LÖVR Demo Docs
for Interactive Music Experiences

assembled by Zachary Talis

Spring 2022

https://github.com/ZacharyTalis/lovr-demo/

Contents

1 Introduction

1.1 What does this doc cover?

We're going to make a monkey spin in LÖVR!

LÖVR is a simple-yet-powerful game engine for VR. By the end of this doc, you'll have tackled these LÖVR essentials:

- Creating, UV-mapping, and exporting a simple 3D model from Blender.
- Slapping some materials onto the model in Substance Painter.
- Assembling a final texture in Krita.
- Writing a simple Lua script that LÖVR uses to display our model.

Parts of this guide are based off LÖVR's "Callbacks and Modules" documentation[1].

1.2 Tools we'll use

You can download everything here for free!

Blender[2] is a 3D-modelling software that does a little bit of everything.

Substance Painter[3] is a nondestructive, mask-based, Adobe-owned texturing tool. It's free with an `edu` email address.

Krita[4] is an image manipulation and painting tool.

LÖVR[5] is a cross-platform VR engine that flouts intuitive Lua scripting and a light footprint.

[1] `https://lovr.org/docs/Callbacks_and_Modules/`
[2] `https://blender.org/`
[3] `https://adobe.com/products/substance3d-painter.html`
[4] `https://krita.org/`
[5] `https://lovr.org/`

2 Blender (Suzanneification)

2.1 A fresh Suzanne

1. Make a new Blender file, delete any default objects, and add a `Mesh > Monkey`. Say hello to our new friend Suzanne[6]!

2. Go into `Edit Mode`.

3. Unwrap some UVs for our Suzanne. You've got two options: either `Smart UV Project` or marking the UV seams yourself.

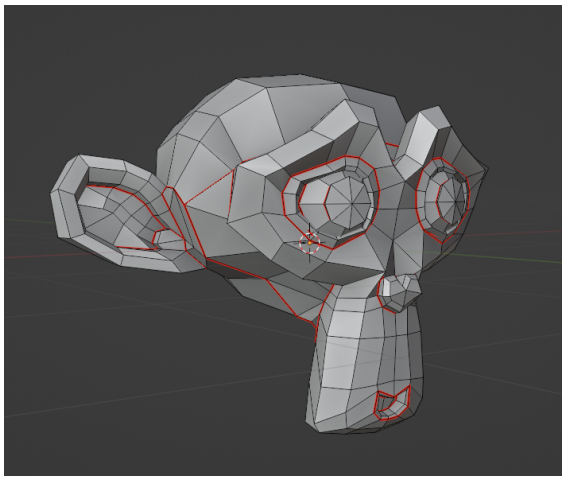

Figure 1: *UV seams marked on Suzanne.*

2.2 Exporting to glTF

1. `File > Export > glTF 2.0`.

2. Change the `Format` to `glTF Embedded (.gltf)`, and export.

[6] `https://docs.blender.org/manual/en/latest/modeling/meshes/primitives.htm#monkey`

3 Substance Painter

3.1 New file

1. `File > New`, as all great things begin.
2. `Template`: set to `PBR - Metallic Roughness Alpha-blend`.
3. `File`: select your exported glTF.
4. `Project Settings`: `Document Resolution` of 2048.
5. Make sure `Auto-unwrap` is disabled, then press `OK`.

3.2 Rendering maps

1. `Edit > Bake Mesh Maps`.
2. Set `Output Size` to 2048, then `Bake selected textures`.

3.3 Smart materials

1. Search the assets browser for smart materials of your choosing. Drop 'em onto Suzanne.
2. For each smart material layer, create a black mask.
3. For each mask, use `Polygon Fill > UV chunk fill` to give Suzanne some pizzazz.

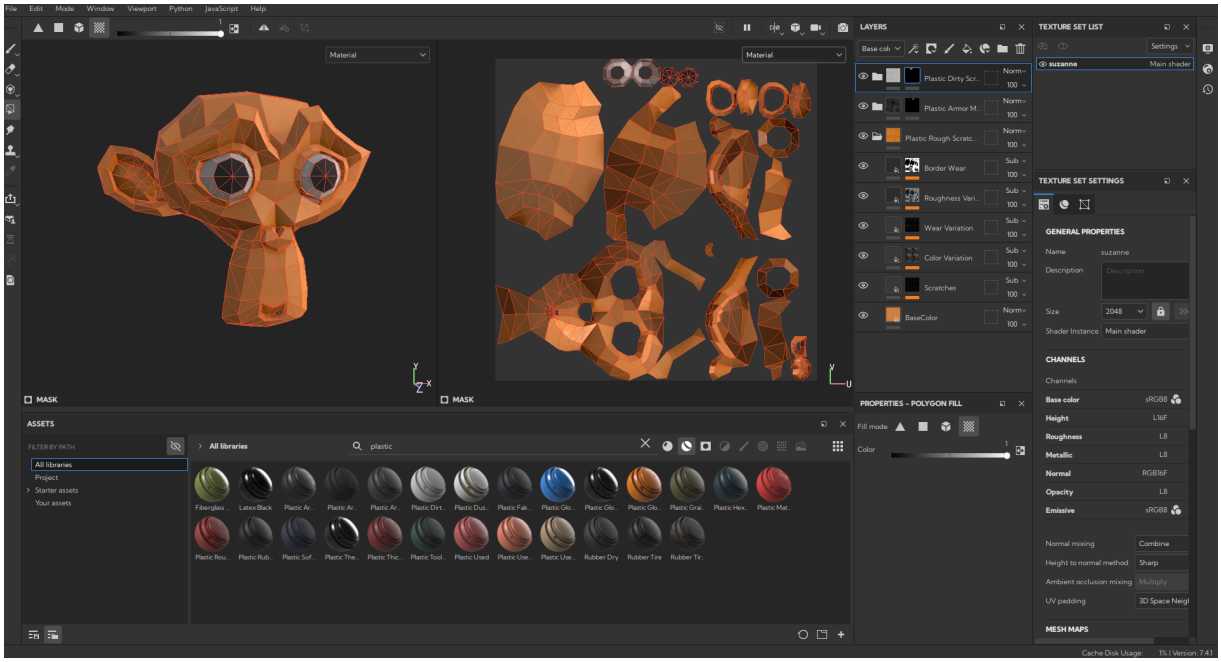

Figure 2: *Substance Painter project with decked-out Suzanne.*

3.4 Exporting

1. `File > Export Textures...`

2. Export using both the `PBR Metallic Roughness` and `Mesh Maps` templates.

4 Krita

4.1 An ambient excursion

1. Open the `ambient_occlusion` and `BaseColor` images in Krita. Plop them onto two separate layers (ambient occlusion on top).

2. Set the ambient occlusion layer to a low opacity and the `Addition` blending mode.

3. `File > Export`, then save as a PNG.

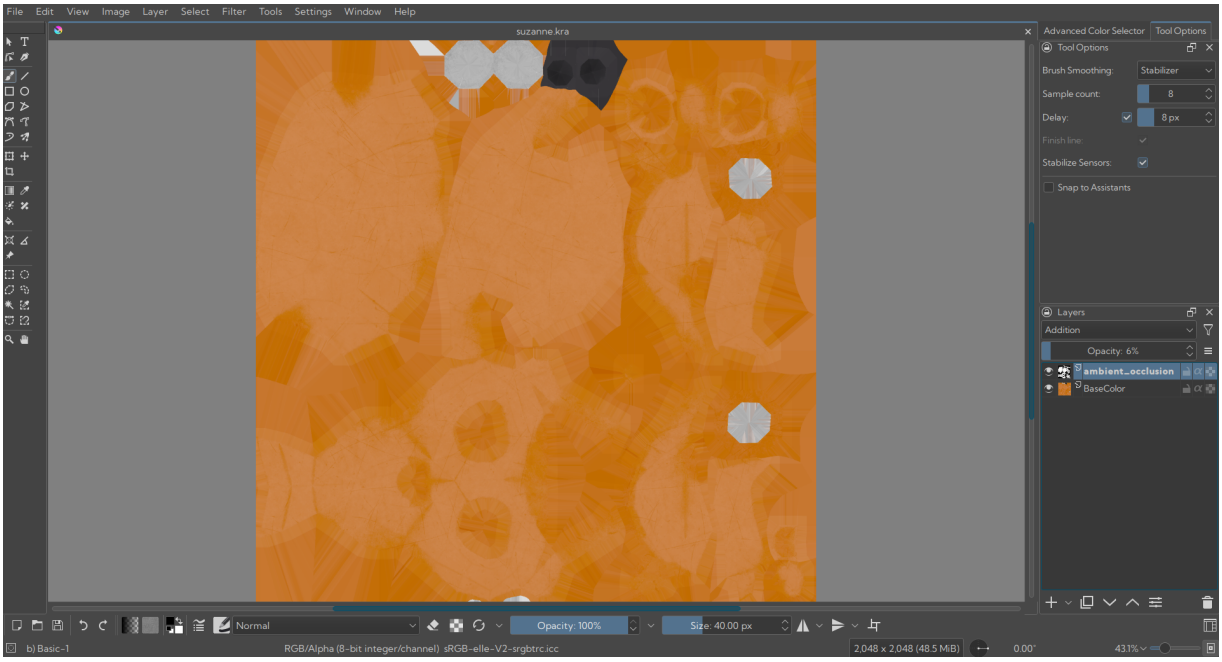

Figure 3: *Krita project with those two texture layers.*

5 Blender (Final Export)

5.1 Giving Suzanne our texture

TK

Then export again.

1. Give Suzanne a new material.

2. Configure Suzanne's material with the texture PNG as shown below in **Figure 4**.

3. Export to glTF once again.

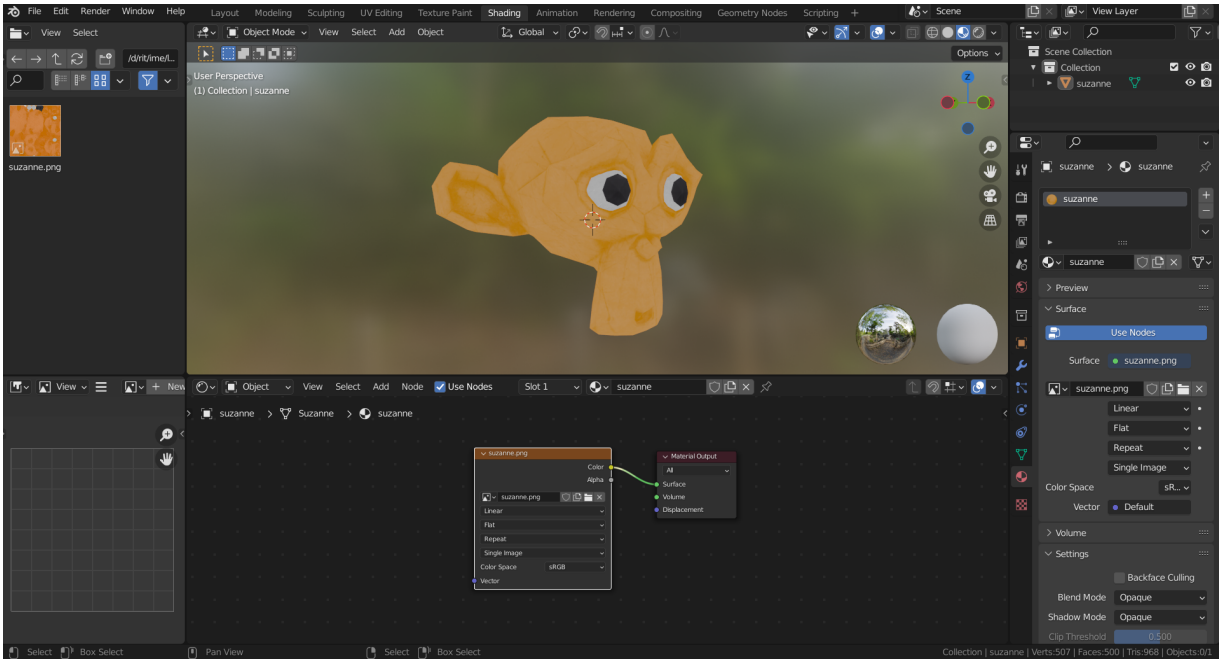

Figure 4: *Blender project showcasing Suzanne's nifty material nodes.*

6 LÖVR

6.1 Project structure

1. Create a folder on your computer that contains:

   (a) A subfolder `assets` with subsubfolder `gltf`. Place your exported model inside here.

   (b) Empty text files `conf.lua` and `main.lua`.

   (c) LÖVR's executable/dependencies from the LÖVR download page[7].

2. Open the directory in a text editor. Visual Studio Code[8] works great for this purpose.

[7] `https://lovr.org/downloads/`
[8] `https://code.visualstudio.com/`

6.2 conf.lua, for convenience's sake

To force LÖVR to run in desktop mode (rather than in VR mode), add the following code to `conf.lua`:

```
function lovr.conf(t)
    t.modules.headset = false
end
```

6.3 Resource imports

Drop this code into `main.lua` to import Suzanne's model:

```
function lovr.load()
    Suzanne = lovr.graphics.newModel("assets/gltf/suzanne.gltf")
end
```

6.4 Spinny Suzanne

Add some more code to `main.lua` to make Suzanne appear (and spin)!

```
function lovr.draw()
    Suzanne:draw(0, 0, -3, 1, lovr.timer.getTime())
end
```

6.5 Running the project

From command line, run LÖVR's executable with the current directory as its sole argument. On Linux, this looks like:

```
./lovr-x86_64.AppImage .
```

You should see spinny Suzanne!

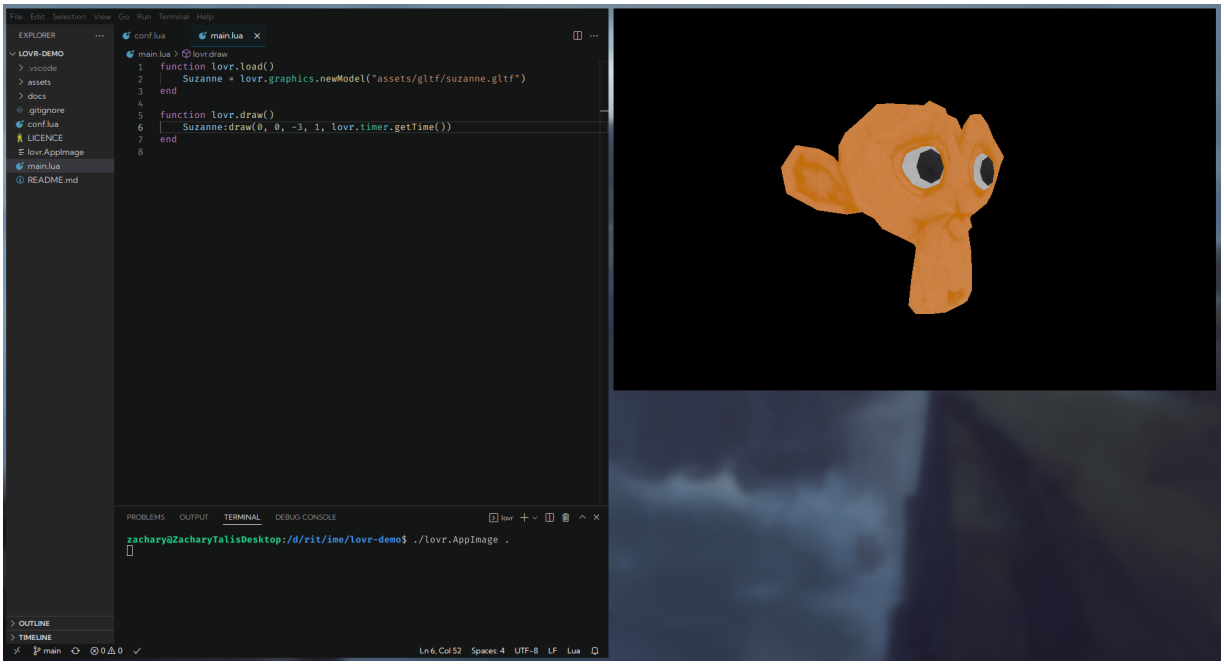

Figure 5: *LÖVR project running and open in Visual Studio Code.*