Autodesk®
HumanIK®
4.5

Using Autodesk HumanIK Middleware to Enhance Character Animation for Games

Unlock your potential for creating more believable characters and more engaging, innovative gameplay with Autodesk® HumanIK® 4.5 middleware. The software's full-body inverse kinematics (FBIK) system enables characters to interact dynamically and more realistically with their environment and with other characters. With the capability to dynamically layer FBIK information onto existing clips at runtime, game developers can create, modify, and reuse character animations, helping save valuable animator time. Streamlined workflows between HumanIK, Autodesk® Maya® 2011 software, and the Epic Unreal Engine® 3 game engine help accelerate the entire process of creating top-quality games.

Autodesk®

www.autodesk.com/humanik

Contents

Top Reasons for Using Autodesk HumanIK Middleware

Create Character Animations Faster

Animators work to create animations that bring their character to life and tell a story. Using HumanIK helps free animators to concentrate on performance rather than all of the interstitial animations for each reach or skeleton possibility. Using HumanIK for full-body inverse kinematics (FBIK), and retargeting for dynamic alteration and construction of character animations enables animators to use the same animation across different environments or different characters.

Enhance Gameplay with Realistic Animation

Autodesk HumanIK increases character believability while preserving creative intent. HumanIK creates the inverse kinematics (IK) solution modifying the animation on top of the hand-animated or motion-captured FK animation, using a biomechanical model to conserve the underlying look and original work. Characters assume appropriate stances on uneven surfaces, climb walls, and pick up objects—even within a frequently changing game environment. The result is more believable character animation, supporting wider ranges of game-play options.

Develop Custom Character Behaviors

Game developers can use Autodesk HumanIK middleware to build custom tools for constructing specific artist-defined character behaviors. Creating custom behaviors can help game developers make more convincing characters for a more unique and memorable game experience.

Manage More Animation Data More Easily

Autodesk HumanIK uses procedural motion adaptation, a technology that helps save memory and animator time. HumanIK procedurally adapts existing character animations to game environments at runtime, minimizing the need for large animation clip libraries and helping reduce the amount of clip production performed by animators. Autodesk HumanIK middleware also offers runtime retargeting technology, enabling developers to reuse animation clips on a wide variety of characters, saving time and enhancing creativity.

Integration in the Unreal Engine 3 Game Engine

The Autodesk HumanIK and Epic Unreal teams worked in concert to bring game developers a robust integration of HumanIK into the Epic Unreal Engine® 3 game engine. HumanIK 4.5 middleware has an artist-friendly toolset—putting more control of the runtime results into the hands of the animator. Artists can retain the creative vision for

their characters from initial animation authoring to the game-play experience with HumanIK.

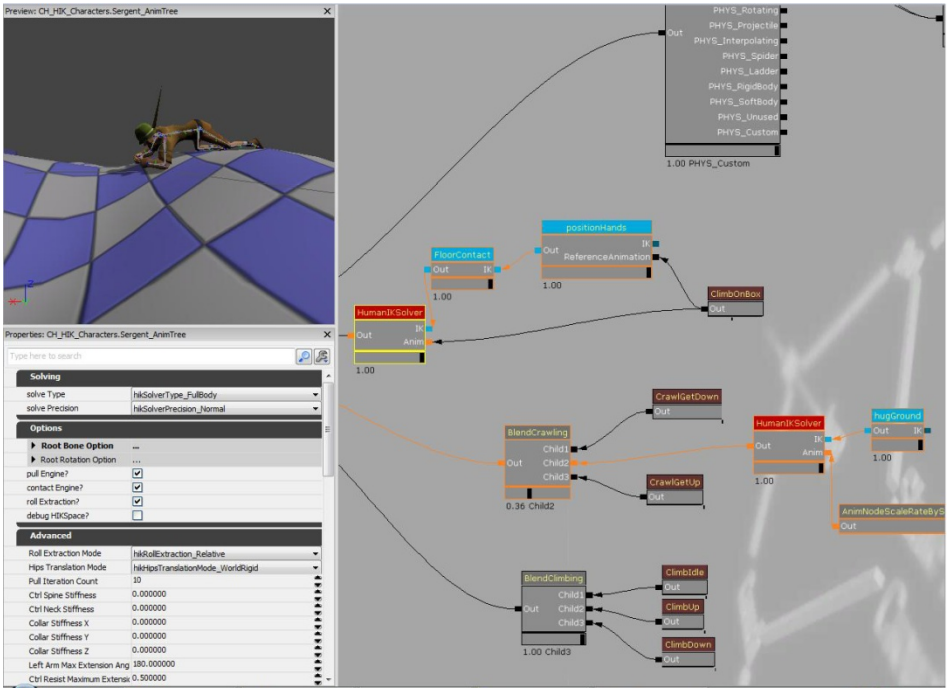


Figure 1. Autodesk HumanIK integration into Unreal Engine 3 game engine.

Artist-Friendly Workflow

Put more power in the hands of your artists with the HumanIK 4.5 characterization tool. Artists can specify bone mappings and validate their work using a friendly graphic interface. Available as a plug-in to Autodesk® Maya® 2011 3D animation software, and as a stand-alone program, the tool helps make character configuration part of an integrated workflow.

Squash and Stretch

HumanIK 4.5 middleware helps bring greater flexibility of animation environment and style to developers with squash and stretch functionality. Whether obtained through motion capture or key framed, character animation becomes more responsive to the real-time situation—characters can adjust their neck, spine, and limbs with squash and stretch. Animators have more power over the activation as well as the ease-in and ease-out settings for greater control in creating more believable characters.

Overview of Autodesk HumanIK Middleware

Autodesk HumanIK animation middleware is a runtime solution for the creation of character animation for games. HumanIK enables more believable character interaction and procedural adaptation of animation to game environments. The result is reduced game-play constraints and much more believable 3D character animation experiences.

Inverse Kinematics

Most animations are created using forward kinematics (FK) to calculate the positions of a character's body parts. FK animations are highly efficient in terms of the demands on a game console processor, but are generally less realistic looking and much less adaptable to changes in the game environment. Inverse kinematics (IK) algorithms help overcome these limitations but are computationally intensive, and much harder to code. Autodesk HumanIK animation middleware provides a library of optimized IK solvers, making it faster and easier to apply IK and enhance animations.

There are several ways to apply IK to a character animation. One of the most common ways is through two-bone IK, typically used for simple body chains, such as the upper and lower leg or arm.

Unlike two-bone IK, HumanIK performs full-body inverse kinematics (FBIK) to help manipulate multibone IK chains. For example, a character can reach for a doorknob using its arm, but aided by realistic movements from the shoulder, torso, pelvis, and legs. The result is more realistic movement and a more believable final pose.

Calculation of FBIK depends on "effectors"—points in 3D space, each with a goal for a certain node or set of nodes in a character's skeletal structure. The HumanIK IK solver attempts to place each node at the location of its corresponding effector. The solver takes into account the constraints placed on each effector (its reach, pull, and resist values), and the movement limitations of each node (such as its degrees of freedom and roll).

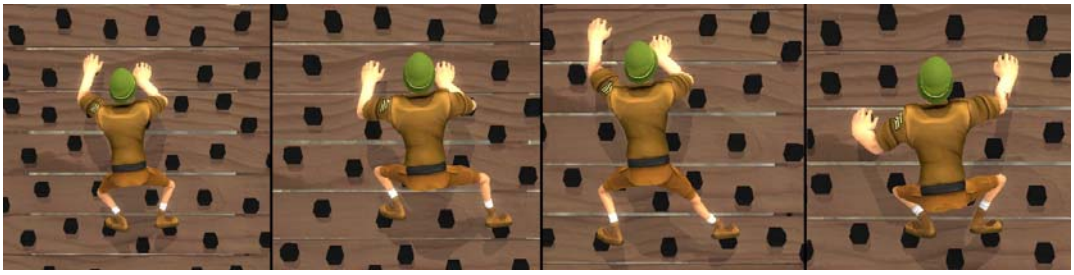


Figure 2. The Autodesk HumanIK middleware library uses procedural motion adaptation to "adapt" a character's position at any given frame. In this example, a soldier climbs a wall with arbitrarily placed footholds. A single climbing animation is created and applied. The HumanIK middleware library procedurally adapts the character's limb positions to realistically grab the footholds, resulting in a more believable climbing animation.

Autodesk HumanIK in Action

In many games, characters appear to float on a surface because the character's animation does not take into consideration variation in the ground plane or environment for foot contact. If such a character needs to walk on uneven terrain, or to run up a staircase, the resulting dissociative and unrealistic animation disrupts the aesthetic experience.

With Autodesk HumanIK, the character's feet are placed on the ground. As terrain changes, the feet dynamically and correctly position themselves. HumanIK uses IK solvers to help adapt the character's position and movements at runtime.

Figure 3. Character foot contact with ground planes with HumanIK off (penetrating through plane) and on (conforming to plane).

IK solvers can help modify many types of character movement. Consider a character crawling across a terrain. Ordinarily, an artist would have to produce many animations to ensure correct foot and hand placement, and specific animations for objects like rocks. HumanIK, on the other hand, uses procedural motion adaptation to reposition the character's limbs for the different situations. This adaptability results in more interesting environments and varied designs, as characters are no longer constrained by keyframed animations.

Figure 4. One crawl animation retargets to varied environments for greater animation reuse, lower memory footprint, and increased flexibility.

HumanIK uses a lightweight, runtime, full-body IK rig, resulting in a more believable animation and look. For example, if a character reaches toward something far away, it does not reach with just its arm as in two-bone IK. The whole body will bend to reach the goal.

Autodesk HumanIK can combine procedural effects. A basic walk cycle can include a “look at” HumanIK controller, which causes the character’s head to turn and look at an object as it flies past. Similarly, a “dodge” movement will instruct the character to look at an oncoming object and move out of its way. This layering offers developers more freedom to create complex and realistic interactive character animations at runtime.

The HumanIK middleware retargeting feature enables developers to take the same animation data, together with procedural effects, and retarget them onto a different rig.

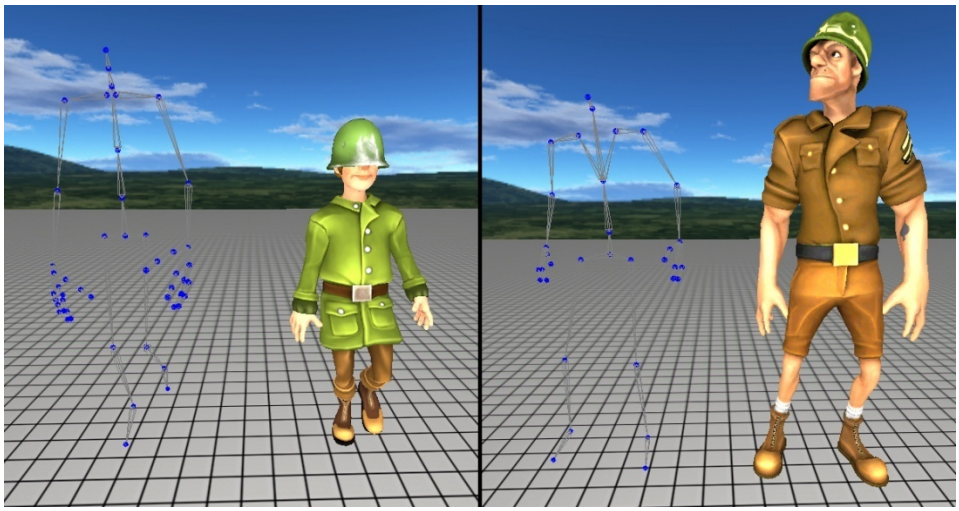


Figure 5. Character animation from the same skeleton retargets onto characters with dramatically different proportions.

By retargeting, artists can use the same animation data for two or more characters, enhancing efficiency in the development of multi-character games.

The Autodesk HumanIK Application Programming Interface

HumanIK is a multiplatform API (application programming interface) in C++. It is delivered as an SDK (software development kit) comprising:

- HumanIK libraries
- Header files for the HumanIK API
- HumanIK samples
- Autodesk and certain third-party libraries used in the HumanIK samples
- Animations, character data, and scenes used in the HumanIK samples
- HumanIK documentation

HumanIK takes FK information provided by the game engine and runs it through IK and retargeting solvers as needed, to recompute the appropriate FK positioning of the character.

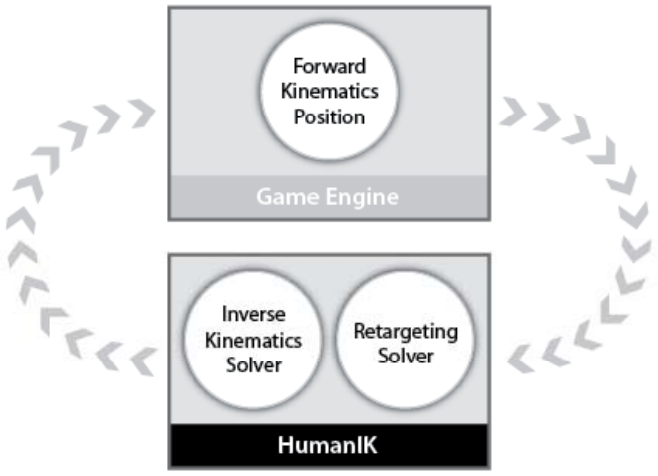


Figure 6. Autodesk HumanIK and the Game Engine.

Autodesk HumanIK Characters

Autodesk HumanIK middleware acts on characters. Any character controlled by HumanIK solvers is represented by an HIK Character, a two- or four-limbed form specific to HumanIK.

HIK Character contains a set of joints, also known as nodes, corresponding to specific parts of a skeleton. One type of node represents the elbow of the left arm, another represents the wrist of the left arm, another the first joint of the thumb, and so on. HumanIK supports at least 170 predefined joint types.

To create an HIK Character from a Character Definition, the user can use the characterization tool pictured below, which is integrated into Maya and Unreal and available as a stand-alone tool. Animators can specify bone mappings and validate their work using the user-friendly graphic interface. If the team prefers to do this step by hand, they can map the character's skeletal components to HumanIK variables by providing HumanIK with the placement of each of the character's joints in a neutral "T" stance. Each of the character's nodes is then set to an appropriate translation and orientation through calls to functions provided in the HumanIK API.

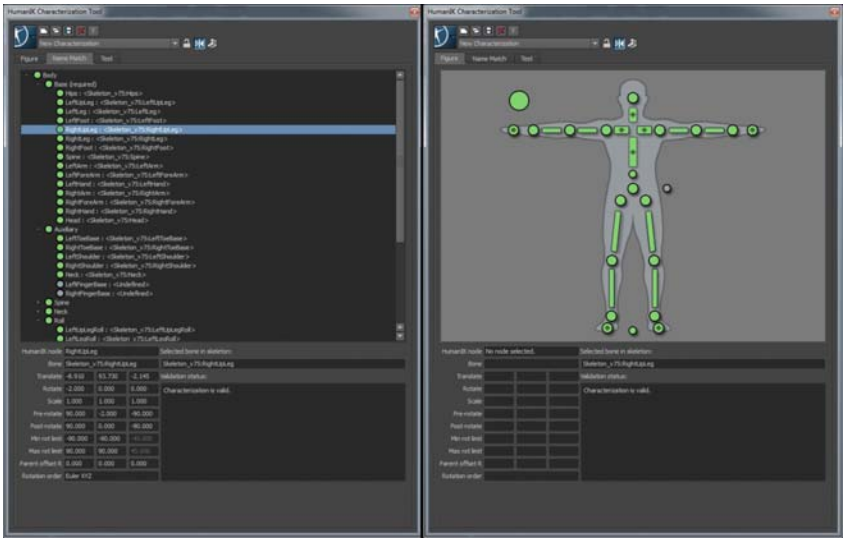


Figure 7. The characterization tool in HumanIK is an artist-friendly interface for specifying bone mappings and validating work.

Character States

Every HIK Character node has defined positions and rotations for different events. Taken together, all positions and rotations define a Character State, which is a specific stance or pose for the character.

When the HumanIK solver is used, an initial pose of the source character's skeleton in a Character State is provided, based on the status of the character in the game engine.

Once the HumanIK IK or retargeting solver has generated an optimized pose for a skeleton, the new target position and rotation of each node to the Character State is saved, and can then be retrieved and reapplied to the character model or animation in the game engine.

Effector Set States

Effectors define the desired position and rotation for different parts of the character body. They are used in sets, each of which defines a coherent set of goal points for the character's limbs at a given moment in time. An Effector Set State to the IK solver calculates the translation and orientation for relevant parts of the skeleton.

For example, if an animator moves the effector for the tip of the left index finger toward the character's wrist, the IK solver moves the other nodes in the character's hand and finger to place the finger in its desired position. If the index finger has multiple nodes, the finger will curl.

Each effector in an Effector Set State can be used to place other special constraints on the solver, such as Reach, Pull, or Resist.

Property Set States

Property Set States contain parameters for configuring the IK and retargeting processes. These can include values such as the stiffness of a character's spine or neck, the maximum extension angle of an arm, or the amount of compensation applied when retargeting to account for differences in scale between the two characters.

How Autodesk HumanIK Can Solve Common Problems in Game Character Production

The following scenarios highlight common character-based production problems, which HumanIK helps solve.

Foot and Hand Contact

HumanIK has a sophisticated floor contact engine that helps keep character's feet, toes, hands, and fingers from passing through oriented planes, floors, walls, or ramps.

For example, in the following image, the character's hands and legs pass through the surface of the ground. When foot and hand contact is enabled for the HumanIK IK solver, the character's hands and feet conform to the surface of the ground. HumanIK automatically adjusts other joints for a natural pose.

Figure 8. Character animation adaption with and without Autodesk HumanIK.

Character-to-Character Interaction

Character-to-character interaction is typically complex, because of the arbitrary placement of the characters and how they might interact. HumanIK is able to help solve this through procedural motion adaptation, enabling characters to interact with each other dynamically at runtime.

Character-and-Prop Interaction

HumanIK can give users finer control over the details of a character-and-prop animation. For example, animators can use HumanIK to help lock a character's grip to a rifle. Moving the rifle causes the character's body to follow in a more natural fashion. If the rifle must be moved to extreme positions, such as pointing directly overhead, one can partially break the HumanIK links to allow a more natural movement.

HumanIK can even be used for interaction in a dynamic game environment, such as adapting a character's feet to a ledge that changes angles.

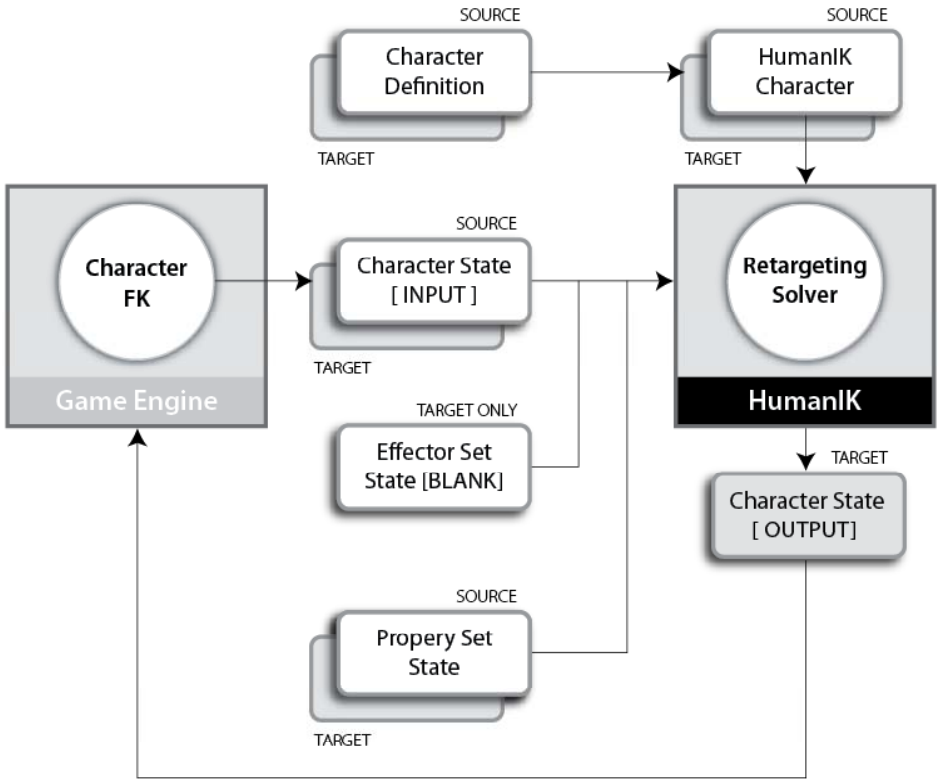
Dataflow Using Autodesk HumanIK

HumanIK Retargeting Dataflow

Retargeting is the process of reusing animations designed for one character on another character with different proportion, scale, skeletal structure, limits of motion, and level of detail.

A game production team, for example, uses motion capture to record human movements and can then "retarget" the animation from the motion capture onto a dwarf character.

The following diagram shows the HumanIK retargeting dataflow:



Character FK to Game Engine

The inputs required by the HumanIK retargeting solver are:

1. Two HumanIK characters representing the source and target characters. The characterization tool available in Autodesk Maya, Epic Unreal Engine 3, and as a stand-alone application is a user-friendly tool for creating and testing the characterization.
2. Two Character States that contain the current FK position and rotation of the source and target characters' nodes (from the game engine).
3. A blank Effector Set State that the retargeting solver sets up internally and uses to call the IK solver on the target character.
4. Two Property Set States that provide a variety of parameters and configuration settings for the source and target characters.

The Retargeting Solver calculates the resulting pose for the target character, and stores the information in the same Character State input.

Animators can apply an IK solver pass to help improve or correct a retargeted animation. Take, for example, an animation created for a giant character grabbing and biting an apple. The same animation retargeted to a dwarf might have the apple hit his forehead. A simple IK solve helps make sure the apple meets the dwarf's mouth, without the need to rework the animation.

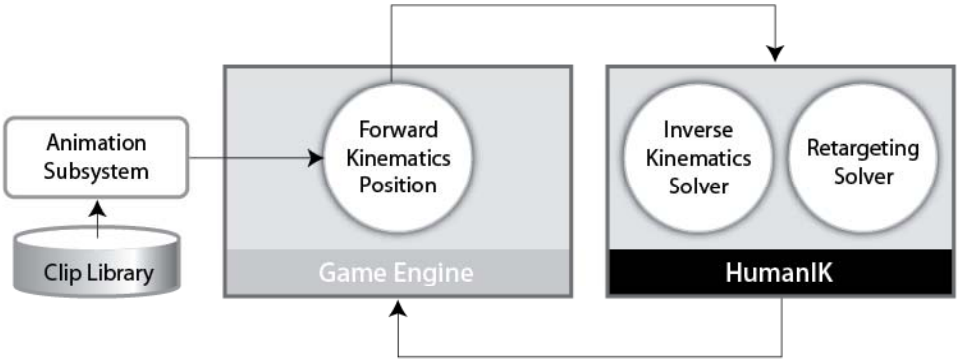
Benefits of Animation Retargeting

- Enables users to create and share pools of animations between characters, greatly improving efficiency in developing multicharacter games

- Useful during the prototyping phase, where one can reuse animations from other projects until the final animation is created

How Autodesk HumanIK Works with the Game Engine

HumanIK is not a self-contained animation engine; instead, it works with an existing animation subsystem, acting as an additional layer that helps modify and reuse existing animation clips. Custom behaviors created using HumanIK can be saved and reused—it is code you can build on.



The typical data flow between the game engine and HumanIK is as follows:

1. HumanIK receives the position and rotation of the joints of one or more characters in the game. These positions often originate from FK animations provided by the animation subsystem, but may also come from other subsystems such as a ragdoll physics engine.
2. HumanIK receives the effector settings from an artificial intelligence (AI) or other system.
3. The IK and retargeting solvers help modify the position and rotation of the character's joints within HumanIK.
4. The engine retrieves the generated positions and rotations, and sets the game character's position.

The HumanIK solvers use the FK positions of the game characters as input, and create new FK positions for those characters as output. This approach has two important ramifications:

Developers can choose to use or skip the HumanIK solvers depending on the current situation of the game. HumanIK solving can also be called on as many times as necessary, blending together multiple layers of IK adjustments. Users can create modular

units of IK adjustments reflecting semantic elements of the movements involved and reuse those units in different circumstances.[1]

For example, one IK pass can make a character reach out to touch a target object, while another IK pass can make the character swivel its head, neck, and chest to look at the target object. The controller can then be reused for other purposes: having the character track a plane flying across the sky, or tracking the movements of the player character across the screen while shooting a gun.

How Autodesk HumanIK Fits into a Production Workflow

HumanIK character definitions can be created with Autodesk Maya and Autodesk® MotionBuilder® real-time character animation software. Autodesk MotionBuilder has HumanIK integrated into its animation engine, which means that characters animated in MotionBuilder are a more effective way to preview what the animation may look like in the game engine.

Commercial Game Engine Integration

HumanIK is integrated into Epic Unreal Engine 3 along with the characterization tool, making it easier to get a prototype running, and putting more runtime control into the hands of the artists. Teams using HumanIK, Epic Unreal Engine 3, and the Autodesk® FBX® asset exchange technology will find even more efficiencies with the tight integration work done by the Epic and Autodesk teams.

HumanIK is a highly modular, C++ library that can be integrated into a game engine to complement existing animation code. While it is not multithreaded, it can be used in a multithreaded environment.

HumanIK is integrated into the Trinigy Vision™ game engine, helping deliver greater efficiency to teams using both products.

[1] It is possible to determine which body parts are affected by an IK solve. Most cases can be solved in one pass. Additional passes are necessary when an IK goal depends on the result of a solve (for example, dodging a bullet while reaching for the holster).

Appendix

Specifications

Platforms
HumanIK is available for use on the following platforms:

- Microsoft® Windows® (32- and 64-bit) operating system
- Microsoft XBox 360® game console
- Sony® PLAYSTATION®3 game console
- Sony PlayStation Portable (PSP®) handheld
- Nintendo® Wii™ console
- Linux® (64-bit) operating system
- Apple® Mac OS® X (32- and 64-bit) operating system

Please contact middleware@autodesk.com if you are interested in using HumanIK on other platforms.

Third-Party Tools
HumanIK works with certain third-party solvers and middleware components: physics and simulation engines.

Verbose Library
A verbose library is available for Windows platforms.

HumanIK Terms

Character States
An object that defines the translation and rotation values for the character's nodes at a given moment in time and, therefore, a single pose or stance for that character.

Effector
Effectors are points in 3D space, each providing a goal for a certain node or set of nodes in a character's skeletal structure. The HumanIK IK solver attempts to place each node at the location of its corresponding effector, taking into account the constraints placed on each effector (its Reach, Pull, and Resist values), and the movement limitations of each node (such as its degrees of freedom and roll).

Property Set State
An object that stores values for a variety of customizable parameters controlling the way the HumanIK solvers manipulate a given character.

Pull
The Pull value of an effector determines the priority of that effector relative to others acting on the character. Setting a high Pull value for an effector will make the character appear to lean toward that effector with its entire body, in addition to the node corresponding to that effector.

Reach
The Reach value of an effector determines the relative weight given to the position and rotation of the corresponding node in the original character state used as the input for the IK solver (typically a position derived from an FK source), and the position and rotation of the effector.

Resist
The Resist (Stiffness) value of an effector determines how likely the corresponding node is to maintain its original angular rotation when the pull of another effector acts upon that node.