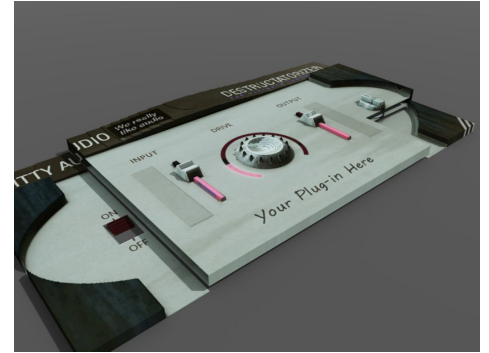




Visualizing a 2D PBR interface with voxels



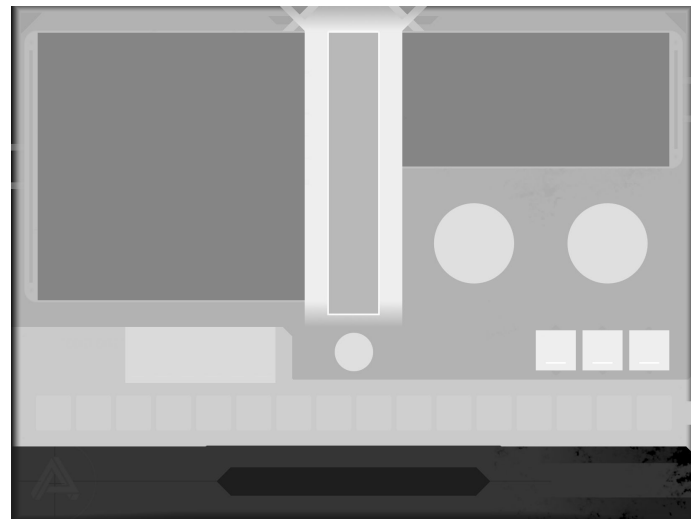
Why not?



Meeting
Apr 18th 2023

Context

- Dplug audio plug-ins can optionally be **PBR**, while staying 2D for authoring.
- **Depth** given by a 2D elevation map
- So, no **3D** render for marketing material.

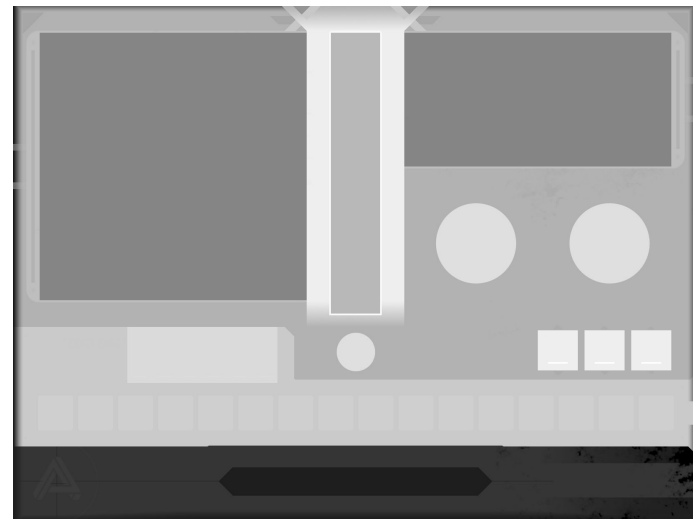


A 2D depth map

Context

- Dplug audio plug-ins can optionally be **PBR**, while staying 2D for authoring.
- **Depth** given by a 2D elevation map
- So, **no 3D render** for marketing material.

But the plugin industry increasingly use 3D models to render 2D UIs!



A 2D depth map

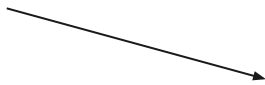


Let's extract voxels from the render!

Result = a [Qubicle Binary Voxel File \(QB\)](#)

Example

.qb voxel model
in a viewer



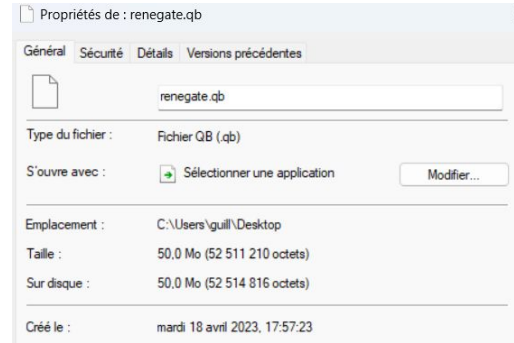
ugh, colors are wrong it seems

What to do with that .qb file?

Open with:

- Drububu <https://drububu.com/miscellaneous/voxelizer>
- Qubicle <https://www.minddesk.com/>
- Vengi voxel tools <https://mgerhardy.github.io/vengi/>

Can export to many other 3D formats with those tools.



HOWTO

In your `gui.d`
constructor:

```
// onScreenshot will be called at next render  
// (this can be called from anywhere)  
context.requestUIScreenshot();
```

In your `gui.d`
oplevel:

```
// Show how to do a .qb export of final PBR render  
override void onScreenshot(ImageRef!RGBA finalRender,  
                             WindowPixelFormat pixelFormat,  
                             ImageRef!RGBA diffuseMap,  
                             ImageRef!L16 depthMap,  
                             ImageRef!RGBA materialMap)  
{  
    // alternatively: encodeScreenshotAsPNG  
    ubyte[] qb = encodeScreenshotAsQB(finalRender, pixelFormat, depthMap);  
    if (qb)  
    {  
        writeFile(`/my/path/to/plugin.qb`, qb);  
        free(qb.ptr);  
    }  
}
```



Demo with Renegade

Note: Dplug voxel export is post-PBR, so additional shadows will apply from voxel renderer => incorrect.



In the future:

1. *Fixing the off colors.*

2. *Maybe could extract a screenshot periodically to render a short .yuv video (no promises).*