Beyond Procedurally Modeled Foliage in *Madagascar: Escape 2 Africa*

Jeffrey Budsberg
DreamWorks Animation

Scott Peterson
PDI/DreamWorks

1 Abstract

Developing vegetation to capture a very distinct and stylized art direction is a great challenge in computer cinematography. Completely procedural approaches constrain artists to model foliage by connecting nodes, modifying numbers, or writing code. On the other hand, manually placing thousands of branches is completely infeasible. Attempts at hybrid techniques have been inadequate, as they are still too biased towards procedural systems.

*Bonsai*, our tree modeling tool for *Madagascar 2*, yields dramatic improvement in stylistic control over previous methods. Modelers use familiar modeling tools to create and connect small branch parts to form more complex networks, which can be connected or dynamically grown into a large hierarchical system of branches. More importantly, this allows the user to put shape language into every branch. *Bonsai* facilitates duplicating and manipulating thousands of branch surfaces to help artists create trees efficiently.

2 Approach

Typically, it is rare to grow a totally procedural tree—some components (trunks, primary branches) are so art directed that they must be built by hand. As such, even a foliage asset created with the aid of a procedural system is in a constant cycle of iteration between multiple departments—some branches are hand-modeled, while other branches and leaf canopies are generated with code, and the final look is constantly in flux. Therefore, a main goal was to achieve a complete representation early on in Maya and give full ownership of the asset to the Modeling department.

The impetus behind the system (Figure 1) is that simple source branches serve as building blocks to create more complex hierarchical networks. Source input can be as simple as a cylindrical model or as complex as a system of connected branches. The user places a new branch by selecting a source (or *Bonsai* can randomly choose from multiple sources) and clicking on another surface. At that position, the new branch is parented to the clicked surface (inheriting its transforms) and assigned an attribute defining which level it exists in the hierarchy.

![Figure 1: A simple branch (A) with growth points (B) and user-controlled instancing (C), results in a complex structure (D)](image)

Further, our system allows better visualization of the final rendered output as a proxy leaf representation is provided for leaf-bearing twigs. Hence, the user sculpts leaf canopies in unison with branches in Maya, thus allowing Art Directors to critique and approve assets in one setting instead of many. We designed *Bonsai* with Modelers in mind, and a great deal of input from artists was utilized in the development of the system.

3 User-Controlled Instancing

After creation, the user has many options with this new branch. It can be translated, reoriented, or scaled. It can also be “pruned back” similar to a real life branch. Pruning entails removing geometry and children branches under the selected node, replacing it with a handle. This handle is essential for fractal growth in our system, as future branches can "grow" from this point.

Using a source branch with such handles, the user selectively grows new levels of branches. Each grow operation instances the source on to the set of selected handles (Figure 1C). Since the newly created branches have their own handles, this growth can be repeated ad infinitum. However, unlike typical procedural methods, the user interactively selects where this growth occurs. The user specifies which handles are selected, how many levels of growth occur and which sources to propagate. The system provides a powerful interface to select, navigate and manipulate the hierarchical structure. Artists can quickly prune parts of the tree that are less successful and grow new sections. Also, any part of the tree can be used as a building block elsewhere on the tree.

![Figure 2: One of the tree assets in Madagascar 2.](image)

4 Storage and Rendering

The exported data from Maya is processed via scripts in PDI/Dreamworks' proprietary scripting language and stored as a hierarchical collection of splines with parametric parenting and radii information. This data is important for animating resistance to wind and object-plant interactions. A procedural shader primitive in our proprietary renderer generates all geometry. The system turns the hierarchical splines into surfaces and fillets and connects separate branches. Branches are organized into shared material groups, based on their respective hierarchical level. These groups can be separated and individually painted if necessary, as texture coordinates are also generated.

---

1 email: Jeff.Budsberg@dreamworks.com
2 email: Scott.Peterson@PDI.dreamworks.com

Copyright is held by the author(s).
ISBN 978-1-60558-466-9/08/0008