RTIK

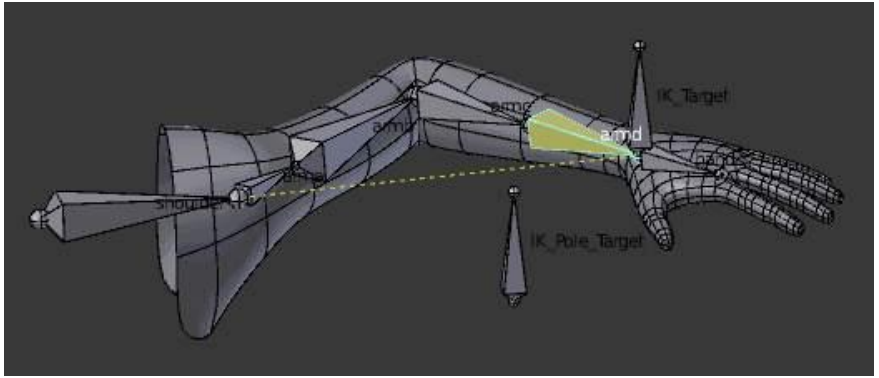
A Real-Time Inverse Kinematics System for Unreal Engine 4

Basic Info

- Repo: https://github.com/hacoo/rtik
- Maintainer: Henry Cooney (hacoo36@gmail.com)
- License: MIT
- Free / Open Source

IK: The problem

- Characters (robots, humans) are often modeled as a rigid-bone hierarchy
- Forward kinematics: the root moves, where does the effector go?
- Inverse kinematics: the effector ‘moves’, how do we get it there?



IK: The problem

- IK is used extensively in animation.
- ‘Static’ IK used for creating animations
- ‘Real-Time’ - any time you need to move a limb, but don’t know where
- Doing IK in real time is harder!

IK: Who needs it?

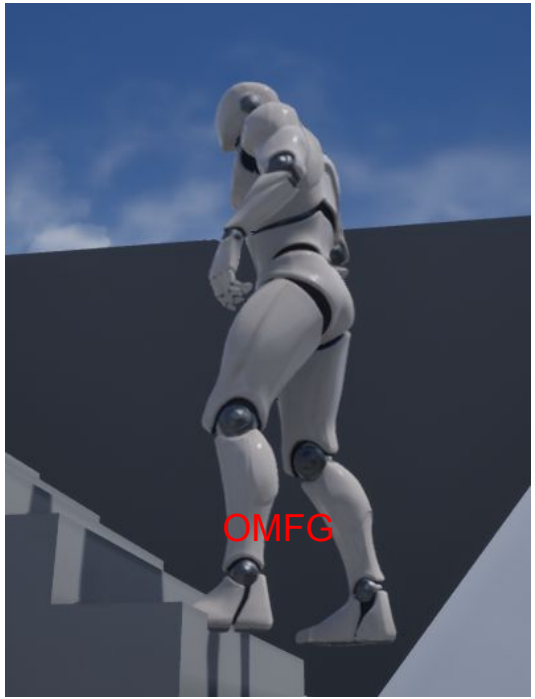
- Almost any game can benefit from IK.
- How much depends on the game type.
  - Great for fighting games, giant robots, climbing / parkouring
  - MMOs, strategy, etc: don’t bother!
- Lots of applications in VR
- Often a necessary layer for generated motion (i.e.: researchers need it)

IK: The problem

- What it really means is:

IK: The problem

- What it really means is:

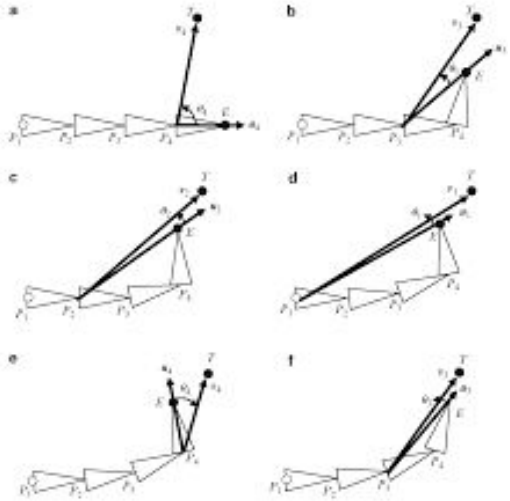


IK: The problem

- What it really means is:

IK: Algorithms

- So, IK is needed when we need to adjust an animation at runtime.
- How do we solve this? With a... solver.
- Many methods exist. I mostly use FABRIK.

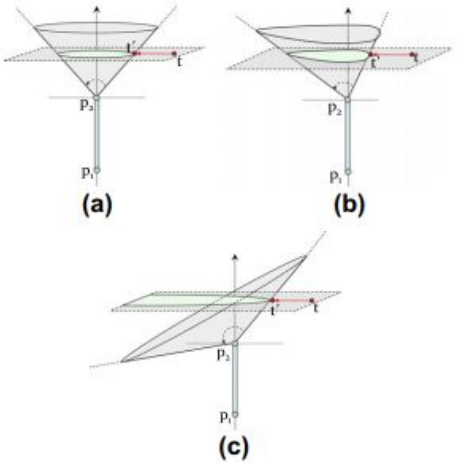


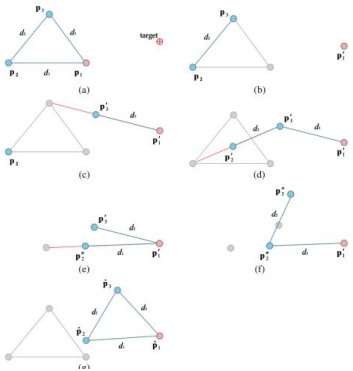
FABRIK

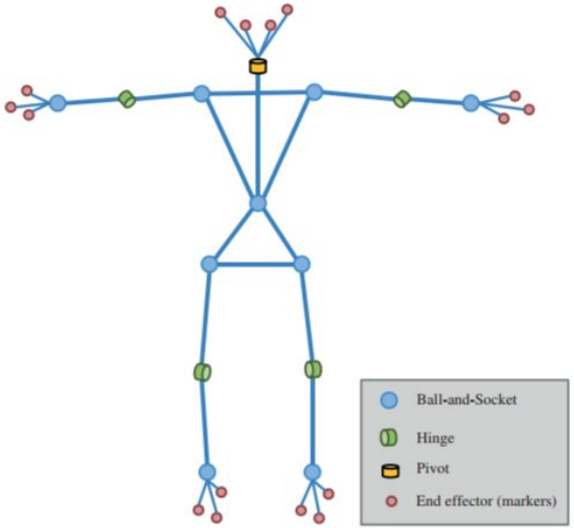
- ‘Forward and Backward Reaching Inverse Kinematics’
- Popular, fast algo with lots of nice properties
- See: http://www.andreasaristidou.com/FABRIK.html

FABRIK

- FABRIK supports ROM constraints (sometimes)
- FABRIK variants: closed-loop, noisy effector
- Can be used hierarchically







Real-time IK

- What's needed in a game
- Adjust a character *while it's animating in real time*
- This is much harder. Solution discontinuities = weird animation
    - A recent high-budget title got some terrible press for this…
- Also it has to be fast!

Existing UE4 Options

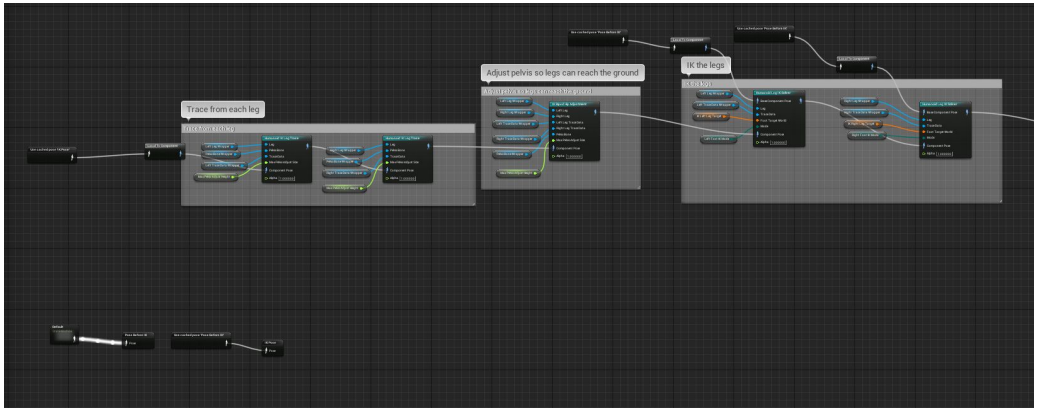
- Included UE4 solvers
    - … But in real time, solvers are the easy part.
    - Epic has their way of doing it, not good for all game types
- IKinema - can be very expensive!
- FinalIK -- oh wait, that’s Unity :D

What RTIK does

- Provides a system of real-time IK primitives
- Can be used to implement full-body humanoid IK
- Low level / modular
- But doesn’t completely throw you to the wolves
- Low level primitives are stitched into a reusable IK setup
- Extended solver functionality

In practice

- It’s a set of UE4 AnimNodes, written in C++
- Reusable for common setups (i.e., humanoids)
- Modular enough that it can be tweaked for custom characters

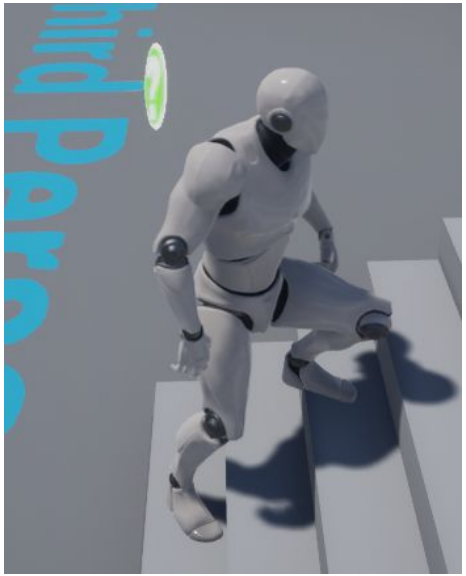


Use model

- Data structures describe chains or other shared data
- Nodes describe operations on these chains (in a pipeline)
- Hopefully, you won’t need to change your pipeline. Just the chains!

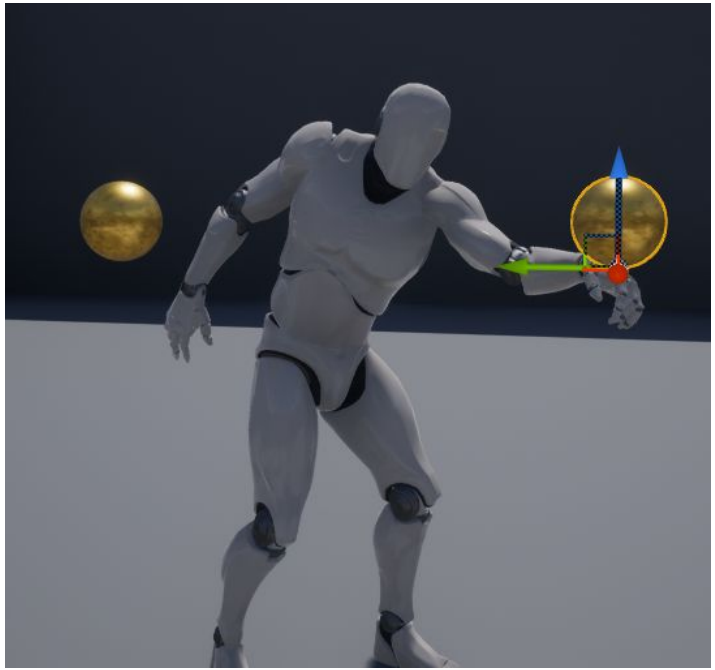
Results

- Very nice IK for the legs:



Results

- Upper body IK works too:



Results

- Performance is good, less than 200us per character.
- Sharing data really paid off. Why don’t AnimBPs support this?

Other lessons...

- Modularity is good. If only AnimBPs supported it better....
- Let the animation tell you what to do!
- Did I mention the thing about sharing data in AnimBPs?

Future Directions

- So many improvements could be made!
- Better torso bending
- Non-human IK
- Ease of use
- Tooling
- VR setup