

# VISION SYSTEM CONTROL PANEL

---

CS3249 PROJECT PRESENTATION

TEAM GLYPH

# Agenda

---

## Team Glyph

Introduction To The AUV System

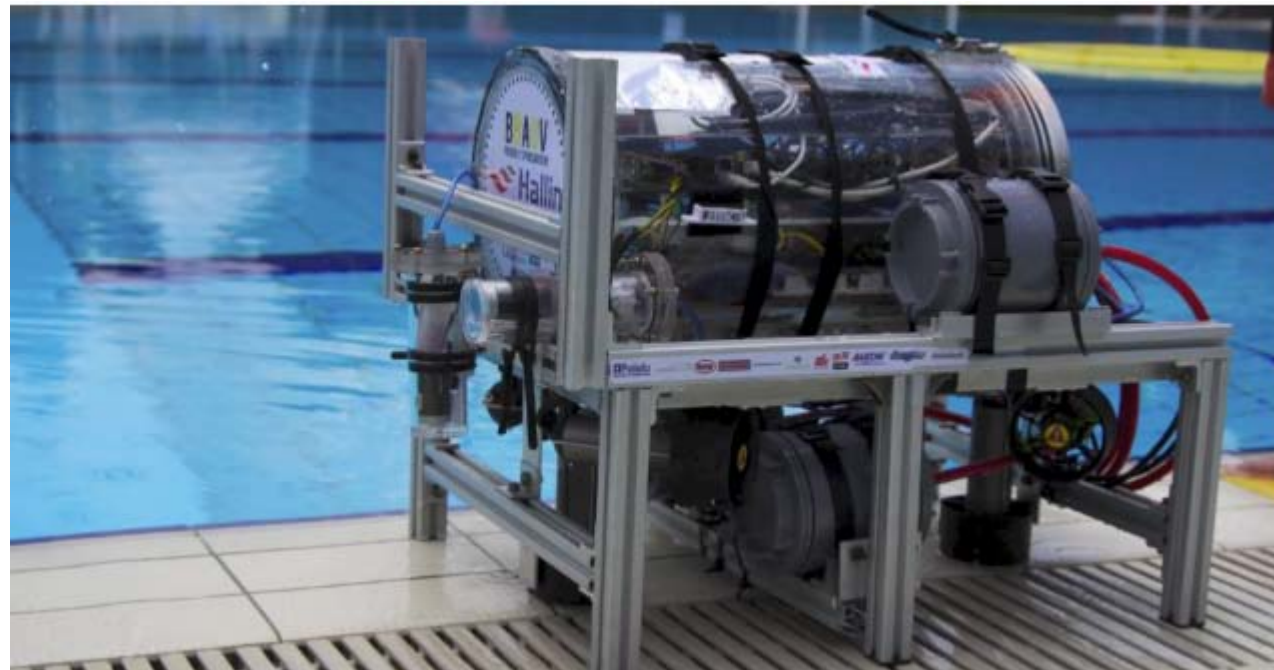
GUI Design

Architecture Diagram And Design  
Patterns

# TARGET USERS

---

The **developers** of an autonomous underwater vehicle (AUV)



# TYPE OF SYSTEM

---

The system is a **stand-alone application**.

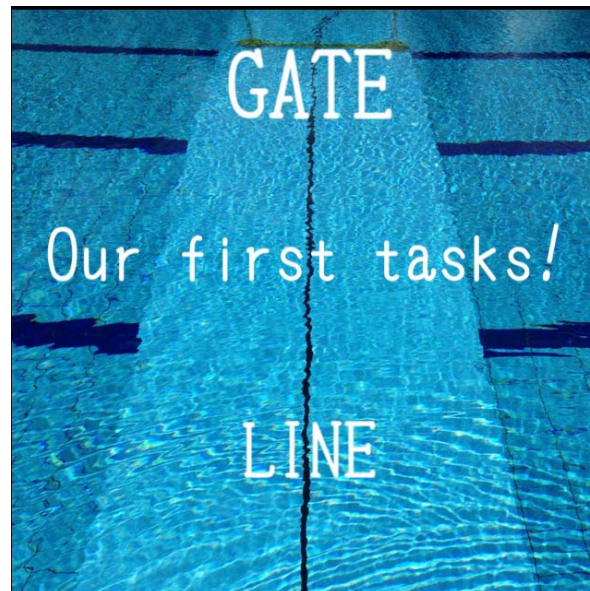


Users will create and test image filters on the desktop before passing the filters on to the AUV.

# WHAT IT DOES

---

The software is intended to allow users to rapidly experiment with various filter chains required in the vision system.

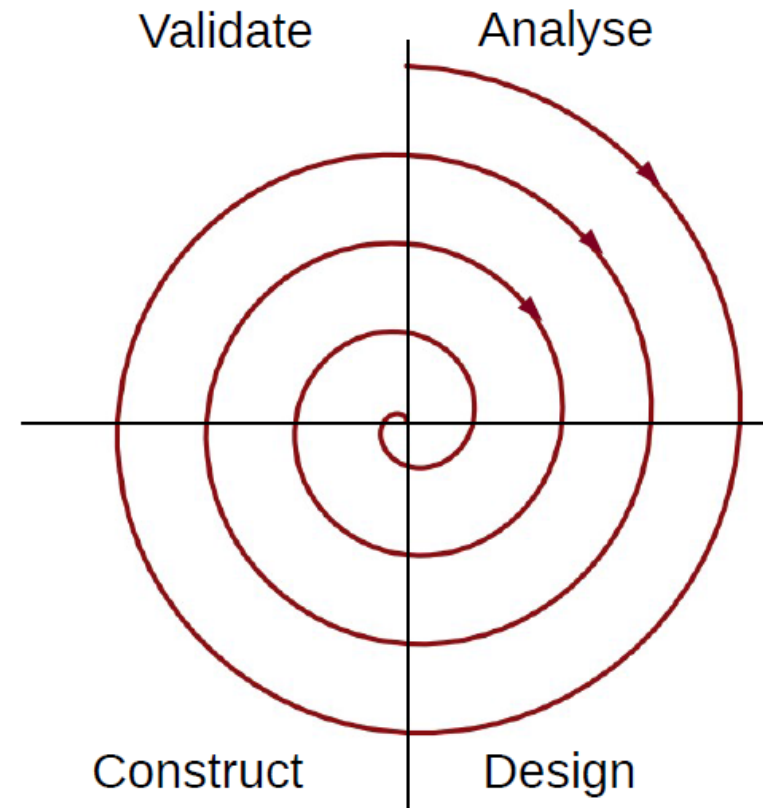
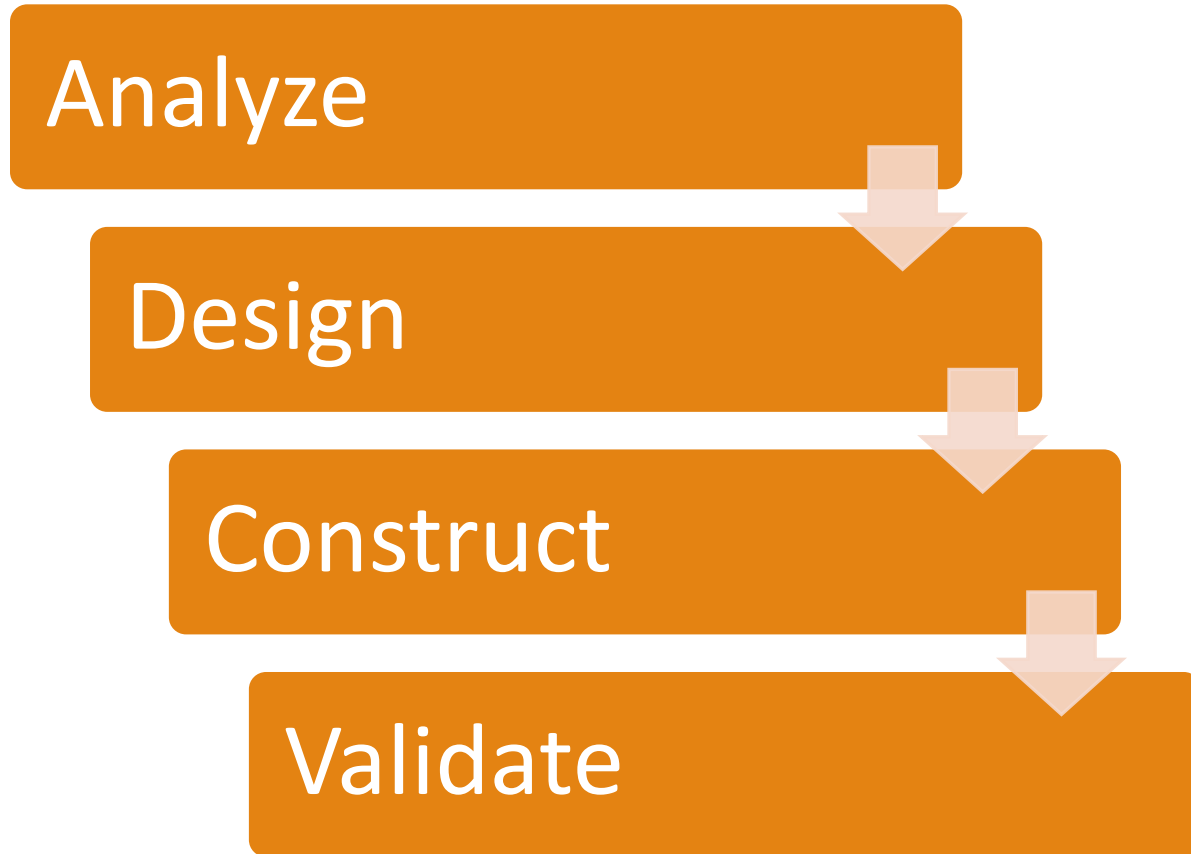


# DEMO

---

# GUI DESIGN PROCESS

---



# MAJOR FUNCTIONALITIES

---

- Video input from camera stream or video file
- Adding of different filters to form a chain of filters
- Selection of video outputs to 2 filter cam widgets
- Changing of settings for filters
- Capture snapshot from camera
- Capture video from camera
- Saving and loading of filter chains
- Keyboard shortcuts



# MAJOR DESIGN PRINCIPLES

---

Tried to balance the **aesthetics** with the **feel** and **functionality** to provide a seamless user experience.

# MAJOR DESIGN PRINCIPLES

---

## Place Users In Control

- Allow users to use either keyboard or mouse
- Display descriptive messages and text

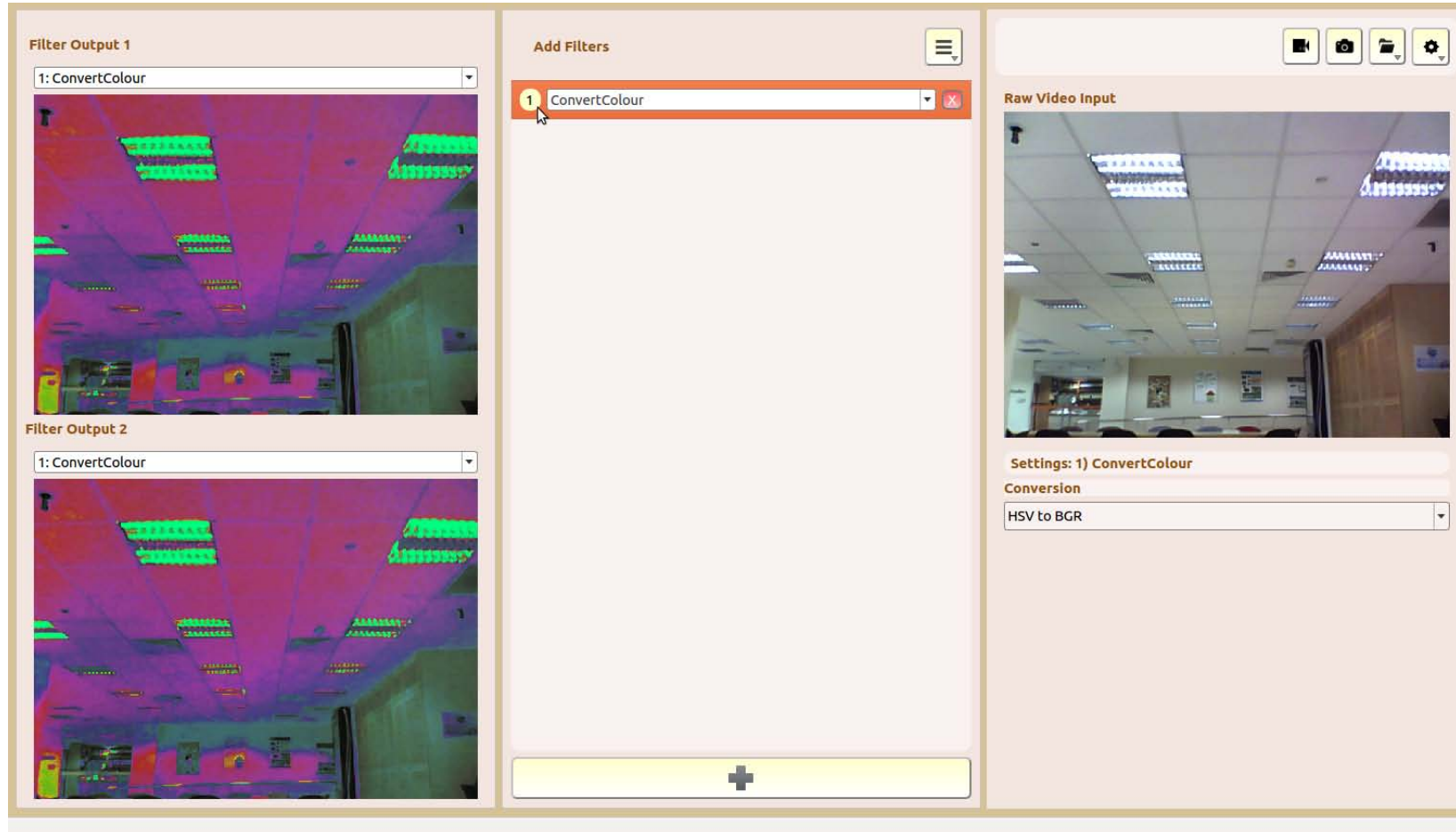
# MAJOR DESIGN PRINCIPLES

---

## Reduce Users' Memory Load

- Relieve short-term memory
- Rely on recognition, not recall
- Promote visual clarity

# SCREEN SHOT #1



# SCREEN SHOT #2

The screenshot displays a video processing application interface with three main panels:

- Filter Output 1:** A dropdown menu shows "1: GaussianBlur". Below it is a video preview window showing a blurred scene of a room with a mannequin.
- Filter Output 2:** A dropdown menu shows "1: GaussianBlur". Below it is another video preview window showing the same blurred scene.
- Add Filters:** A central panel with a list of filters:
  - 1 GaussianBlur (highlighted)
  - 2 ConvertColour
  - 3 ConvertColour
  - 4 ConvertColour
  - 5 ConvertColourEach filter has a red 'X' icon to its right. A large yellow button with a plus sign is at the bottom.
- Raw Video Input:** A panel on the right with a video preview window showing the original, unblurred scene. Below the preview, the settings for the selected filter are shown:

Settings: 1) GaussianBlur

Kernel Size: 5

Sigma: 1.50

# SCREEN SHOT #3

The screenshot displays a video editing software interface with three main panels:

- Filter Output 1:** A dropdown menu shows "1: GaussianBlur". Below it is a video preview window showing a blurred view of a room with a grid ceiling and fluorescent lights.
- Filter Output 2:** A dropdown menu shows "2: Invert". Below it is a video preview window showing the same room as Filter Output 1, but with inverted colors (negative effect).
- Add Filters:** A central panel with a list of filters: 1 GaussianBlur, 2 Invert (highlighted with an orange bar and a mouse cursor), 3 ConvertColour, 4 ConvertColour, and 5 ConvertColour. Each filter has a red 'X' icon to its right. A plus sign (+) is visible at the bottom of this panel.
- Raw Video Input:** A panel on the right with a video preview window showing the original, unprocessed video of the room. Below the preview, it says "Settings: 2) Invert" and "No settings available." There are icons for camera, photo, gallery, and settings at the top of this panel.

# SCREEN SHOT #4

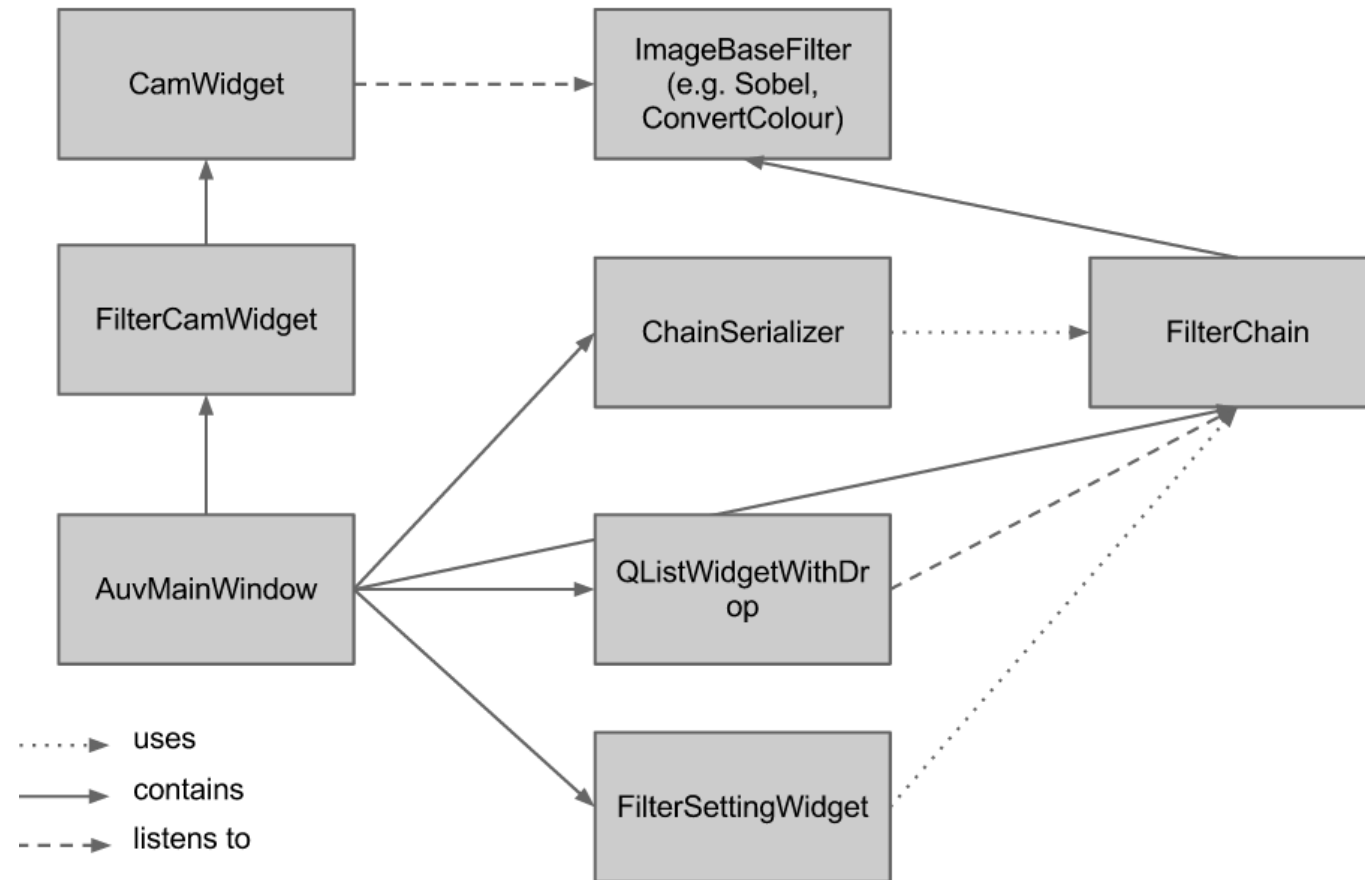
The screenshot displays a video processing application interface with three main panels:

- Filter Output 1:** A video preview window showing a room with a grid ceiling and fluorescent lights. A dropdown menu at the top is set to "1: Invert".
- Filter Output 2:** A second video preview window, identical to the first, also showing the inverted image. Its dropdown menu is also set to "1: Invert".
- Add Filters:** A central panel with a list of filters:
  - 1 Invert
  - 2 ConvertColour
  - 3 GaussianBlur
  - 4 ConvertColour
  - 5 EqualizeHistEach filter entry has a red 'X' icon to its right. A yellow bar with a plus sign is at the bottom of this panel.
- Raw Video Input:** A video preview window showing the original room. Above it are icons for recording, camera, folder, and settings. A tooltip "Record Your Video Stream" is visible over the recording icon. Below the video is a settings section:

Settings: 1) Invert  
No settings available.

# ARCHITECTURE DIAGRAM

---





# DESIGN PATTERNS INVOLVED

---

## Observer Pattern

- Updating Cam Widget Image

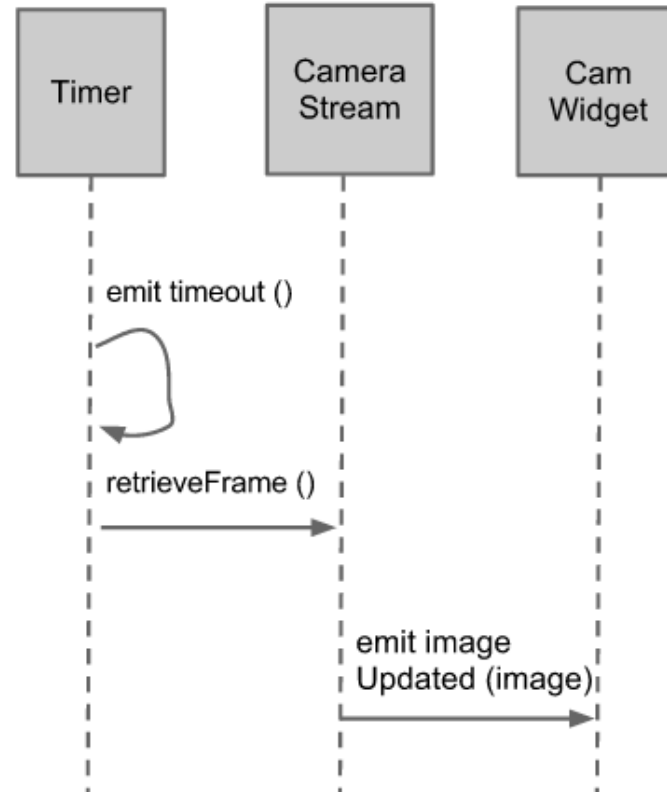
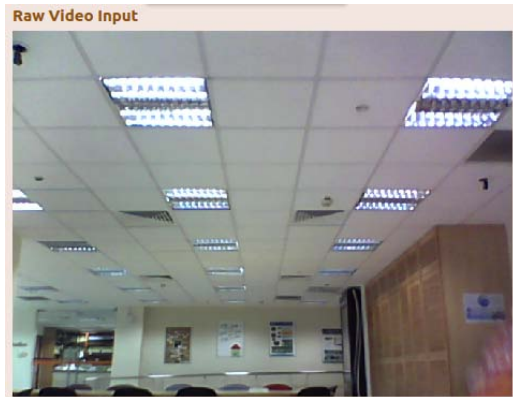
## Classic MVC Examples

- Changing Filter Type in Filter List
- Drag & Drop in Filter List
- Changing Filter Selection in Filter List
- Changing Filter Settings Property

# DESIGN PATTERN

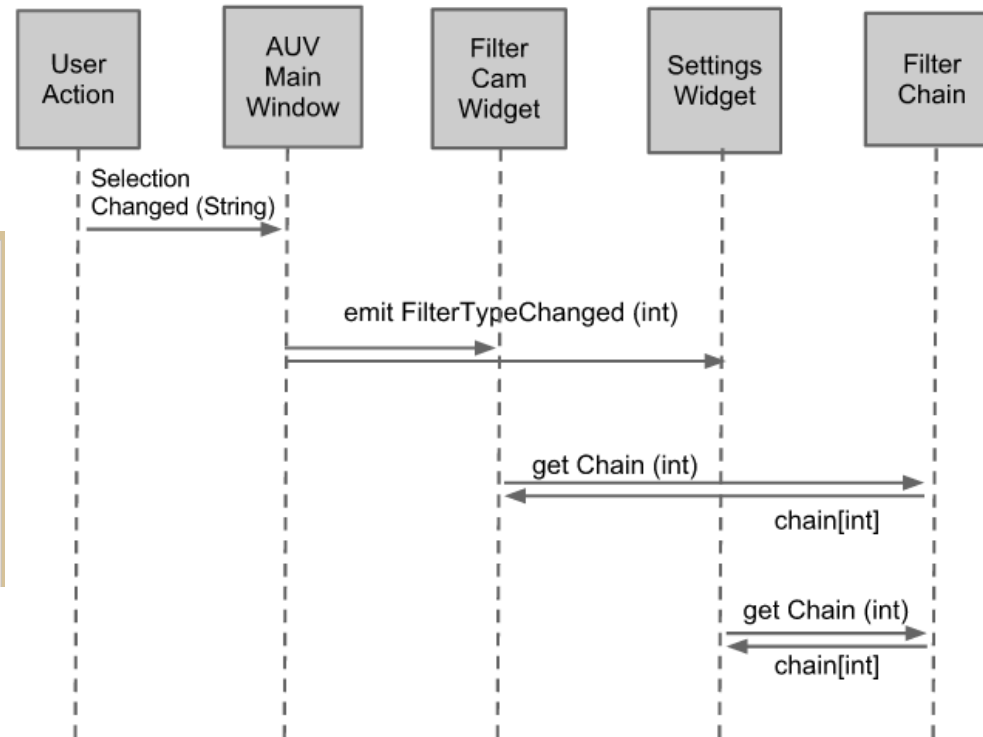
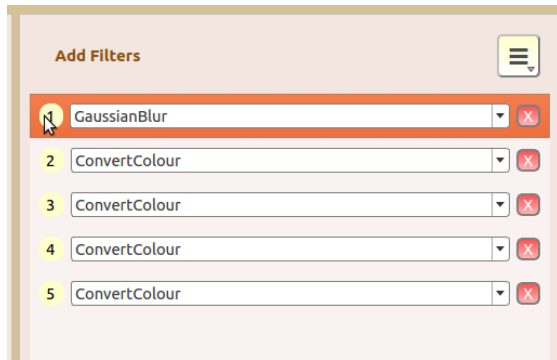
---

## Updating Cam Widget Image (42 FPS)



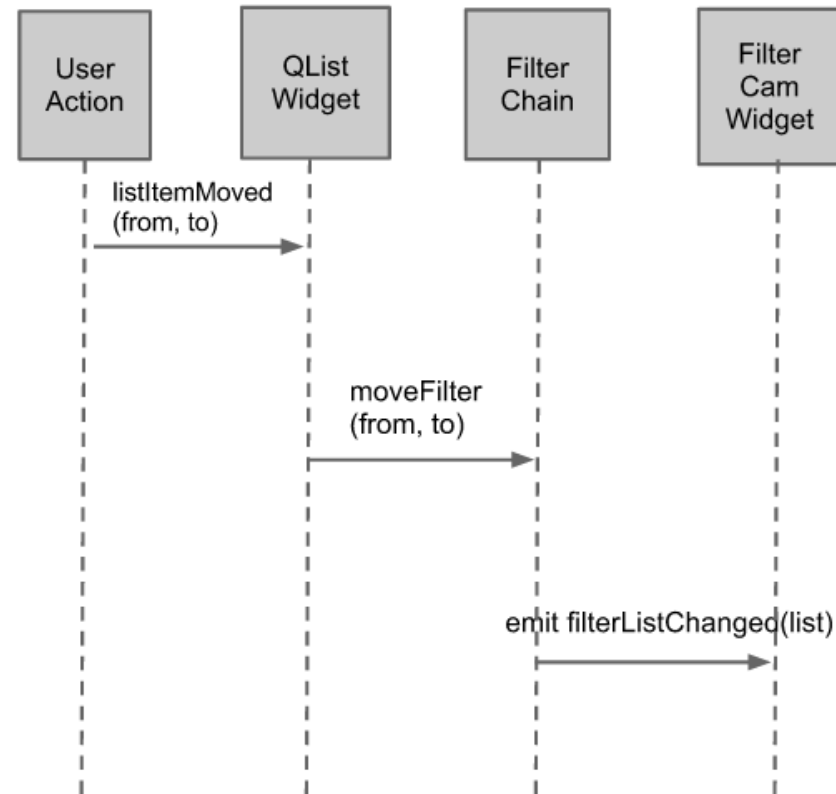
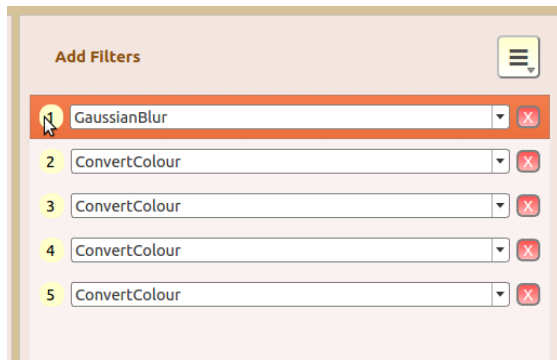
# DESIGN PATTERN

## Changing Filter Type in Filter List



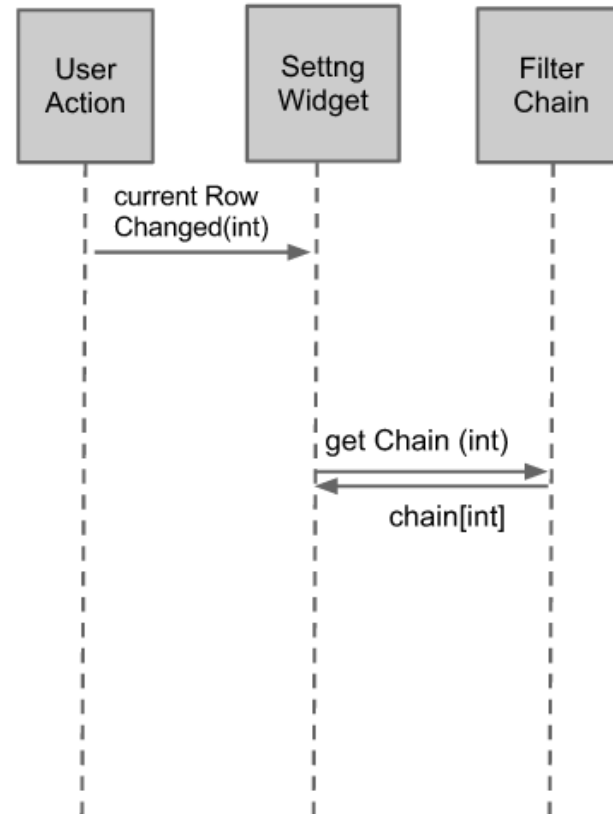
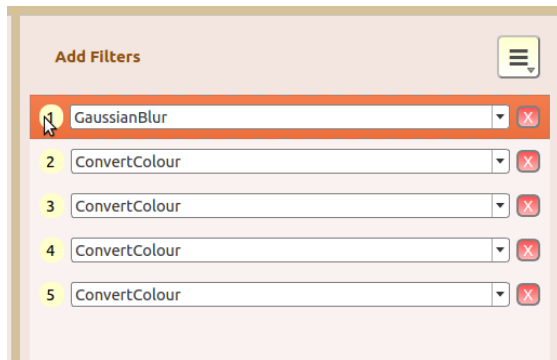
# DESIGN PATTERN

## Drag & Drop in Filter List



# DESIGN PATTERN

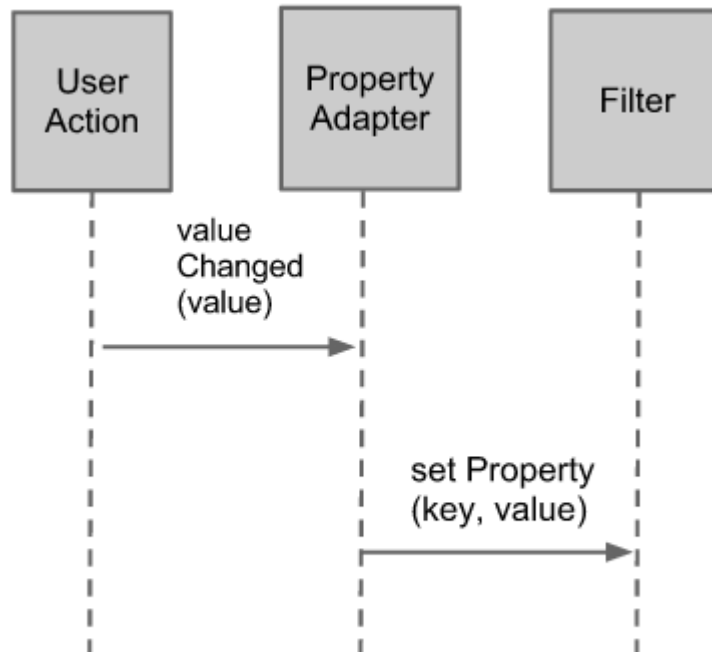
## Changing Filter Selection in Filter List



# DESIGN PATTERN

---

## Changing Filter Settings Property



# IMPLEMENTATION ISSUES

---

## File Dialog

- Has unresponsive behavior on Ubuntu but works perfectly Arch Linux

## Settings Widget Styling

- Background color does not cover entire widget
- Appears only behind sub widgets



# IMPLEMENTATION ISSUES

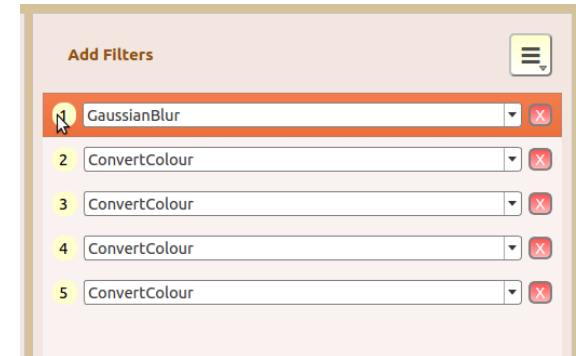
---

## Saving Videos

- Dependent on codecs installed on user system

## QListWidget deleting item

- `currentRowSelected()` emitted before item is deleted
- Item gets deleted and items after it shifts up
- `currentRowSelected()` does not get called again
- Slots received takes in the wrong index
- Need to emit additional signal for settings to update itself





# IMPLEMENTATION ISSUES

---

## Supporting Libraries

### OpenCV

- Transforming video outputs
- Used By: ImageFilterBase

### Jsmn

- Json Parser
- Persistence
- Used By: ChainSerializer

**THANK YOU!**

---