

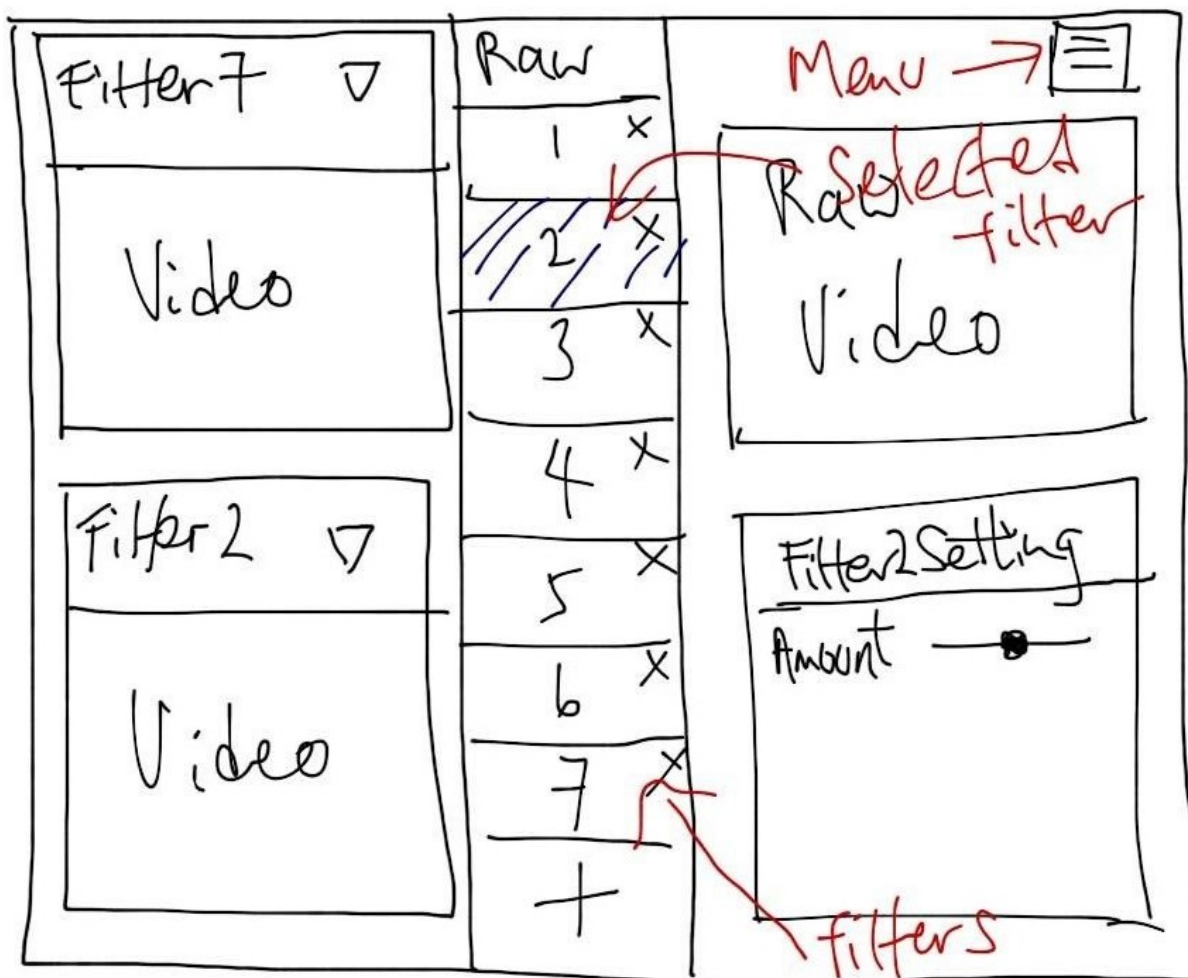
Group

Team Glyph

Madhu(A0074807Y), Jonathan(A0073924B), Wei Kuan(A0072749U)

Change of Plan

After several round of discussions we have decided to change parts of the layout design. Below is a mockup of our new design. The left portion consist of the filters camera widget. The middle section consist of a list of filters added. The right side consist of the menu button, the raw video feed as well as the settings section. When the filter in the middle is clicked, the settings section will display the settings for that filter. The menu button shows a popout context menu of additional options such as video record, screenshot, saving filters.



Current Status

- Connecting to camera
 - We have created a camera stream object that connects automatically to the camera to obtain video feeds.
- Camera Widget
 - A widget to hold the image frame and updates constantly with the camera stream
- Filters
 - Filter objects that allows image filters to be added onto a camera widget
- Layout skeleton
 - Created a skeleton of the layout using Qt creator and added a camera widget to it. However, we have plan write this part programmatically instead.
- Development of Filter Widget
 - The filter widget development have started but not complete at the moment.

Difficulties encountered and approach to resolve the difficulties

- Video stream
 - In order to allow for multiple filters, we realised that the video stream have to be shared to different camera widgets. However, there can only be 1 object accessing the camera at one time.
 - Therefore, we have created a video stream object and allow multiple camera widgets to access it at the same time.
- Smooth updating of video feed
 - Although video is captured to the video stream, the camera widget have to update it's image constantly. Using a periodic update might skip frames.
 - Instead, we used signals and slots so that camera widgets update their image when the camera stream's image have changed. This allows the camera widgets to update immediately to reduce delay.
- Qt Creator
 - Although Qt creator allows us to create window or widget layouts through a drag and drop manner, we found out that it is quite difficult to use in terms of autolayout. At least, we have spent quite some time trying to fix the layout but could not perform it properly.
 - Alternatively, we have decided to code the UI layout programmatically. Although it might take some time trying to figure out the syntax and methods available, we should be able to understand the layout constraints that we have set ourselves.
- Filter Settings

- Since there are many types of settings, most of them will probably have similar types, such as a seekbar or a spinner. Coding all of them at UI widgets might be painful and redundant.
- The solution is to have a description of the filters in the filter class. It include certain attributes such as setting type (seekbar). For a seekbar, it will then have a maximum and minimum amount as well. These settings will be stored in a form of dictionary. Thus, the code to generate the filter settings UI can be reused based on the needs of the different filters.