Interactive Environment Creation with Sprout

Daniela Hasenbring
Sony Pictures Imageworks
dhasenbring@imageworks.com

Jeremy Hoey
Sony Pictures Imageworks
jhoey@imageworks.com

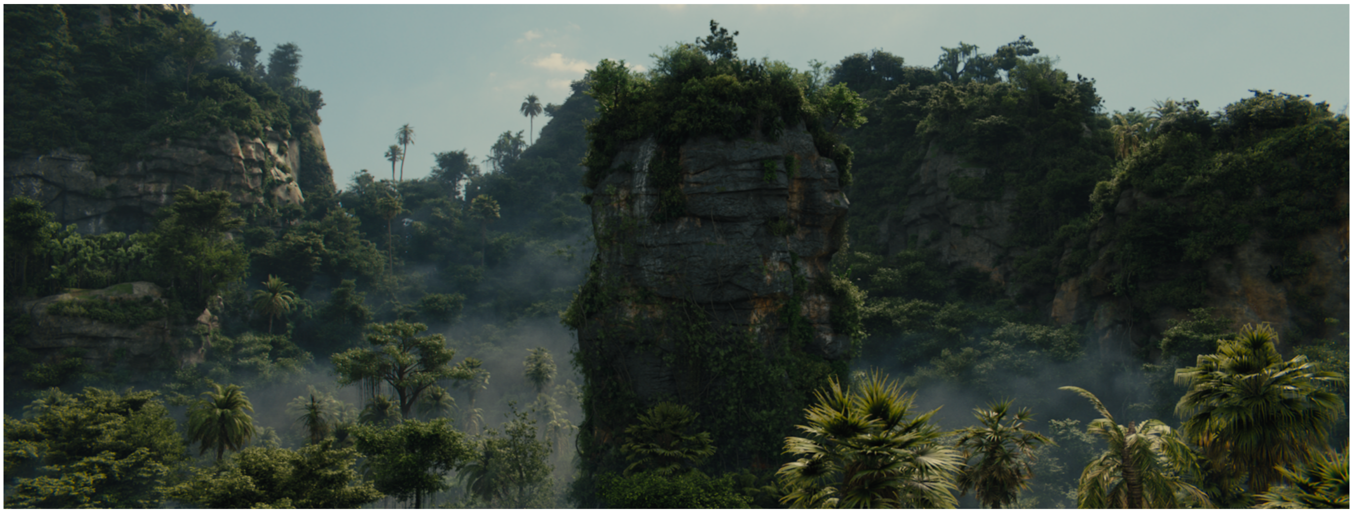


Figure 1: Jungle Environment from an upcoming VFX show. ©2017 Sony Pictures Imageworks. All rights reserved.

ABSTRACT

Sprout is our proprietary Maya-based tool for hand-dressing digital environments with large quantities of high-resolution assets like trees, plants and rocks.

It was developed at Sony Picture Imageworks (SPI) to address the need for an interactive artist-friendly tool that was fully integrated into SPIfis existing pipeline. Prior to the development of Sprout, environment dressing at SPI was done primarily in Houdini or procedurally at render-time and was thus the province of FX TDs.

In Sprout, artists can load any asset and quickly fipaintfi instances onto any other geometry using a brush paradigm familiar to anyone who has used Photoshop. Sophisticated lightweight OpenGL representations keep performance nimble, and all instances remain fully editable by the artist to allow for highly art-directed environment dressing.

Sprout has made environment dressing at SPI available to a larger variety of artists, being leveraged most recently for photoreal jungle environments for an upcoming VFX motion picture as shown in Figure 1.

CCS CONCEPTS

•**Computing methodologies** →*Non-photorealistic rendering;*

KEYWORDS

layout, environment, dressing, interactive

ACM Reference format:
Daniela Hasenbring and Jeremy Hoey. 2017. Interactive Environment Creation with Sprout. In *Proceedings of SIGGRAPH '17 Talks, Los Angeles, CA, USA, July 30 - August 03, 2017,* 2 pages.
DOI: http://dx.doi.org/10.1145/3084363.3085046

1 GENERAL WORKFLOW OVERVIEW

Sprout was created to address some specific needs missing from SPI's existing tools and workflows:

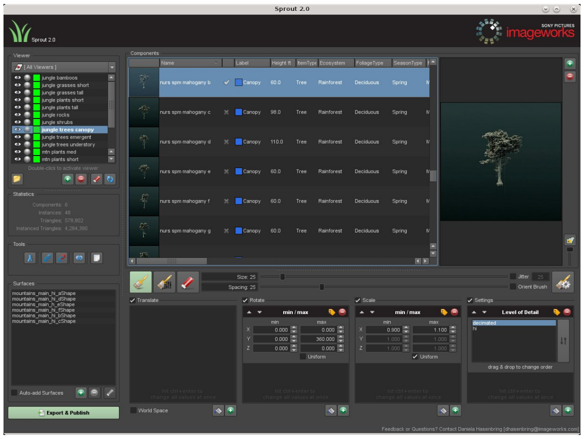
- A tool that was easier for non-technical artists to use than the Houdini-based scattering workflow,
- A tool to be able to hand paint instances rather than laboriously place instances by hand or rely on procedural scattering tools.
- The ability to quickly and easily edit every aspect of any individual instance for highly art-directed environments.

Sprout is above all approachable and can be learned by any artist in just a few minutes.

In an easy to use interface the artist can load any asset as a component and paint instances onto other geometry. The brushes used to paint can be fully customized in shape and come with a modifier stack to easily control or randomize per-instance properties like rotation, scale, level of detail, frame offsets and others.

Once placed, instances can be either further modified by using the same brushes in an edit-mode or by simply using the standard Maya transformation tools.

SIGGRAPH '17 Talks, Los Angeles, CA, USA
© 2017 Copyright held by the owner/author(s). 978-1-4503-5008-2/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3084363.3085046



Components: 6
Instances: 48
Triangles: 578,802
Instanced Triangles: 4,284,390

Name		Label	Height ft	ItemType	Ecosystem	FoliageType	SeasonType	F
nurs spm mahogany b	✓	Canopy	60.0	Tree	Rainforest	Deciduous	Spring	M
nurs spm mahogany c	✗	Canopy	98.0	Tree	Rainforest	Deciduous	Spring	M
nurs spm mahogany d	✗	Canopy	110.0	Tree	Rainforest	Deciduous	Spring	M
nurs spm mahogany e	✗	Canopy	60.0	Tree	Rainforest	Deciduous	Spring	M
nurs spm mahogany f	✗	Canopy	60.0	Tree	Rainforest	Deciduous	Spring	M
nurs spm mahogany g	✗	Canopy	60.0	Tree	Rainforest	Deciduous	Spring	M

mountains_main_hi_aShape
mountains_main_hi_dShape
mountains_main_h_eShape
mountains_main_hi_fShape
mountains_main_hi_bShape
mountains_main_hi_cShape

□ Auto-add Surfaces

Size: 25
Spacing: 25
□ Jitter 25
□ Orient Brush

☑ Translate
□ World Space

☑ Rotate

	min	max
X	0.000	0.000
Y	0.000	360.000
Z	0.000	0.000

□ Uniform

☑ Scale

	min	max
X	0.900	1.100
Y	1.000	1.000
Z	1.000	1.000

☑ Uniform

☑ Settings
Level of Detail
decimated
hi

Figure 2: The main interface of Sprout.

In order to keep performance at an optimum, components can be automatically decimated using SPI's algorithm for real-time voxelization on the GPU and are swapped out for lower or higher resolutions on the fly while moving the camera around.

Once the dressing is done, Sprout can then export the instances as either an Alembic point cloud for precise reconstruction with Katanafis point instancer, or as an explicit Alembic scene graph.

Sprout can also import any Alembic scene graph and convert to an instance array for high-performance, lightweight viewing and manipulation in Maya. This functionality has been successfully tested on environment assemblies of over 10 trillion polygons - viewable in the standard Maya viewport.

2 OTHER USES

Due to the modular nature of Sprout, it is not limited to only allow painting of instances. Other use cases such as distributing instances with dynamics, flood filling areas based on parameters or fully scripted instance placement can easily be implemented.

Sprout's API is slim, lightweight and gives full control over each instance via Python and MEL.

3 TECHNICAL DETAILS

Sprout's backend is written as a software independent C++ library that can be used by any software package that supports custom viewport plugins. It handles everything from loading geometry, storing it in a format that is optimizied for being used on the GPU, decimating high poly geometry on the fly, building texture atlases, to rendering that geometry as efficient as possible.

Due to it's very abstract design, loaders for new file formats or new render engines can be easily added - currently it supports Alembic files and has a rendering implementation for Legacy OpenGL, OpenGL 4.0 and the Maya Viewport 2.0 MPxSubSceneOverride interface.

For ideal performance, a texture atlas is created on the fly for meshes with multiple textures and UDIMs. For that, all textures are downsized and merged into one big texture. This ensures that any geometry can be drawn with just one draw call.

Similar to Maya's newly added level of detail feature, the implemented OpenGL render engines support swapping of level of details on the fly while navigating the camera, based on the distance of each instance to the geometry.

The paint interface shown in Figure 2 is independent of the viewer backend and is entirely written in Python and PyQt.

A new custom brush tool was developed as a Plug-in for Maya for an easy what-you-see-is-what-you-get feeling.

ACKNOWLEDGEMENTS

We are grateful to Daniel Canfora and Xavier Roig for their great ideas and pushing this project forward, as well as to Dustin Wicke who came up with the awesome name and idea to develop such a tool in the first place.