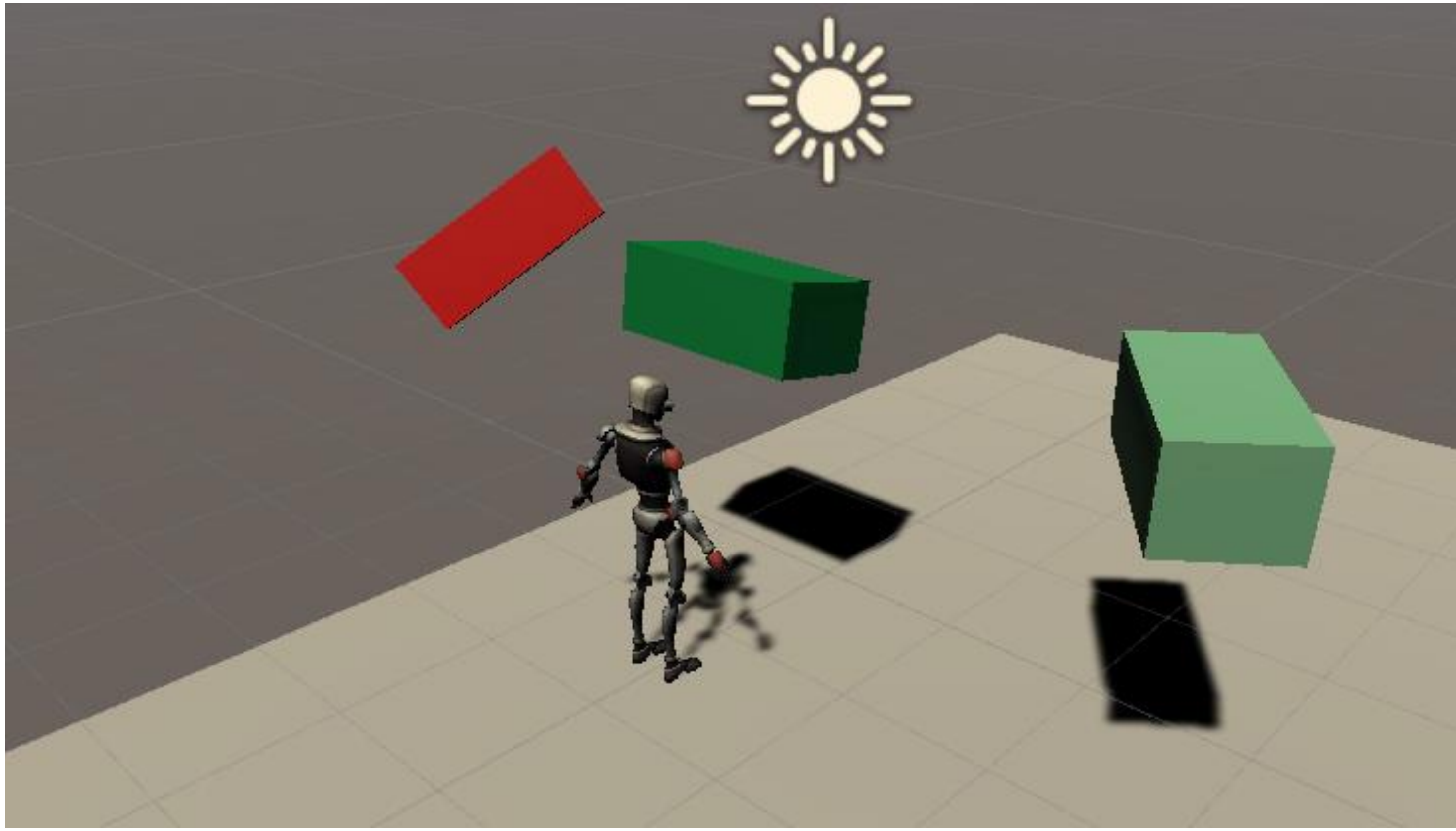
06. Introduction to Constraints

Joints

Twist and Swing

Last week

- Axis angle rotations in practice
- Quaternion rotation
  - Making a rotation
  - Finding a rotation offset
  - Maintaining a rotation offset
  - Removing a rotation offset



Additional resources for rotations and quaternions

- http://www.euclideanspace.com/maths/algebra/realNormedAlgebra/quaternions/index.htm

- https://www.youtube.com/watch?v=SCbpxiCN0U0&list=PLW3Zl3wyJwWOpdhYedlD-yCB7WQoHf-My&index=32

Exercise 5 (last week)

Write your own Quaternion class that:

- Always keeps values normal
- Can multiply quaternions
- Can invert quaternions
- Can convert from axis angle
- Can convert to axis angle
- Optionally, gives a warning if it is rotating more tan 180º

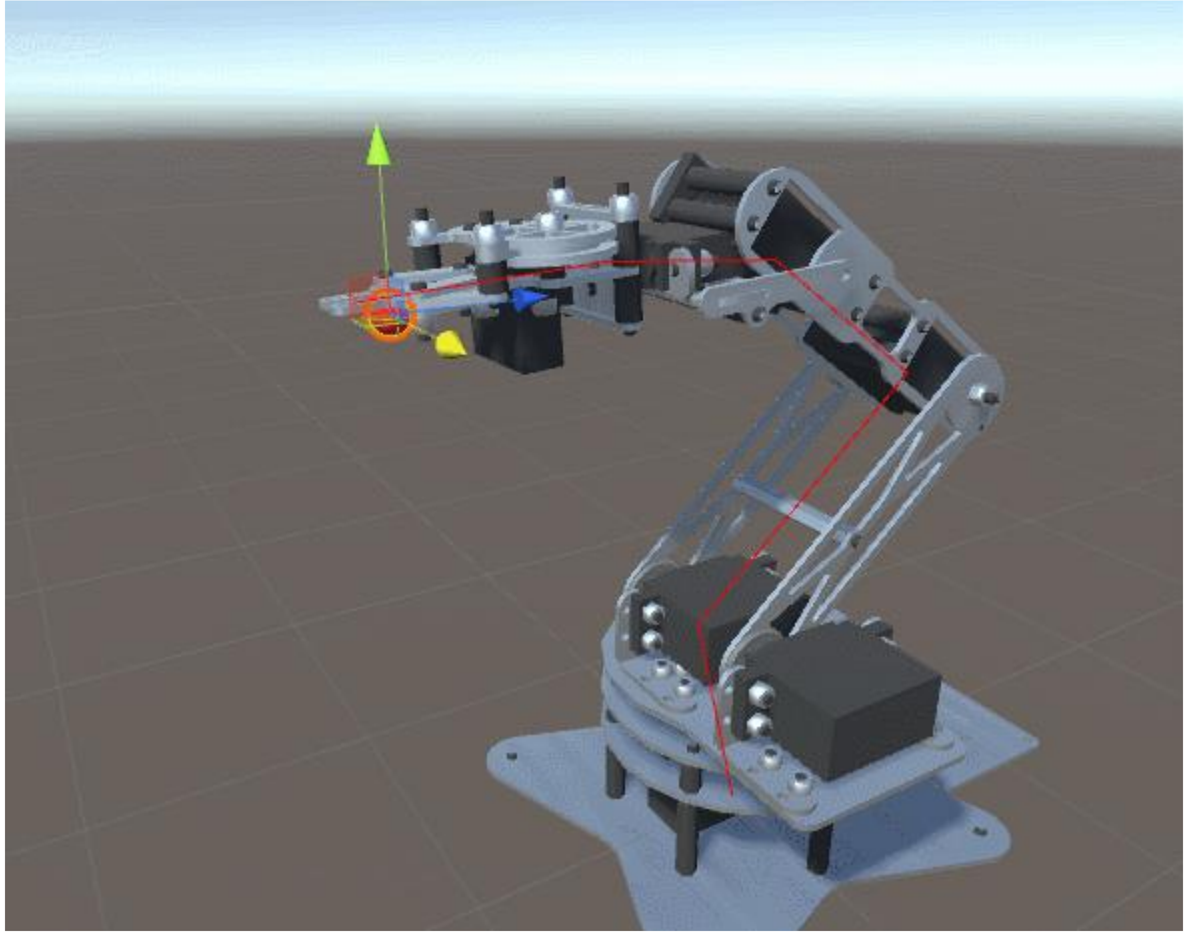
- Check that exercise 4 still Works when using it

To design the class, imagine that in the future you might want to encapsulate it in a .dll

- Base it solely on the Mathf library
- Make it independent from gameObject

Outline for next weeks

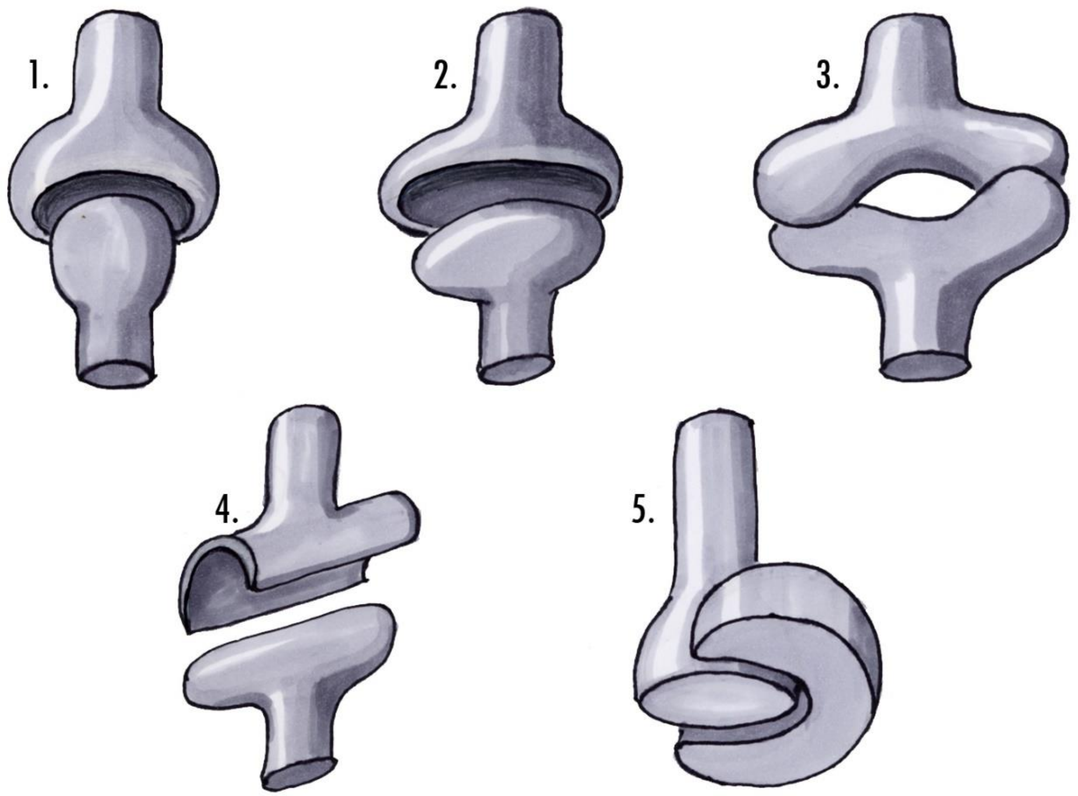
- Intro to Constraints
- Forward Kinematics
- Inverse Kinematics (IK)
  - Cyclic Coordinate Descent
  - Fabric
  - Gradient Descent
- IK with constraints



Theory: Joints and Constraints

Different kinds of joints

1. Ball-and-Socket *
2. Condyloid
3. Saddle
4. Hinge *
5. Pivot

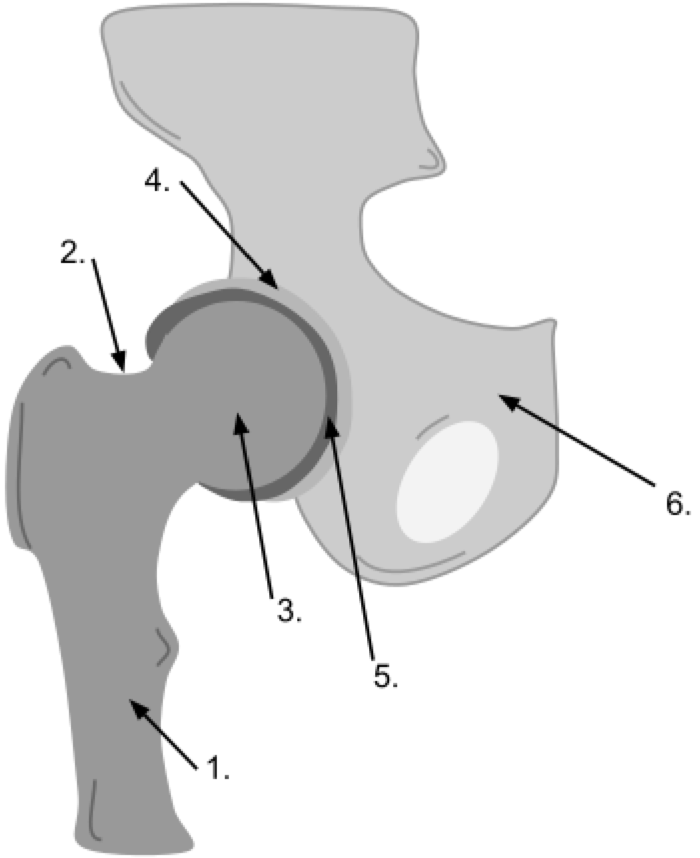


* Most used

Theory: The ball and Socket Joint

Details of hip joint

1. Femur
2. Femoral Neck
3. Femoral Head
4. Acetabulum
5. Acetabular Labrum
6. Pelvis

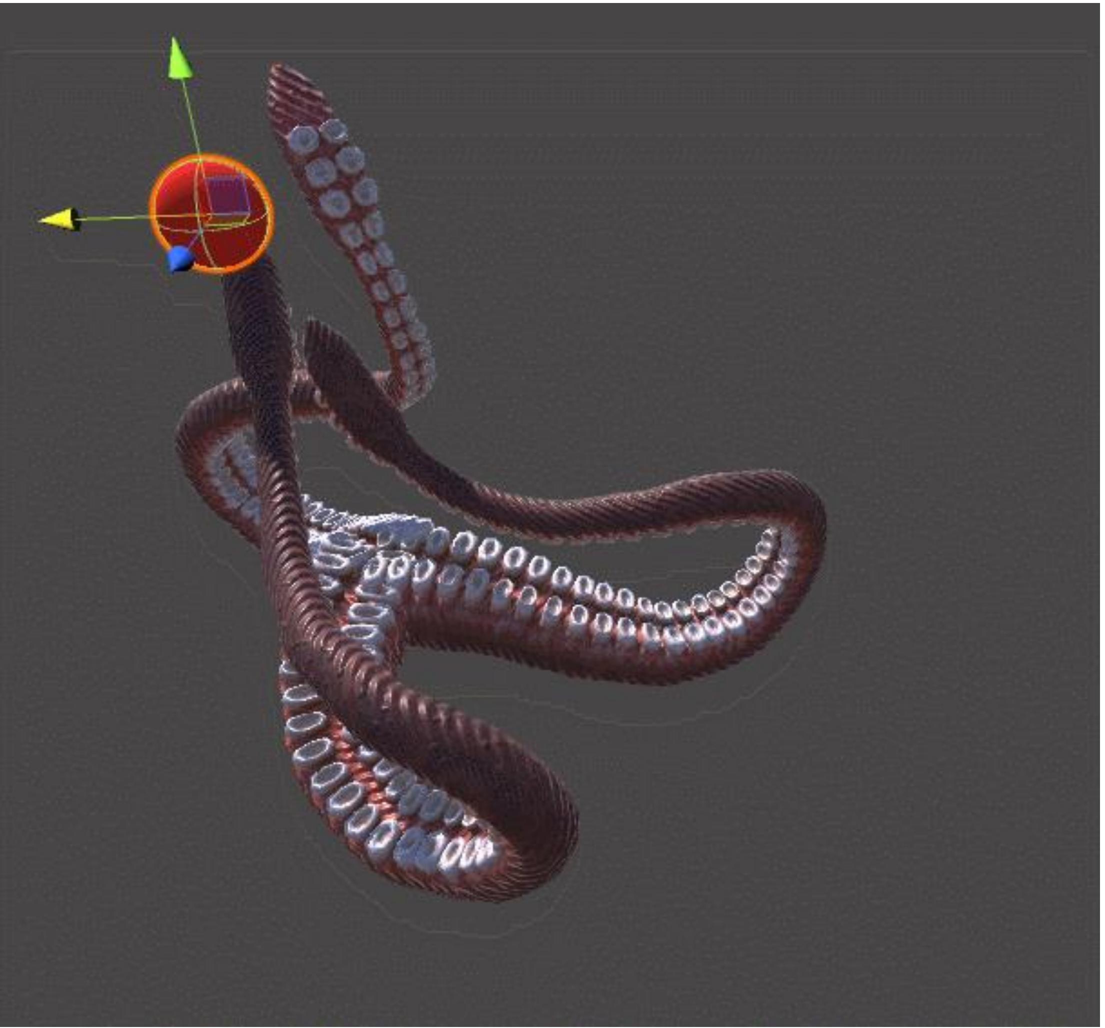


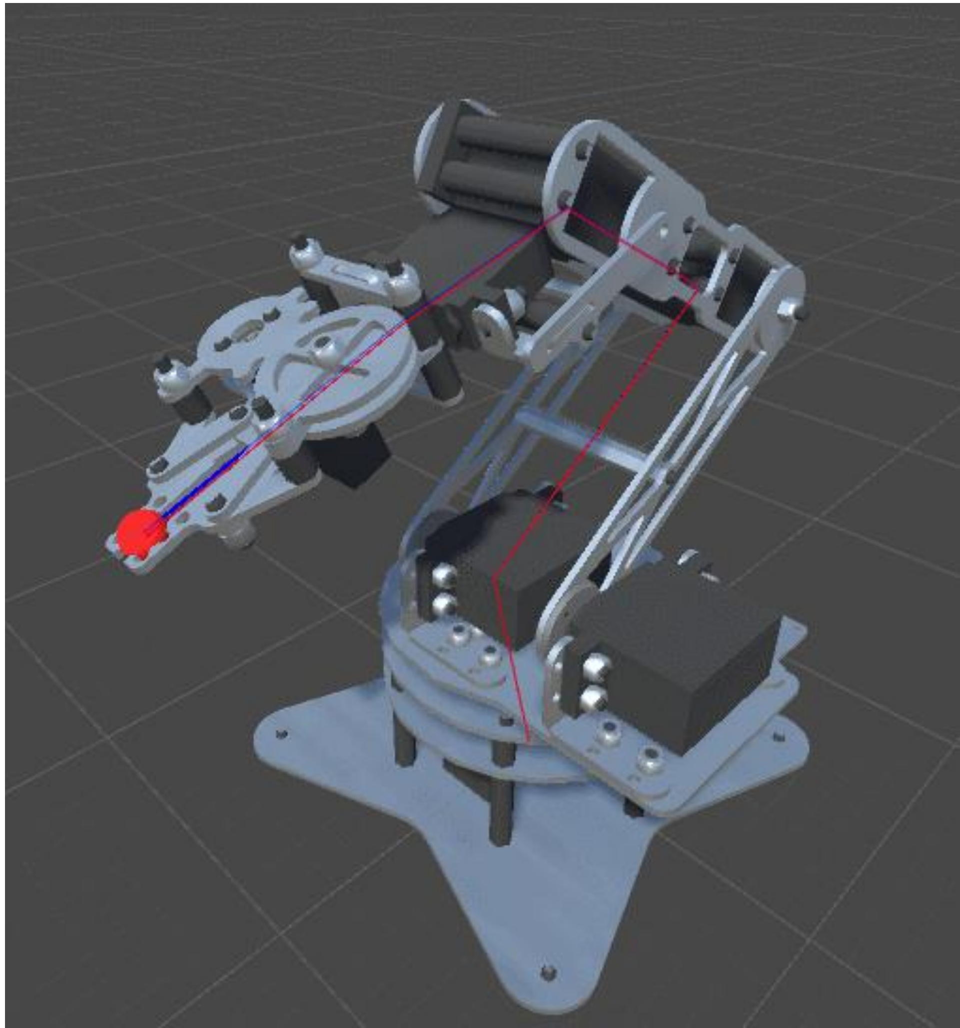
Theory: The ball and Socket Joint

It allows rotations in all directions, but with constraints

Hinge will have additional constraints

Why Different Constraints?

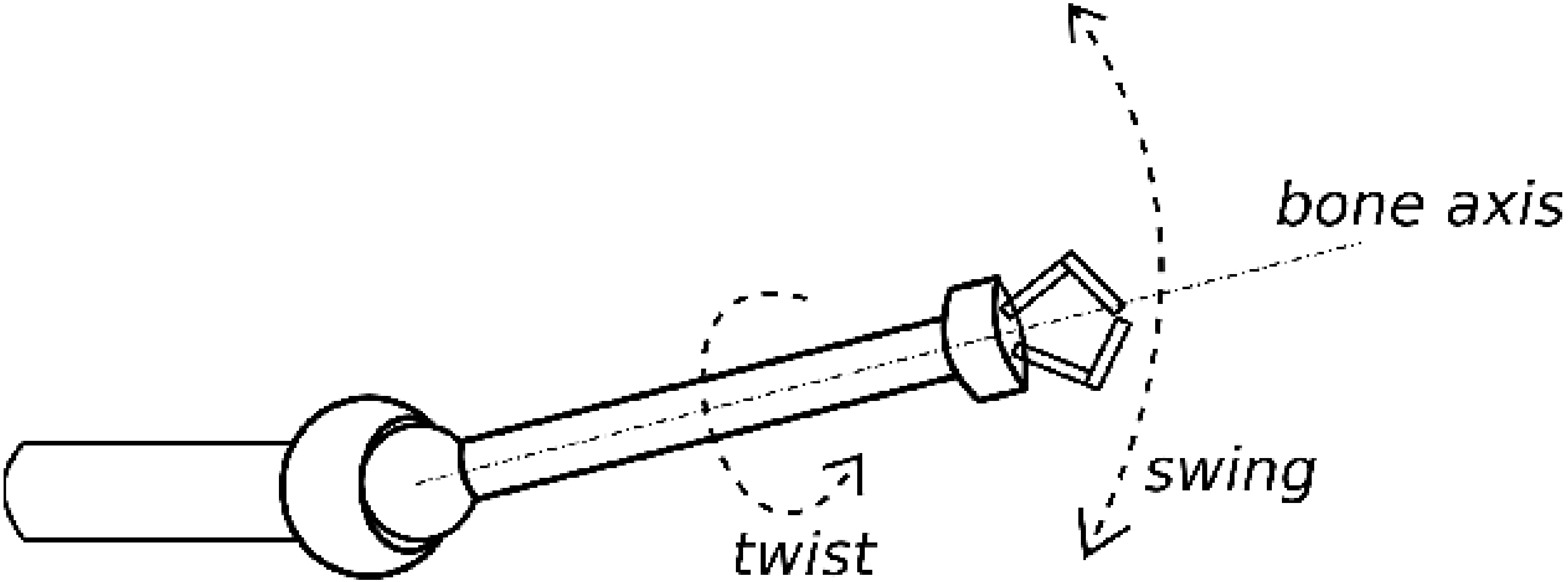




Why Different Constraints?

- What are the different constraints of the two previous models?
- How do we capture these in a 3D real-time Simulation?

Theory: twist and Swing decomposition



Decomposition aligned with the z axis

Given rotation q_r

$$q_r(x, y, z, w) = q_{twist} q_{swing}$$

Twist (on z):

$$q_{twist} = (0,0, q_z, q_w) \frac{1}{(\sqrt{q_w{}^2 + q_z{}^2})}$$

Swing:

$$q_{swing} = (0,0, \frac{q_w q_y - q_x q_z}{(\sqrt{q_w{}^2 + q_z{}^2})}, q_w)$$

Decomposition aligned with the z axis

Given rotation q_r

$$q_r(x, y, z, w) = q_{twist} q_{swing}$$

Swing?

Twist (on z):

$$q_{twist} = (0,0,q_z,q_w)\frac{1}{(\sqrt{{q_w}^2 + {q_z}^2})}$$

Decomposition aligned with the z axis

Simpler algorithm:

Given rotation q_r

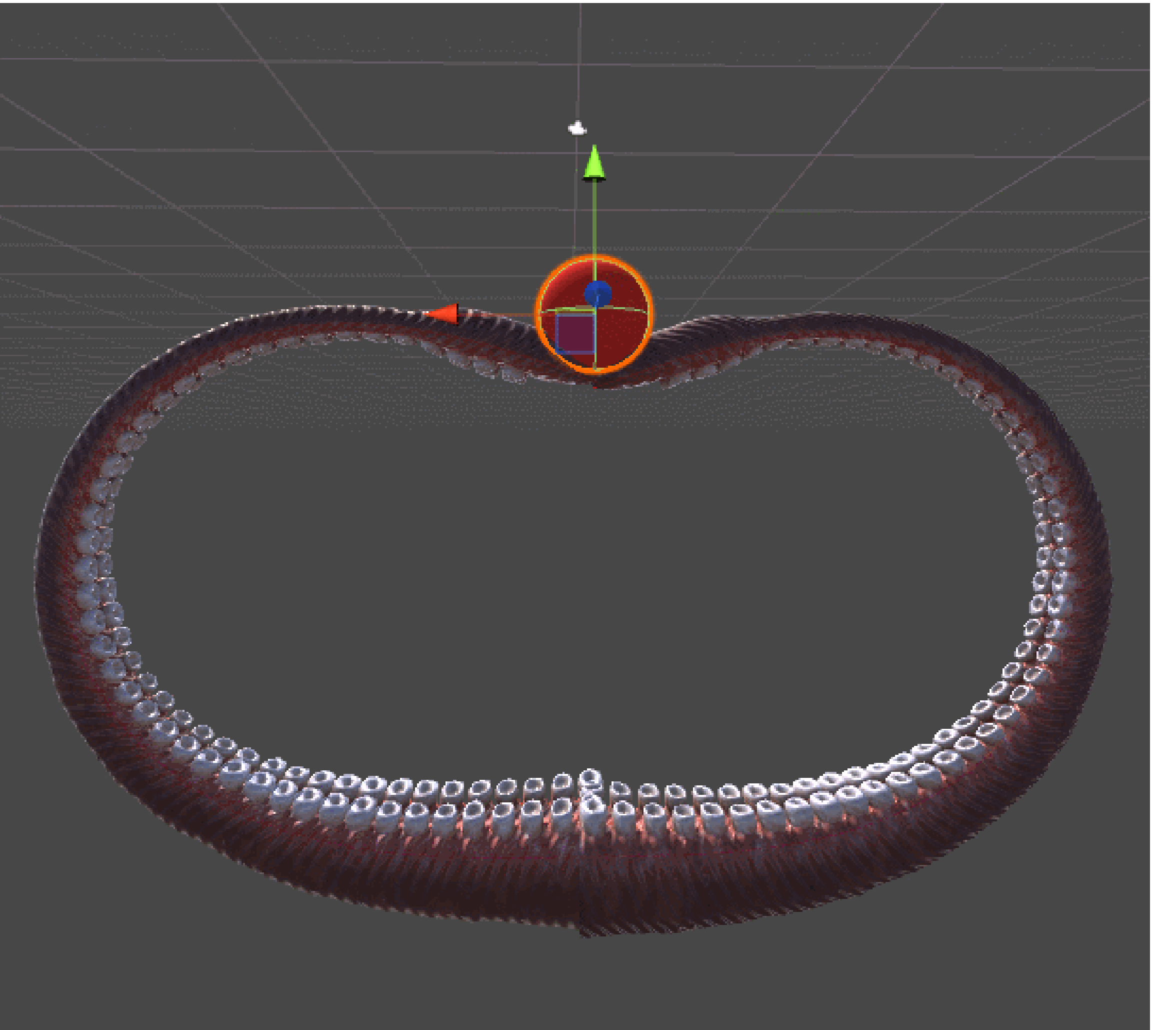
$$q_r = q_{twist} q_{swing}$$

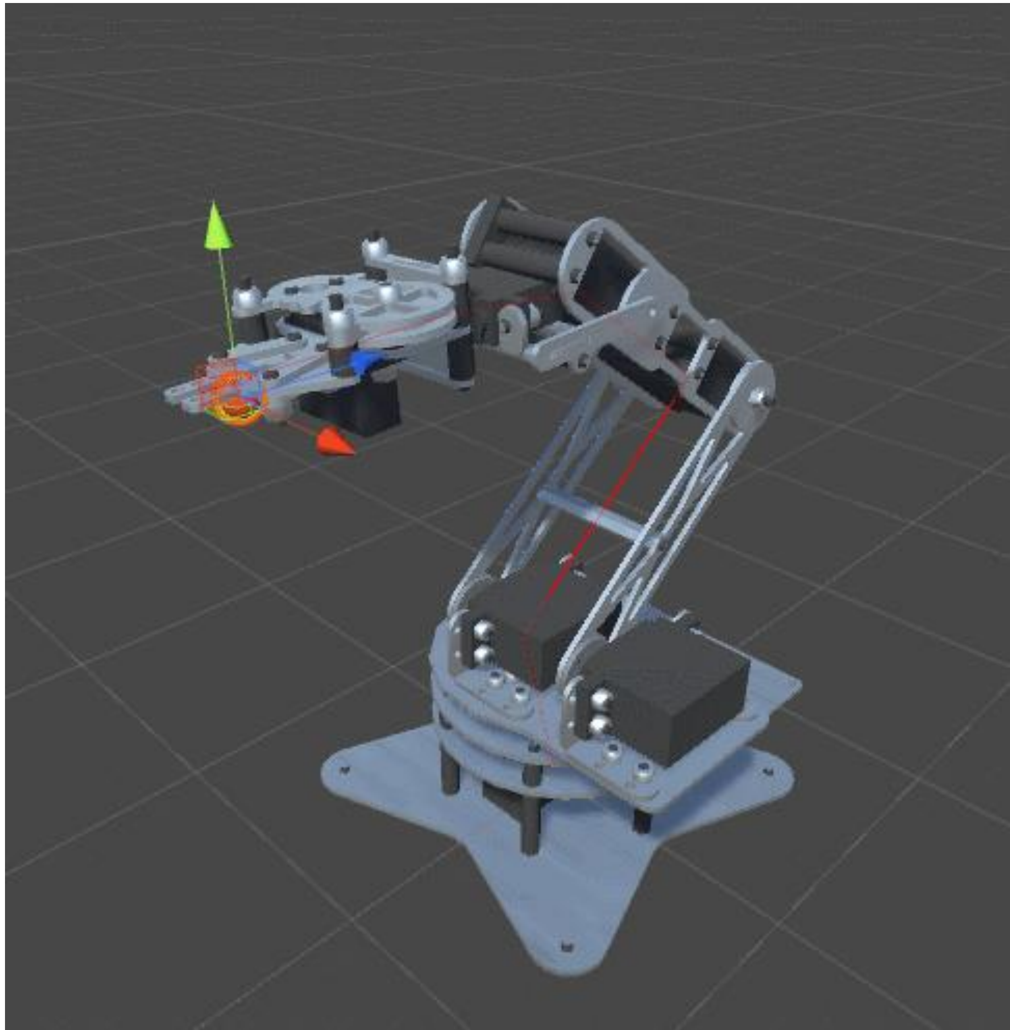
Algorithm:

q_{twist} = normalize(Quaternion(0, 0, qr.z, qr.w));

q_{swing} = conjugate(qt) * qr ;

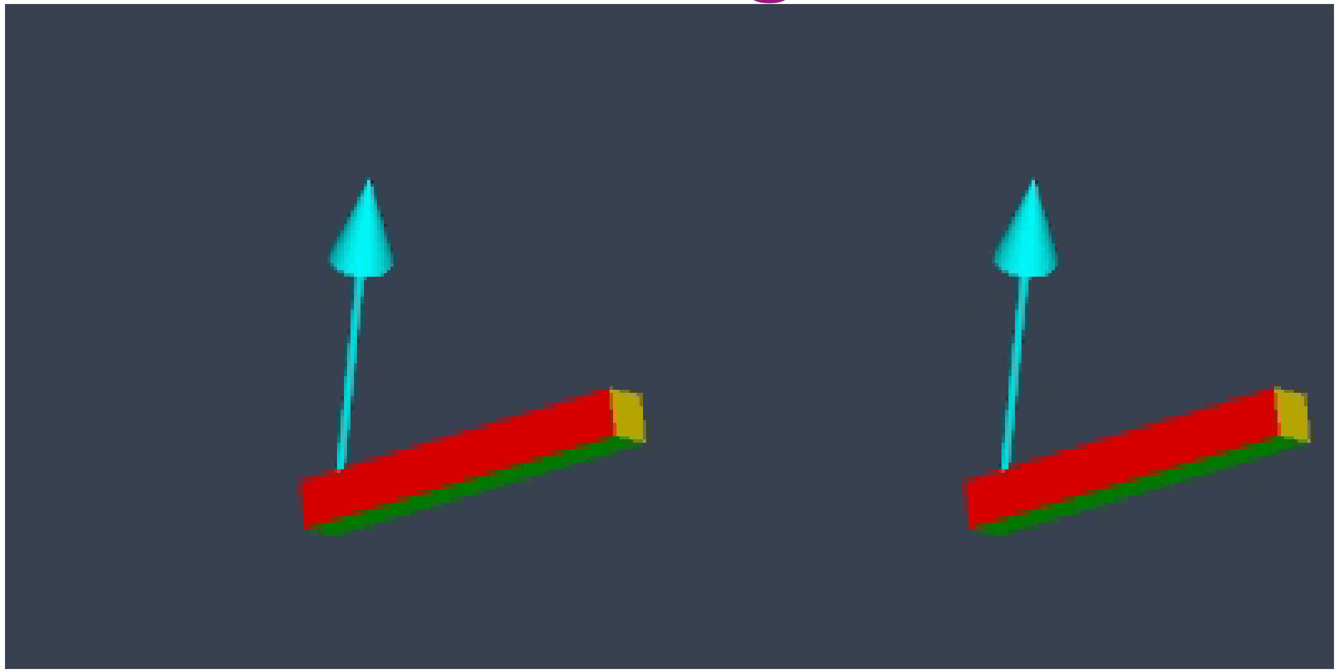
Why different constraints (in IK)?





Additional resources for twist-swing decomposition

What is the difference between
these two animations?



- https://www.gamedev.net/forums/topic/696882-swing-twist-interpolation-sterp-an-alternative-to-slerp/
- https://stackoverflow.com/questions/3684269/component-of-a-quaternion-rotation-around-an-axis
- http://www.euclideanspace.com/maths/geometry/rotations/for/decomposition/