Contents

[Introduction: What is Modular Procedural Rigging and Animation? 1](#_Toc307492221)

[Increased Efficiency in Turbine’s Character Pipeline 1](#_Toc307492222)

[MPR&A Sounds Great, so how does it work? 2](#_Toc307492223)

[A More In-Depth Look at How the System Functions. 3](#_Toc307492224)

[In closing 3](#_Toc307492225)

# Introduction: What is Modular Procedural Rigging and Animation?

Procedural rigging means that rigs are generated through a scripted procedure. At first, an aspiring character TD is taught to create effective character rigs by hand using the built in tools provided by a software package such as Autodesk’s Maya. Once the TD gains some experience and learns what works and what does not, it becomes much more efficient to script the process of generating rig components. I solid rig can take forty hours or more to generate. By scripting the rig build process, a week’s work can be done in seconds.

The downfall of a procedural system is that there is very little wiggle room. When a TD creates a rig by hand, each component can be tailored to meet the needs of the character being rigged. In one case it may be useful to have an IK spine, where another shot may call for FK. This is where modularity comes into play. Rather than having one push button solution that builds an entire rig, we partition the rig build into several components or modules, that can be added as needed.

# Increased Efficiency in Turbine’s Character Pipeline

Turbine relies on manually generated rigs for use in animation. This presents a number of problems which I will outline here.

* As mentioned earlier, it takes a tremendous amount of time to create a solid rig by hand. Consider the hundreds of characters populating our games and multiply the average of forty hours per new character rig. That is a huge allocation of hours that could be spent creating new tools or content. Granted, it will take will take time to produce a fully functional. MPR&A system, however the long term savings far exceed the initial cost. A working prototype has already been developed at the cost of approximately five-hundred hours. Based off previous calculations that is equal to about twelve new character rigs.
* It is imperative that we be able to stay competitive in the market. Several studios have already seen the benefit in adopting a more advanced character pipeline. A prime example is the system developed at Bungie for use in the popular Halo series. The reason Bungie made the decision to develop this tool set is simply stated in this quote. “These techniques allow riggers to quickly iterate on character development while neighboring departments continue to work at full speed”. That brings us to the next point.

# MPR&A Sounds Great, so how does it work?

In this section I will provide an outline off the MPR&A system and its desired functionality. In a later section I will describe a more technical outline of the tools proposed structure.

The ideal character rigging and animation pipeline would include the following.

* An entry point for character artists. This means that a modeler could apply rigging components while modeling. This would assist in constructing geometry that can provide optimal deformation for animation. This also opens the door for allowing character artists to bind and weight the mesh.
* An automated skin weighting solution. I don’t know of any weighting solution that gets a character to100% complete, but I think we could hit 80%.
* The ability to easily set up a new rig. This means we need a simple yet effective system for determining joint placement as well as tools for saving out templates that could expedite future character setup.
* The ability to handle any kind of creature. This tool should allow for the rigging of bipeds, quadrupeds, dragons, cars, doors, or anything imaginable.
* The ability to add and remove rig components on a per animation basis. If the animator needs IK on a limb, then IK can be added on the fly. If a new animation module is required, then a TD could create that module, thus making it available for all future and present animations.
* An easy way to integrate motion capture or re-targeted animation data. A pose/animation library could hold limitless animations that could be applied to a new character. An animator could then add rigging components as needed to edit the performance.
* A breadth of tools to assist the animator. This includes space switching, animation mirroring, arc tracking, and blending between controls.
* A tracking system that provides easily accessible data such as bone count, vertex count, and usage data.

The MPR&A system will provide a series of blueprint modules that represent a variety of joint setups. For example the system would include hinge joints, single joint chains, splines and more. These modules could be further combined to create hands, legs, or entire biped templates. When the modules are locked and published as a character, the user ends up with a well organized “smart” joint system. Each joint will have recorded data describing its usage. The data will include information such as what type of animation modules can be applied.

An animator can now load this published character and begin applying animation modules. Animation modules can be applied or removed at any time during the animation process. New animation modules can be developed by a TD in a non-destructive manner.

* Iteration can greatly reduce the scope of a game or even cause it to flop. It is the prerogative of the art lead and game designer to ensure that the highest quality and aesthetic value is achieved. The character pipeline should be designed to facilitate the needs of the product while allowing the team to continue work during the iteration process. Imagine this scenario. “*A new character comes down from the art department and is sent to rigging, where a character TD must create a rig by hand. This rig is then passed off to animation for the creation of an animation set that may consist of some fifty animations. After a week of work the animator comes to the forth animation on his list only to realize that the rig is not capable of achieving the desired performance. At this point the rig must be altered to accommodate the conditions or the animator must sacrifice quality in the interest of saving time. Assuming the rig change is approved, a new problem has now been created. The change is destructive to the animations which have already been painstakingly created. Now let’s assume that the worst is behind us. The animation set is complete and in game, but now the art director detects a fatal flaw that can only be remedied by completely redesigning the character. So now we have a new character, with new proportions, and the animation set no longer works. Now the whole process must begin again.*” This is an extreme case, but it does happen and at great cost. So how do we remedy this problem? A more advanced pipeline holds many of the answers.
* Hand built rigs means hand crafted errors and inconsistency. Preferred naming conventions change between TD’s, spelling errors occur, and a host of inconsistencies creep in when hundreds of rigs are created by hand. A consistent rig removes the learning curve from the end user and allows pipeline specific tools to be easily created. Animations are more easily shared, behavior projects have a common bond, skin weighting is easily transferred between characters, and the art department is just happier in general.

# A More In-Depth Look at How the System Functions.

The MPR&A system will rely on a unified studio MAYA API. The API will hold a multitude of class objects that can be called upon to build out the functionality of the MPR&A tool as well as a number of other tools. In this way the MPR&A system is using a unified code base thus ensuring standardization and easy integration with other studio tools.

The MPR&A system is divided into two primary UI windows. The first UI contains all of the modules need to build out the character setup. The setup modules are IK driven, which allows for easy placement in 3D space. Module selection triggers a UI event that populates the UI with any associated preferences such as joint orientation, and naming. Modules can be “hooked” together to define the intended joint hierarchy. Modules can be duplicated, mirrored and moved in a mirrored mode to ensure symmetry. The UI also provides the ability to save and load templates for later attributes within easy reach of the animator. This includes custom attributes such as stretch, twist, and icon scale.

# In closing

The MPR&A tool promises to reduce production costs while providing an integrated solution for rigging and animation. The modular nature of the system reduces iteration times, and allows more flexibility in meeting the needs of the end user. The tight integration with a unified MAYA API will expand the possibilities for tools development while allowing the integration of tracking, management, version control, and other key systems.

**References**

[http://www.bungie.net/inside/publications.aspx](http://www.google.com/url?q=http%3A%2F%2Fwww.bungie.net%2Finside%2Fpublications.aspx&sa=D&sntz=1&usg=AFQjCNEEbrAISJ44KhN72TF1WCKUTx0Trg)