3D Real-time Rendering Pipeline

Alejandro A Carriles
Arts and Entertainment Technologies
The University of Texas at Austin
Austin, USA
alex.carriles.p@gmail.com

***Abstract*—The purpose of this paper is to research specific processes that fill in the blanks to create a clear pipeline process for exporting 3D scenes from Maya to Unreal. This paper will delve into the systematic process of rendering for cinematic purposes. With the issue of time being the costly factor in rendering, we will look into creating a more efficient way to save time in this computer-based process. By utilizing the real-time rendering factors that the Unreal Engine 4 has to offer, there is a way to combine both software in a manner to make an easier workflow that allows for time to be saved and therefore allocated to more production development.**

Keywords—Autodesk, rendering, Unreal Engine 4, Maya, real-time, pipeline.

I. Introduction

A. Topic Description

Rendering has been considered to be the most time-consuming aspect of animation. When wanting to create a render preview or render the entire project, the processing power of the computer itself is what does the heavy lifting; leaving the user with nothing to do but wait… Unreal offers real-time rendering that saves an impressive amount of time as opposed to the standard method of cinematic rendering offered via Autodesk Maya or 3DS Max. While there are many methods to connect specific aspects from Maya to Unreal, there is no simple and clear-cut direction. Some of the many processes involve transferring models, animation data, and 3D camera movement.

B. Research Questions and Scope

With every process comes a challenge. The camera, for example, always imports into Unreal with a 90° rotation offset because Maya's Y and Z axis are inverted in Unreal [4]. For this specific issue, there is a way to run a Python batch code in Maya that corrects the problem when exporting the camera data.

This may sound straightforward, but the amount of research it took to find this solution was not as straightforward. For this reason, this paper will serve as a compilation of all of my findings. Hopefully, others will utilize this research as a guide for transferring data from Maya into Unreal to ultimately create a simpler and more efficient real-time rendering pipeline for 3D animation.

C. Overview and Sources

Real-time rendering has a multitude of outlets and possibilities, so for the purpose of organization and clarity, an outline of the specific topics that will be covered shall be clearly stated. This paper will primarily focus specifically on how each rendering system is predominantly used, both the pros and cons to each, and the trade-offs of using one over the other. We will then delve into how we can transfer 3D data from Maya into Unreal as well as adapt a Live Link that allows any changes made in Maya to automatically be updated in Unreal.

II. Intended Rendering

A. Maya

Autodesk has a proven track record for creating some of the best 3D software on the market. Maya in particular has some remarkable 3D tools and render systems that have led to it becoming the industry standard. The biggest issue, however, is the fact that you cannot preview how a scene will look in the final render without actually rendering it. So, what you end up with is a process by which you adjust your scene, wait for it to render, tweak it again, re-render, etc. [Fig 1]. Ultimately, by the time you're ready to do the final render of the scene on the highest quality, it takes an immensely long time. This effectively becomes the speed vs. quality tradeoff [3].

When you want to render something in Maya, you will encounter this speed/quality tradeoff. If you want to render something with very high fidelity, you're going to have to wait a while. Conversely, if you need something rendered out quickly, then you'll have to sacrifice the