

# SIR ARCHIBALD GOES TO TOWN!

AN ANIMATED OPENUSD BASE SCENE.



BY: ILA SOLEIMANI

## Table of Content

**Introduction**

**Structure**

**Unpacking**

**Layers Structure**

**Geometry**

**Materials**

**Variants**

**Textures**

**Animation**

**Lighting**

**Coming next**

**Rendered scene**

*This work is licensed under Creative Commons Attribution-ShareAlike 4.0 International License.*

## Introduction

This project began as a personal learning sandbox, where I explored the vast possibilities of Universal Scene Description (openUSD) for creating animated CGI scenes. While there is an abundance of high-quality openUSD sample scenes available online, I noticed that many of them tend to be either overly complex and heavy (such as the *AnimalLogic Lab* and *nVidia Attic* examples) or excessively simple and specialized. Finding examples with a moderate amount of data, inclusive of the most common functions, can be challenging.

Motivated by this gap, I designed this project to serve as a middle-ground reference for those looking to explore openUSD in a more approachable yet versatile context. By offering a balanced composition of data and functions, this project aims to provide a valuable learning resource for individuals seeking to deepen their understanding of openUSD and its applications.

The project demonstrates one of the many ways a openUSD stage (i.e., asset or shot) can be composed, following a structure that is similar to the guideline from the **USD Working Group**<sup>1</sup> but seasoned by personal preferences. The project is designed for a single scene, with a focus on portability, ease of understanding, and ease of modification, making it an excellent resource for those new to openUSD or the ones who are looking for an openUSD toy to play with in their DCC of choice.

This guideline goes over the concept and external structure of the scene without getting into the technical details and the internal structure of the components.

---

<sup>1</sup> <https://github.com/usd-wg/assets/blob/main/docs/asset-structure-guidelines.md>

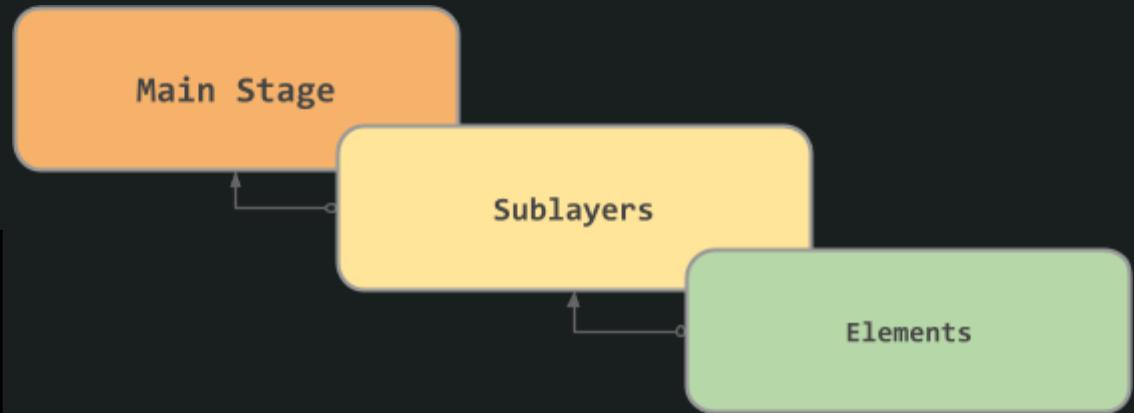
## Structure

The composition arc of this USD stage is illustrated in the following simplified schematic. It highlights the primary components and their relationships, providing a high-level overview of how the various elements come together to create a cohesive and animated CGI scene.

### Unpacking

Unpack the content of .zip file into a folder:

```
SirArchibald  
  /stage.usd  
  /assets  
  /anim  
  /lo  
  /lgt  
  /cfx  
  /fx
```



You will only need to open the layer ***stage.usd*** into your working space if you only want to render it. All file paths are relative and they should work as long as the files and subfolders are not renamed. ***stage.usd*** and the root folder can be renamed of course. The only non-usd contents are the texture maps, located inside the ***./assets/*** and ***./lgt/*** folders.

## Layers Structure

Here is the breakdown of the composition arc sublayers. Each sublayer is placed in a folder named after its respective function.

/assets

/SirArchibald

/geo/ <b>geo.usd</b>	<i>Static geometric shapes in default posture, plus geometric variant definitions.</i>
/mtl/ <b>mtlx.usd</b>	<i>Shader definitions in standard MaterialX and assignments, plus variant definitions.</i>
/tex/*. <b>exr</b> & *. <b>png</b>	<i>Textures, texture variants &amp; preview textures.</i>
/crv/	<i>Anything curve such as Grooms, fur and hair. <b>Currently empty.</b></i>
/assembly. <b>usd</b>	<i>Package containing geo and mtl as sublayers.</i>

/treadmill

/geo/ <b>geo.usd</b>	<i>Model, Static &amp; unshaded.</i>
/mtl/ <b>mtlx.usd</b>	<i>Contains an example of a simple materialX procedural Shading.</i>
/assembly. <b>usd</b>	<i>geo &amp; mtl sublayers.</i>

/anim/

*The geometric, deformation-based (point position) animation data. There are two sublayers here: one is for the hero model (SirArchibald), and the other is for the moving floor (treadmill)*

/lo

/cam/**camera01.usd** | *Camerawork, Animated.*

/lgt

/scenario01.usd | *Image Based Lighting, HDR image included (Courtesy of substance3d.adobe.com/assets)*

/cfx/

*Character animated effects such as groom and cloth simulations. Currently empty.*

/fx/

*General animated effects such as volumetrics and RBD. Currently empty.*

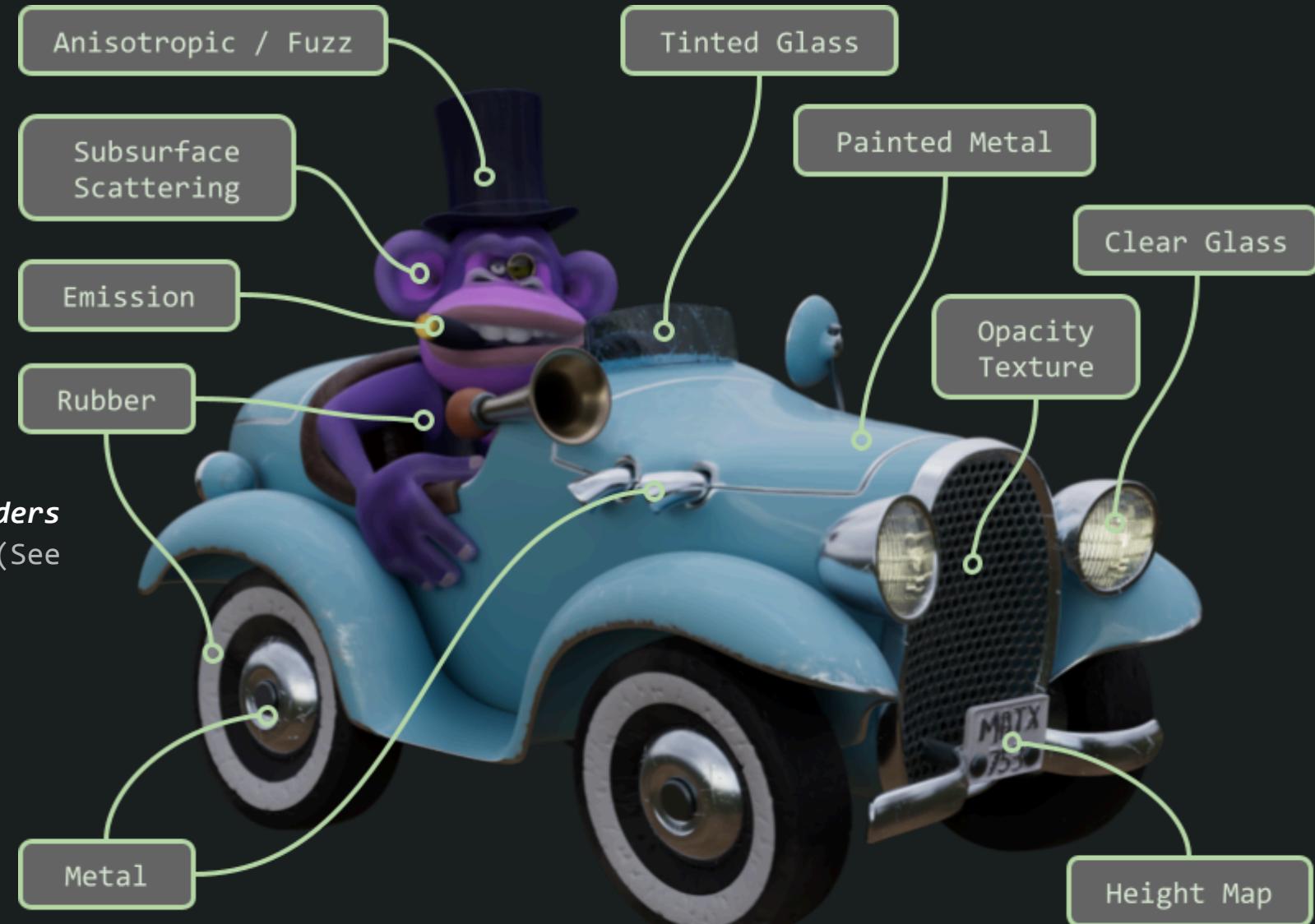
## Geometry

At the outset of this project, the focus was not on animation, and the geometric model was crafted in Autodesk Maya™ in a static pose to serve the primary purpose of showcasing a still CGI scene. However, as the project evolved, the decision was made to incorporate some animation elements. Although rigging for animation became more challenging and less efficient due to the initial static model, it was an acceptable trade-off given that the project's primary focus was not rigging or the animation process itself. The geometry information is encoded in binary USD instead of ASCII for portability and saving disk space.

## Materials

Main model has a single PBR material assigned. The different properties are controlled by various textures. It is defined in standard **MaterialX** format in Houdini Solaris and stored as a standalone usd file. The goal was to have as many different material properties as possible in one set.

Additionally, three **UsdPreviewShaders** are used for the preview purpose. (See variants)



## Variants

There are 2 variant sets available: *Wheel type* (geo) and *Paint job* (mtl). Material variants are defined using different textures and are controlled via a couple of *switch nodes* in the **MaterialX** network. Display shader and its variants are defined independently with only 3 textures (color, metalness & roughness).



Paint: Blue / Wheels: Chrome Wide



Paint: Green / Wheels: Painted Narrow



Paint: Yellow / Wheels: Painted Wide

## Textures

Adobe Substance 3d Painter™ was used to create all texture maps. There are two sets of textures:

A set of 2k textures (20 x 10 UDIM tiles) for **rendering**, stored in .exr format & encoded in **ACEScg** colorspace.

- color (+ color variants)
- metalness
- roughness
- translucencyAmount
- translucencyColor
- sssAmount
- sheenAmount
- sheenColor
- sheenRoughness
- anisotropyAmount
- anisotropyAngle
- inverseIOR
- height
- normal
- emissionAmount
- emissionColor
- coatAmount
- coatColor
- coatRoughness
- opacity

A set of 1k textures for **preview** in .png format.

- color + color variants
- metalness
- roughness

## Animation

To facilitate the animation required, a basic rig setup in Autodesk Maya™ was implemented. This minimal rig served its purpose well, allowing for the demonstration of fundamental animation principles within the context of this USD stage. The exported animation layers (located in `./anim/`) only contain deforming cache (point-position transform) on all the geometries.

The camera motion is baked into the `./lo/cam/camera01.usd`



→ The frame range is set from **1001** to **1288**.

## Lighting

For this scene, one simple lighting scenario is provided, which is basically a *UsdLuxDomeLight* with an hdr image attached. Nothing fancy, and you can easily replace it with your own lighting setup. Open the *stage.usd* file with a text editor, and replace the *scenario01.usd* with yours.

```
#usda 1.0
(
    endTimeCode = 1288
    framesPerSecond = 24
    metersPerUnit = 0.01
    startTimeCode = 1001
    timeCodesPerSecond = 24
    upAxis = "Y"
    subLayers =
        @./lo/cam/camera01.usd@,
        @./anm/treadmill01.usd@,
        @./anm/sirArchibald01.usd@,
        @./lgt/scenario01.usd@
    ]
)
```

**Note:** The *scenario01.usd* sublayer features an unusual inclusion: *animated focus distance* data for the camera. While this may seem unconventional, it was implemented during the lighting stage and inadvertently became part of the lighting export. To incorporate this animation into your own lighting scenario, you can copy the relevant section from the original file and paste it into your setup. In the next version of this project, the animated focus distance data will be properly transferred to the *camera01.usd* sublayer, ensuring a more streamlined and organized composition.

## Coming next

Hopefully, in the next version(s), I'll add groom, smoke & fire, more lighting scenarios, better asset structure, more color variations, openPBR materials, and more.

## Rendered scene

And finally, here is a link to a video rendered with Houdini Karma™, and post-processed in DaVinci Resolve™.

<https://vimeo.com/1024551071>

Have fun!

---

XII.II.MMXXV

*Disclaimer: This is a work of fiction and any resemblance to a real primate is purely coincidental. The author of this work is strongly against smoking.*

*This work is licensed under Creative Commons Attribution-ShareAlike 4.0 International License.*