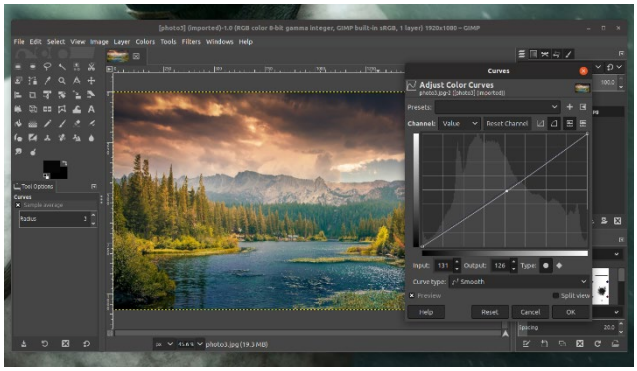
# A2 - Making an Image Editor (20%)

**OVERVIEW**

Using *OpenFrameworks*, create an experience that allow one to modify a photo using various image processing techniques.



**Required**

- A UI that allows one to:
  - Modify brightness, contrast, saturation, and RGB colour balance.
  - Apply (4) original filters.
  - Select and display a colour.

**Recommended**

- Explore various methods to increase and/or hide performance issues (e.g. using pointers and/or adding in visual feedback).
- Explore a self-contained application that can load in and save modified images with.
- Consider creating your own custom buttons, sliders, toggles, drawing tools etc.
- Explore others methods of applying filters i.e. LUTs (Look Up Tables), Deep learning etc..

**Examples**

- Photoshop / GIMP
- Instagram filters and/or image annotators

If you have an idea that is outside of the listed requirements, please talk to me.

**SUBMISSION**

Each student will submit a "***lastnameFirstname_IMD3005_A2.zip***" via CULearn by the deadline specified. This zip will contain your OpenFrameworks Project, and it **should work from within the "OPENFRAMEWORKS_PATH/apps/" folder**.

The following components of this assignment are expected:

1. A readme.txt of (3) paragraphs describing:
   - Overview of what you did (i.e. what are the controls? Why this design?)
   - What was challenging.
   - What went well (i.e. how did you solve the above challenges?).
2. Openframeworks project zipped with only the '/src' and '/bin/data folders' included (please delete all other files/folders).
3. Between 10s - 30s .mp4 video and/or .GIF showcasing the working project.

NOTE: *Late policy defined by class policies on CULearn . Plagiarism policies will follow Algonquin College's regulations found here; [http://www.algonquincollege.com/policies/policy/plagiarism/](http://www.algonquincollege.com/policies/policy/plagiarism/)*

**LEARNING OUTCOMES**

This assignment aims to educate in the following ways:

- To plan and develop an application that uses the image processing techniques for a practical purpose.
- To understand, through experimentation, some of the advantages of disadvantages of per-pixel image processing techniques.
- To further build on current knowledge of Openframeworks, sensor-based interaction, and C++ principles.

**EVALUATION RUBRIC**

| Outcomes | Exemplary | Proficient | Sufficient | Poor |
|---|---|---|---|---|
| **Creativity** (30%) | The overall experience is very unique, creative, and well-developed. The provided project constraints are used in an unexpected way to create a very original experience. | The overall experience is unique and creative. The end result is original and makes good use of the constraints imposed by the project. | The overall concept is creative, though takes an obvious everyday direction. Could use a little more thought to make it into something extra-ordinary. | The overall experience lacks creativity and originality. An execution with little thought. |
| **Interaction** (30%) | The interactions are very responsive. Their use is both original and intuitive so that little to no instructional training is required to use the experience. | The interactions are responsive. Their use is original and mostly intuitive; however, some instructional training is required to use the experience. | The interactions are mostly clear and responsive, however can be difficult to use and understand at times. | The interactions rarely work, and their influence to the on-screen result is not completely clear. |
| **Code** (10%) | A high level of technical complexity is used to implement advanced features and/or creatively solve difficult technical problems that are well beyond the basic requirements. This is very evident by looking at the source code. | An above average level of technical complexity is used to implement features and/or creatively solve difficult technical problems that are beyond the basic requirements. This is evident by looking at the source code. | An average level of technical complexity is evident in the project. The technical problems solved are the basic requirements of the project. The source code demonstrates an understanding of the basic required subject matter. | A poor level of technical complexity is evident in the project. The basic problems required by the project are not solved and/or the source code demonstrates a poor understanding of the basic required subject matter. |
| **Presentation** (30%) | Great effort has been put into making the on-screen portion interesting and visually pleasing. The visual elements seem as though they are a part of the same experience. | Good effort has been put into making the on-screen portion interesting and visually pleasing. Some further attention to detail would add polish to the experience. | Some effort has been put into making the on-screen portion interesting and visually pleasing. Could use more fine-tuning. | The on-screen portion lacks serious effort to make it visually interesting and feels incomplete/ unfinished. |