

ANIMATION FOUNDATION

WHAT'S NEW!

TRAINING TOOLS

A. O. T. M.

TIPS & TRICKS

SEMINARS

CONSULTING

WHO WE ARE

CONTACT US

COOL LINKS

IK AND FK

IK vs. FK - 2nd in a two part series on IK and FK systems

Getting to the bottom of IK by starting at the bottom

In the last article I described the differences between FK and IK. As you may remember Forward Kinematics, is the direct rotation of the bones in the skeleton system. In Inverse Kinematics, motion is inherited in the inverse direction up the system, hence the term "Inverse." We create angles between the bones of skeletal systems by setting key frames on IK handles or on an upper portion of the hierarchy. The solution is provided by an Inverse Kinematics Solver that is built into most 3D software packages. But what the heck is an Inverse Kinematics (IK) solver?

IK by the numbers

If you're anything like me, math has not always been your strongest subject. So how in the world am I going to understand how an IK solver works if I can't even balance my checkbook? Well after further investigation it is relatively simple and gives me a better understanding of how the software works when I am animating or setting up a character. An IK solver is an algorithm* that the software uses to calculate information you are providing every time you move certain parts of a joint system.

Let's dissect what the computer knows when you build an IK skeleton. In all of our examples we will be talking about a two joint skeleton, but IK will work on more than two joints. IK works because we know the length of the joints (in our example the joints are green.) We also know the two positions in space represented as the top of the skeleton and the goal or IK handle. The Blue null is the top of the chain and the red null is the IK handle With these four positions we can obtain a value of the angle in between the two joints by running an IK algorithm. So when I move my red null (the effector in soft and maya) the blue null (skeleton root) remains in place and the distance between them gets larger or smaller. Please take a look a [Movie1](#) for reference. Changes in this distance inform the solver how to calculate the angle in between the two joints. That's it, almost automatically we have one of the most effective ways to create realistic motion in our characters.

*al•go•rithm (lg-rthm)

n. Mathematics A step-by-step problem-solving procedure, especially an established, recursive computational procedure for solving a problem in a finite number of steps.

1 | 2 |

[Next](#) | **Back**

[What's new](#) | [Training](#) | [AOTM](#) | [Tips](#) | [Seminars](#) |
[Consulting](#) | [Who we are](#) | [Contact Us](#) | [Cool Links](#) |
[Home](#)