Summative Reflection Report

Judging from the conclusion, my work must have failed. My original intention was to use the Raspberry Pi as a carrier to run processing to achieve better results and support touch screens. In this way, the Raspberry Pi can be turned into an inherently interactive smart mirror. But when I started my progress, I found that the difficulties were beyond imagination, some took a lot of time and some couldn't even be solved.
My first question is the adaptation of the Raspberry Pi to processing. After I entered the Raspberry Pi system, I found that the processing program in the Raspberry Pi did not provide any adaptation software packages. I had to download it myself. And install the runtime library, but the mirror source of some libraries has expired. In the end, I can only download a 2013 integration package and rewrite it into the Raspberry Pi, and then slowly update each software library.

Then for the three important components of Raspberry Pi, touch screen and camera, which are required for my work, they did not provide any drivers. When I connected the Raspberry Pi with VNC viewer, I found that its display resolution is actually The rate is actually wrong, and it is easy to solve. But when I installed the camera, I found the difficulty: not only the Raspberry Pi itself lacks the driver for the camera, but also the processing software cannot correctly recognize the camera. The camera driver must be packaged into a recognizable Gtreamer program and loaded into processing, which is difficult to do. It almost took me most of my time to figure it out. Otherwise, even if I can call up the camera on the terminal, my picture is always black in processing, because the camera is being occupied by the system at this time, otherwise the system cannot correctly feedback the camera information . After a lot of queries, I finally found a way to call up the camera screen displayed correctly in the system, but I found that this method will always reduce the quality of the camera.

At this point I have spent a lot of time, but the trouble is far from over. For the processing software in the Raspberry Pi, because the operating structure of the system is different from the computer, the audio and video library cannot be used. You must download the GL Library to replace the video library, otherwise the software will continue to search for the camera. Because of this, I cannot export the sound because I can't find a library for the same sound. And because the GL Library is adapted for processing 1.5 and has not been updated, I cannot use any pixel function, otherwise it will conflict with the main program, and I originally planned to use the Blob direction library to track particles, and for this reason I had to give up . It should be noted that because the referenced libraries are different, my code cannot be run on the computer. I am very sorry.

In short, my final results are far from what I expected. I used the touch of the screen to

switch between particle effects and mirror effects. Due to the performance of the Raspberry Pi, when switching to particle effects, the frame rate will be lower. I added two simple borders. Each time you tap the screen, the effect will be switched immediately, and drag will make the particle effect follow the movement. In order to reflect the mirror, I placed the screen vertically, and I did not fix every component for the convenience of debugging.

I have uploaded all the codes I used to https://github.com/panyicheng40/ADAD3402, the above is all my project records, thank you for your reference.