Blog 4

This Week I put my efforts into working on the exporting part of the project as to make sur I have something working for the mid-demo this section turned out to be trickier than I first made it out to be

I started by building out the UI first no as I wasn’t familiar with tkinter at this point so I spent a while trying to figure out how to layout different elements. The difference between place, .pack and .grid

For this screen I used a grid to layout all the frames and with for the elements I packed them inside. After experimenting with the different I found this to be the easiest way to place the elements where I wanted them.

I attempted to add a delete function on the list but had some trouble getting it to work as. Ill leave this along with the filter options for a later date.

After my failure at that I decided that I needed to focus on what was more important, which is the export button. Once I had that done I would feel more relived and able to start building out the pipeline of my app

Turns out Scratch can be a pain sometimes. I made a prototype to learn earlier on in the project to use as a base and when I finished that up I tried exporting 3 pngs

Text, table

Description automatically generated

Text

Description automatically generated

This was the output as a ZIP (because sprite3 compresses using zip). It turned out that this didn’t work. As Scratch didn’t want recognise this as a valid sprite. I spent a while trying to debug this and do more research.

I then tried recreating my desired sprite using scratch to try reverse engineer the solution this is what I got Text

Description automatically generated

Text

Description automatically generated

It looked practically the same as the other one apart from the assetId, I went looking further in the sprite documents <https://en.scratch-wiki.info/wiki/Scratch_File_Format#Assets> I looked at the assetId section and it said it was the MD5 hash as the asset file. Because of this I tried to find out the md5 hash of the images using a ubuntu terminal. It came out as a different value which worried me a bit. After some research that lead to not a whole lot I decided to try using the values I was getting from the images to see if it did work. Turns out I does and the sprites were working as expected. My guess after turning the images from different formats and back is that Scratch internally turns the images into a different format or compresses them internally leading to a different hash value.

Additionally, while researching assetId I found out what rotationCenterX/Y was which just says where the centre of the costume is. For this I just halved the width and length of the image to save some hassle later.

Next week I have a few goals I want to achieve.

* Refactor my current code to be a lot cleaner
* Start implementing the camera elements to work in the main project.
* Get the program to a state where I could do a mid-project demo.