Final Project

Sam Kellman-Wanzer

December 2019

1 Introduction

The Goal of my project was to create a two way file converter using PyMesh. PyMesh is a rapid prototyping platform focused on geometry processing. It provides a set of common mesh processing functionalities and interfaces with a number of state-of-the-art open source packages to combine their power seamlessly under a single developing environment. PyMesh allowed a seamless intregration of two-way GLTF file conversion.

2 converter.py : Creating the gLTF File

The first step of the project was using pymesh to parse the data of basic graphical file formats like .obj and .ply and store their information. An obj file is nothing more than a list of vertices and faces. With these arrays, we have all the information to create a gltf file. PyMesh supports .obj, .off, .ply, .stl, .mesh (MEDIT), .msh (Gmsh) and .node/.face/.ele (Tetgen) formats. Using that functionally, the user can simply enter in the name of that file, and if it is supported by PyMesh, it can be converted to a gLTF. After using the vertex and index array to create the file, converter.py saves the file in gLTF format, allowing that same file to be reconverted back to it's oaring, or a different format.

3 gltfconverter.py: Reconverting the small and portable gLTF file

Now that we have a gLTF file, we need to parse the data that contains the vertex, and face arrays used to create it in the first place. a gLTF file is simply a long and messy string. I used pygltflib, a python library used for converting, parsing, and representing gLTF files. pygltflib allows the file to be loaded in JSON format.

4 Data URI's: Decoding a string and reconverting it back into an array

The gLTF file was loaded into gltfconverter.py. In order to save it as a mesh, the vertex and face arrays needed to be parsed from the base64 encoded string. Python has a built in functionality, for that, but the data needs to be reconverted into the proper format. Vertex arrays are typically represented in floats, where face, or index arrays are represented in short integers. Now that the vertex and face arrays are back into their original form. PyMesh can recreate the file in any of the supported file types.