The Gamebryo Animation Tool is a feature-rich tool that may take some time to get familiar with. Several tutorials are provided here to help speed that process along. These tutorials are meant to be read sequentially. They cover things such as the basics of working in the tool environment and information about how to add and preview sequences, transitions, and sequence groups. In-depth analysis is not provided here; instead, a step-by-step approach is used, along with many screen shots, that can be easily followed.

The following tutorials will help you familiarize yourself with the Gamebryo Animation Tool:

- Tutorial 0: Getting To Know the Tool Environment
- Tutorial 1: Adding and Previewing Sequences
- Tutorial 2: Working with Transitions
- Tutorial 3: Using the Interactive Preview
- Tutorial 4: Sequence Groups

There are several how-to's that are more targeted than the tutorials and are designed to address commonly asked questions.

- How-To 1: Changing Sequence ID's
- How-To 2: Mixing Characters and Sequences

http://www.gamebryo101.com/learn

http://www.gamebryo101.com/tv?task=viewvideo&video_id=147
Homework Ten

Perhaps run this in conjunction with

CharacterAnimationDemo Overview

The CharacterAnimationDemo application demonstrates many of the advanced features of the Gamebryo animation system, including the NiActorManager animation controller and the NiSkinningLODController character level-of-detail controller. The character used in the sample is a mother Lenguin, "Momma".

The NiActorManager class allows the application to easily control the many artist generated animation sequences available in Momma. The application simply needs to provide the actor manager with a single "event code" that tells the manager what sequence to use during a given animation. The NiActorManager class handles all the complexities of animation blending and transitioning.

NiActorManager also provides a callback system for applications to manage animation sequence events. In the CharacterAnimationDemo, the walk and run animation sequences were tagged with "text keys" to identify when in those sequences Momma's feet strike the ground. CharacterAnimationDemo implements a very basic footprint mechanism to demonstrate this functionality — a left or right footprint becomes visible when these events occur. When one footprint is made visible, the other disappears. (Note: we recommend something more advanced and artistic in terms of application behavior for games that want to have their characters leave footprints behind them).

NiSkinningLODController gives the application control over the number of bones that are used to deform Momma's skin. BoneLOD allows an application to dynamically adjust how many bones in a skeleton affect its skin. It allows both the polygon count of the mesh and the number of bones to be swapped out at run-time by switching between levels of detail.

When the application starts, Momma is in front of the camera playing her various idle animations. The currently active idle animation is chosen randomly by some simple logic in the application code. The application code works with NiActorManager to determine when a new random idle animation should be chosen. Momma can also transition to walk or run animations based on user input. When Momma walks or runs, she "drops" footprints on the ground.

Details forthcoming.