CSE 4310: Introduction to Computer Vision
Summer 2021
Program #1: Raster graphics editor

In this assignment, you will create a primitive image manipulation program using OpenCV. The goal is to become familiar with basic OpenCV data structures and manipulation, such as pixel access, ROIs, copying, visualization, etc.

The program should take a single command line argument, which will contain the file path for the input image (assumed to be a 3 channel color PNG). The program will only operate on a single input image when executed (i.e., manipulating a different image will require a restart).

Once the program is loaded, a few basic image editing tools will be implemented. Right clicking in the display window should toggle through the list of available tools. The program console should display the currently activated tool, but you don't need to change the appearance of the cursor or display any buttons / GUI controls. The tools to be implemented are as follows:

> **EYEDROPPER**: When active, a left mouse click will change the current color value stored in memory (we will call this the "eyedropper" value). The eyedropper value should be initialized to white (255, 255, 255) when the program is loaded, and the BGR values should be printed to console when it is changed.
>
> **CROP**: When active, dragging the left mouse button from a click location to a release location should define a rectangular area which will then be immediately cropped. The cropped area should replace whatever is currently displayed in the image window, and the window should automatically resize to the cropped area.
>
> **PENCIL**: When active, any left mouse clicks or left mouse button drags should change the color of the target pixel to the current eyedropper value.
>
> **PAINT BUCKET**: When active, a left mouse click at a pixel location with color *X* shall be changed to the eyedropper value. Any of that pixel's 4-connected neighbors (top, bottom, left, right, but NOT diagonals) with color value X shall also be changed to the eye dropper value, and those pixel's neighbors shall be evaluated and colored accordingly. This process should repeat until no new suitable connected pixels can be added to the fill area. The result of this tool should be that a continuous 4-connected blob of a single color in an image, when clicked on, should be changed to the eyedropper value while no other pixels should be changed.
>
> **RESET**: When active, a left mouse double click in the image window will replace the window contents with the original, unedited image as it was when the program was initially loaded.

Your program should run on the class development environment using OpenCV 3. If you implement your program using C++ (highly recommended), you must include a working

CMakeLists.txt file with your source code. The program should be able to be compiled and executed by running the following set of commands in the program directory if C++ is used:

```
cmake .
make
./program1 <PATH TO FILE>
```

For example…

```
cmake .
Make
./program1 test.png
```

You may choose to use python for the assignment, though this is not recommended. Your program should then be able to run in the class development environment with the following set of commands in the program directory:

```
python program1 <PATH TO FILE>
```

For example…

```
python program1 test.png
```

You may develop your application on your own system / OS, but it must run properly on the class development environment at the time it is evaluated. Submit your source code and all necessary files in a single zip file by the deadline. Late submissions will incur a penalty of 10 points per day after the deadline.

Points will be assigned as follows:

1. Program accepts input correctly and displays image – 10 points
2. Program implements EYEDROPPER feature successfully – 20 points
3. Program implements CROP feature successfully – 20 points
4. Program implements PENCIL feature successfully – 20 points
5. Program implements PAINT BUCKET feature successfully – 20 points
6. Program implements RESET feature successfully – 10 points

Partial credit may be given, at the discretion of the grader, for items which are not fully functional or contain bugs. You may be asked to demo your program to the grader if errors occur during the initial run. Write your code as cleanly as possible with proper formatting and comments in order to bolster your case for partial credit on non-functional features. It is highly recommended to follow a standard coding style, such as ANSI C++, Google C++, etc. in your program (most modern C++ IDEs have built in formatters that can do this automatically).