VR software UI/UX dev logs

# Some preliminary ideas:

Make use of 3D vr game elements as UI instead of usual file browser archetype. Example ideas:  
use bookshelves to show timelines, use book flip to browse images, throw a disc tape to wall to play video etc.

10th March 2024

Followed this amazing tutorial: [Godot 4.0 File Explorer Tutorial (File Selecting) - YouTube](https://www.youtube.com/watch?v=mC4Wb_NHKA8) for basic file browser setup, it works so good, now just need to add support to open images and videos (stitched). For some reason, the images .jpg saved before stitching is most likely corrupted as it cant be opened with error 16, this causes me so much delay debugging lmao hate it. But thankfully the stitched output opens with no problem. Other than that, I also need to change the ffmpeg commands to output .ogv video files instead of .mkv for compatibility/support with videostreamplayer in Godot. More in processing logs.

Ok, great! The video player also works (bare minimum, no resizing/UI optimisation etc), but it works! Just adapted the controls and base etc from the SBS video player, the hide/show nodes part kinda sketchy but works for now!

Next, I should make some basic timeline features in 2d first, then maybe have some book UI for 3d VR. Just gotta keep cooking and be creative!

To do this, its probably easier to use directories/folders as timeline tag/indicator, so I need to reorganise folder, refer to processing logs.  
  
OK just realised, before anything else, I should figure out a way to get all files show their thumbnails, or else this will be worse than just using windows explorer.

Image thumbnail works! Now towork on video thumbnail :skull:

Other than that, I should prioritize getting metadata reading working first before other stuff next!

Video thumbnail stuck now, don’t know whats wrong or how to fix, gonna move on to metadata.

There’s apparently no straightforward way to get metadata info of image on godot, so I’ll rewrite processing script of places detection to produce .json file for each file instead and read that.

Ok fixed the video thumbnail while contemplating on how to copy the .json from img0/vid0 to output lmao, basically the gist is the problem is that I didnt instantiate any textureRect for the videoplayer to get texture from so adding hidden\_screen texturerect fixed it, just need to hide it straight away etc.

Next, metadata filters.

11th March 2024

Ok I added thumbnail for both pictures and videos, and fixed layout/aspect ratio so it looks better.

Reading .json is easy, the problem is how to read all and filter etc. Ok so to start from basic, im going to implement basic search/query first.

Ok very basic search/query from filename done! Used enter/submit instead of change detect due to performance issue from thumbnail reloading.

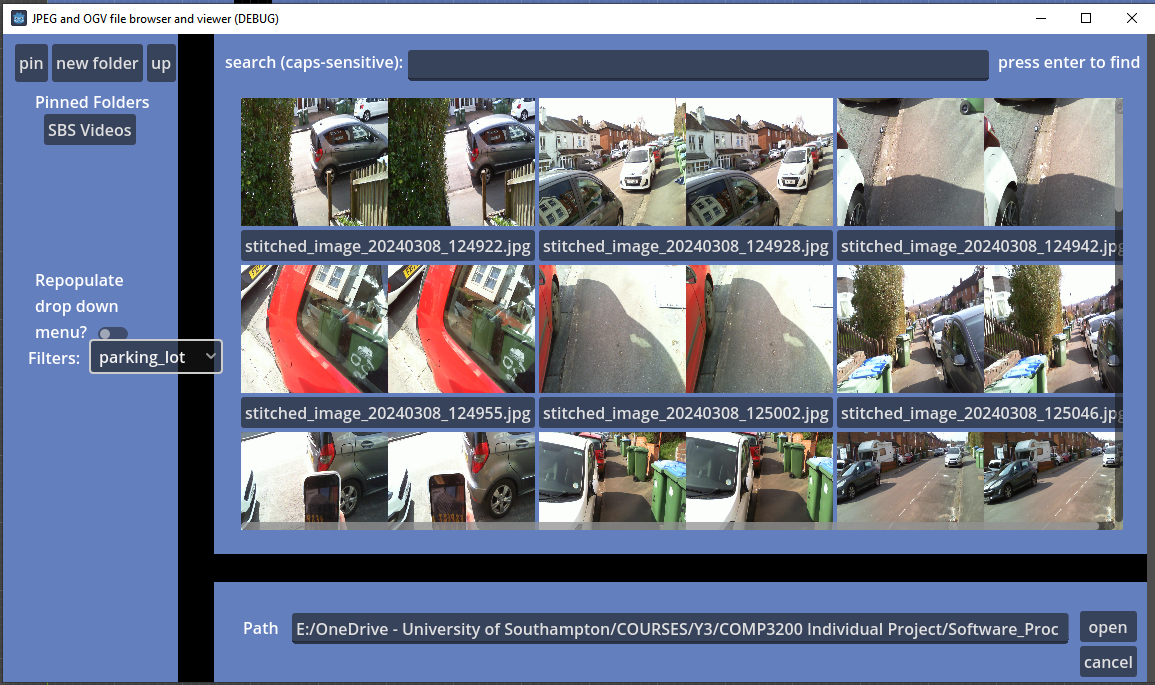
Now time to do some basic filter using metadata, so time to have a dictionary to metadata?

Ok I think I need to do some more processing on the extracted .json file for each week output as planned, then I can refer this to show filter drop down with number of item for each etc.

Ok added basic drop down with automated element from json analysis. Known bug is pressing up button when videos thumbnail haven’t finished loading crash it (something about freed object so I commented await line to make it faster)

Now to implement metadata filters algorithm.

Ok holy shit

This took like 2/3 hour or so, damn but finally image filter works, the filter check button is kinda counter intuitive and hacky solution for now, but it works, worst part is it needs to run twice to load the correct dropdown for some reason:  


15th March 2024

Added support for videos metadata filtering as well though a bit spaghetti code rn ngl including metadata and json .py scripts. Next, add a convenient button/UI choice to filter through weeks folders

Maybe a dropdown choice then a button to go to the path!

Also, add next/previous button and change the browse button to go back to current path on lineedit.

18th March 2024

Loading… added when opening folders and stays when loading .jpeg/.ogv files, delay needed for it to work, so 0.01 s delay is added.

Now adding next/previous button. Worked perfectly using Globals array variable to store the filename, only limitation is that the browse folder need to be done first or else it wont be populated! THIS ALSO WORKS PERFECTLY WITH FILTERS WOOOO because it uses current layout of folder populated including the filters NICE!

Now adding dropdown choice for weeks folders etc. Ok bare basic done, ideally should be dynamic and have the disabled according to folder directories but claude reached limits rn so I’ll just continue with trying to integrate this with VR first instead

Also, added the browse dir retain functionality.

# VR INTEGRATION

Ok finally after going again from fresh and on Godot 4.2.1 instead, the project named VR Lifelog Data Viewer is going pretty well, just follow official docs XR basic tutorial etc [XR — Godot Engine (stable) documentation in English](https://docs.godotengine.org/en/stable/tutorials/xr/index.html). Best thing is that there is assetlib tools to simulate XR controls so I can test faster and easier though it only works from the editor for debug purpose rn. For now importing scenes and basic VR setup/movement is done, now time to add the SBS shader into the Screen and image display.

OK added both video and image shader and screen for both, very spaghetti and hackish way but works for now. Also wth recheck your shaders assignment etc properly, this took me like hours to debug bruh.

Now test with VR. Ok for some reason my godot run in editor don’t work for VR, getting OpenXR: No viewport was marked with use\_xr, there is no rendered output! Error. But after exporting as .exe and running it works as expected, there was some bugs that is squashed, kinda weird how it suddenly appear and didn’t realised it before, mainly the file=false Boolean change when pressing file button multiple times, and still unsure bug about SBS\_Screen duplicates showing due to autoplay emit, need further research, other than that, I need to change the thumbnail gen to be better cus rn its so bad. (Add more delay)  
OK thumbnail gen might not be fixable easily in software, changing delay from 0.001 to 0.01 and 0.1 have negligible improvement, and changing to 0.2 is too slow (hangs up), best is to bet on better hardware later to get better first frame, or just use -ss 5 or higher to skip more frames in ffmpeg commands.

# Deploy to android as plan B?

Due to the fact this is quite storage intensive and very badly optimised software, using my own PC is ideal especially because the files/data are in my HDD, but as Plan B/fallback, I’ll try to export it as .apk for my quest pro and run it locally on headset to see if its viable with less dataset and just proof of concept, ideally, the files would be hosted on the cloud but that’s beyond the scope and time constraints I have rn.

For some reason, im getting some weird vendors and openxr problems/bugs on my projects, but it works fine on demo one, so rn I think best idea is to just create new project and slowly test and integrate back working v3 parts one by one.

FIXED the weird no output rendered bug, apparently it’s the XR input simulator addon is the culprit, removing it (not disabling it) from the autoload fixes it, just disabling it doesn’t fix it.

KEEP THIS IN MIND EVERYTIME USING IT! [OpenXR: No viewport was marked with use\_xr, there is no rendered output! · Issue #8 · Cafezinhu/godot-vr-simulator (github.com)](https://github.com/Cafezinhu/godot-vr-simulator/issues/8)

