A New Approach to Procedural Character Rigs

Erik Malvarez  
Senior Character TD  
Blue Sky Studios

Scotty Sharp  
Senior Character TD  
Blue Sky Studios

Stephen Unterfranz  
Supervising Character TD  
Blue Sky Studios

1 Introduction
For 'Horton hears a Who' the character technical directors at Blue Sky Studios developed a new approach to procedural character rigs. Many rigging systems reduce or eliminate redundant work by importing pre-existing components into a character scene or using scripts to generate components or entire rigs; our system goes several steps further. Essentially an object oriented approach, it allows the artist to assemble the template for a character rig independently of the character itself. The template exists as a network of nodes that defines each component's attributes and functionality, identifies argument joints and surfaces, and represents the components' relationships to each other and to their arguments. These highly modular networks offer consistency, robustness, and flexibility.

2 Evolution of the Blue Sky Character Rig
On Blue Sky's second film Robots, the rigs were relatively simple, providing the animators with 236 keyable objects and 1,759 keyable attributes on the lead character Rodney. For Ice Age: The Meltdown, several common rig components were identified and automated, allowing for faster creation of more complex rigs; however, these were prone to varying from character to character as they were refined, and difficult to customize or update once installed. For Horton Hears a Who, senior members of the department identified early the huge demands of aesthetic performance, as well as a much larger character count. This meant that the rig components themselves would need to provide more functionality (the Mayor of Whoville has 804 keyable objects and 7,042 keyable attributes!) as well as facilitate efficient and frequent updates, allowing the artists to concentrate on appealing deformation.

One key aspect of the new system would be to separate the aesthetics of the deforming model from the objects, attributes, and underlying structure the animators would use to create the performance, allowing parallel development of the deformation skeleton and the control rig. Artists would be able to refine shaping of the character independently from the network of components that would ultimately drive that deformation.

3 The Concept and Structure
The system is designed to provide character TDs the ability to set up the control rig of a character efficiently and consistently. The control rig is made up of a variety of rig components (rc's) - previously prototyped MEL scripts that construct all the groups, constraints, ik handles, etc. required to deliver specific animation control over a region of a character. A character rig begins as a Maya skeleton and a referenced model file containing surfaces deformed by a skin cluster and corrective blend shapes. The system requires that the deformation skeleton follow strict hierarchical, labeling, and naming conventions.

Rig components are represented in three forms in the system. First, as the basic command line form of the rc MEL script, requiring input arguments (typically, a list of joints in the deformation skeleton and a set of optimization and behavior options); this will install a fully functional rig component in the current scene. This component is said to exist in the “land of the living”. Second, as a script node (rcSN) that contains argument attributes to provide the rc script for installation; the script node may exist in the scene without the component itself if it has been created, but not installed. This node is said to exist in the “land of the dead”. Third, as the component user interface (rcUI), which is parented to the rc manager; the rc manager is the top level interface for managing all components currently created in a scene (installed or uninstalled).

Additional rcSNs were created to identify and connect argument transforms or attributes, as well as package groups of nodes to organize the scene file. Combined in networks called character components, the rcSNs provide object oriented assembly of a character rig. In a rig's dormant state, all the components and their relation to each other may be represented by as few as 500 nodes; these modular networks contain all the information required to call the most current rc scripts, creating extremely complex character rigs in about three minutes. The system foundation is a library of generic, common usage procedures (including commands to populate sets, modify attributes, assemble hierarchies, create and remove script nodes, install and uninstall components, and build user interfaces), updates to which can be propagated throughout a character component the next time it is installed.

4 Advantages of the New System
The most immediately apparent advantage of procedural rigging is consistency. Shared libraries and a singular interface allow each technical director to avoid redundant or divergent work. In preproduction, an established criteria for continuity ensures that a component is not being developed in isolation, but rather is following a clear set of guidelines. During production, assembled networks of rig components can be shared across characters, reducing set up schedules and allowing for transfer of animation from one character to another of similar anatomical structure.

The next and perhaps most compelling advantage of procedural rigging is robustness. As animation preferences alter with the requirements of the picture, additions to a particular rig component can be rapidly disseminated. This also creates an environment in which improvements to rig functionality are cumulative across pictures; for example, one component will become more customizable as features are added without sacrificing any of its original installation options. Combinations of components are also possible, meaning complexity can be added or removed in layers.

The final and most exciting advantage is flexibility. The procedural rigging system provides insulation from proportional design alterations. A procedural network can be uninstalled, the skeleton modified to match the new form, and reinstalled at no cost. This also allows for automated builds on multiple characters; one character may be the generator for many similar variations. Finally, with the correct preparations made during the character evaluation process, shot-by-shot customization by the animator is now possible. A rig component can be added or removed depending on the demands of a particular performance.