

Timothy Chrismer
Technical Artist

211 Liberty Avenue
Apartment 337
Lafayette, LA 70508

lossenc@gmail.com
(337) 257-0302

Technical Artist Reel- Shot Breakdown

Note: I am responsible for **ALL** aspects of these pieces, unless otherwise stated.

Unity Production Shader

Resources used: Unity, CgFx

During the concept development and pre-production phases of LITE's first project using the Unity engine, I built shaders based on what I felt Unity lacked. By default, Unity does not treat occlusion differently from lightmaps in fact, most examples call for both maps to be combined into a single lightmap. I started this shader to split these two maps in order to control them separately. Then, using my knowledge of compositing techniques, I added the ability to control occlusion color in Unity's editor. I added this same ability to the lightmap by treating it as a mask instead of a texture. Using red, green and blue lights, I could enable artists to control light color based on channels. I also saved the lightmap's alpha channel to serve as a shadow mask. Placing the power of color and intensity in the level artist's hands, I was able to reduce map bakes and cut iteration considerably.

Maya Attributes Manager

Resources used: Maya, Python, PyMel

The shot shows my Maya Attributes Manager in use. This script was created in the middle of a project to help streamline the production and to reduce artist iteration. The project used LITE's proprietary engine, LIME. Since the engine, itself was still new, various functions such as different face culling, alpha blending, and shader overrides called for specific calls from the exported data to the engine. These calls were eventually simplified to special engine-recognizable attributes that could be added in Maya. However, as the content team continued to use these attributes, it was clear that adding them was becoming a bottleneck, wasting production time. Artists would have to look up correct attribute spelling and types every time. To remedy this, I built the Maya Attributes Manager in Python and PyMel. This tool stores the correct attribute information to be added and is populated by swappable configuration files. In this way, attributes could be added or removed with a single button click and the attributes list could change according to the needs of the production.

Visualization Video

Resources used: Maya, Nuke

Responsible for: Lighting, Look Development

Rigging by: Daniel Tiesling

Animation by: Tom Bennett

This clip is from a 30-second commercial that LITE produced as a self-promotional piece highlighting its visualization capabilities. I headed up lighting and look development for the entire commercial. As artists would complete animations, I would address lighting and shaders based on the art direction I was given. The final piece can be found at <http://www.lite3d.com/videos>.

Unity Logger

Resources used: Unity, C#

I created this system to serve the needs of a project LITE worked on with researchers at the Department of Transportation and Development in Louisiana. The project used the Unity engine to help generate a training environment. The researchers, themselves needed to gather data based on each trainee's performance. We decided that each instance of the training environment needed to write to log files based on position, selection, timing, and sequence. No such logging functionality existed in Unity. Using C# I wrote a class that would create a single filestream instance that could be accessed by other scripts. I then wrote component scripts that would write to the log file, open, and close the filestream based on input and application state.

Timothy Chrismer
Technical Artist

211 Liberty Avenue
Apartment 337
Lafayette, LA 70508

lossenc@gmail.com
(337) 257-0302

Bamboo Plant

Resources used: Maya, Photoshop, BodyPaint3D, Shake

I created this piece as part of a photorealistic rendering class that I took during my time at SCAD. The requirements of the project were to create a photorealistic piece using only CG elements. The scene was modeled in Maya, while the textures were created in Photoshop using reference photographs and painting. I removed any texture seams using BodyPaint3D. Rendering in layers, I was able to tweak final lighting conditions in Shake.