LW: Image Manipulation (Rule of Three)

Overview

Objectives

- Address problems when using dynamic memory in classes (OOP).
- Implement the rule of three to avoid memory leaks and ensure deep copy.

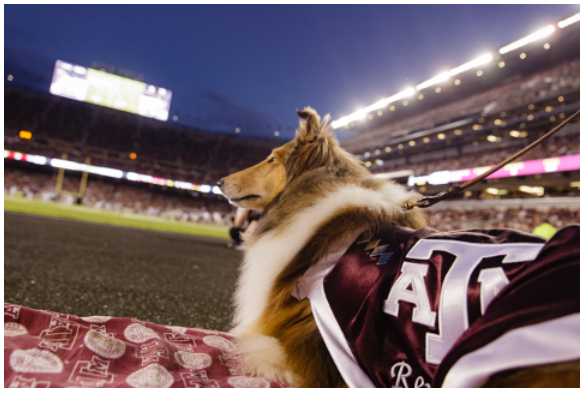
Getting Started

- Download and read over the starter code (look at `Image.h`).
- Read over the introduction and requirements.
- Implement the rule of three for the `Image` class.
- **Rule of 3 says that all three (destructor, copy constructor and assignment) must be implemented. But you can earn credit for the lab by implementing 2 of the 3. (50 / 100)**
- Implement copy constructor and assignment first to earn 50 points. The remaining 50 points will come from correct implementation of destructor to remove memory leaks after correct implementation of copy constructor and assignment.
- You are expected to have a foundational understanding in the rule of three to complete this lab work, the steps are somewhat universal.

Introduction

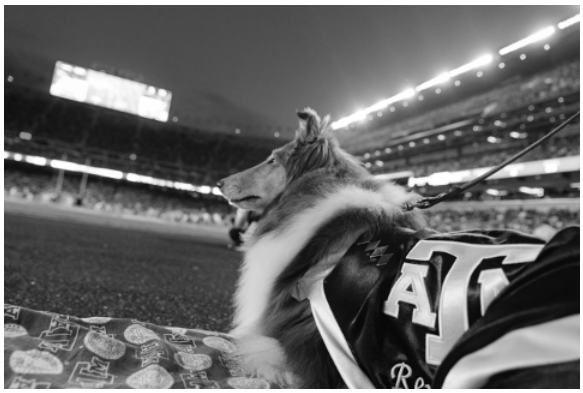
After learning all about classes, your friend wrote a program to put image information into classes. They created an image class and converted the pixel struct into a proper class. As you run and play with the program you notice some peculiarities.

- First you load this image of reveille.[1]

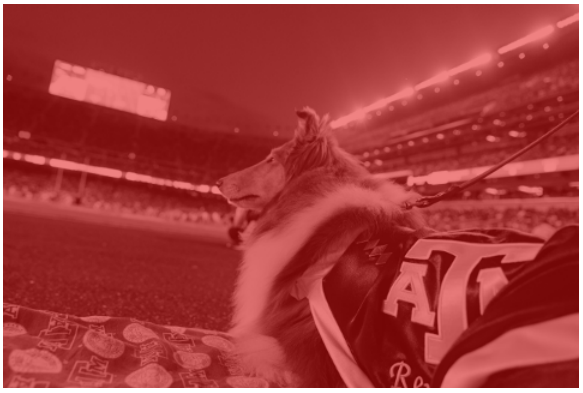


- Then you convert it to grayscale

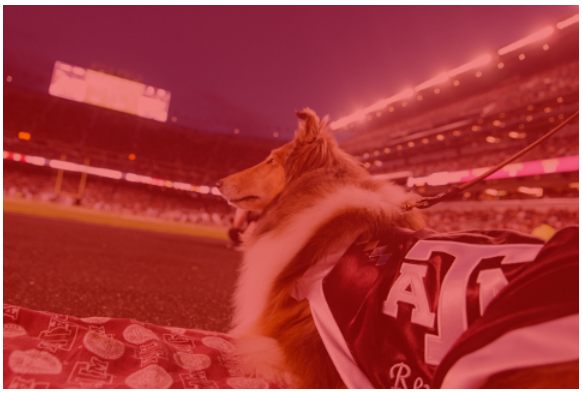
[1] Image taken from https://tamu.photoshelter.com/gallery-image/Reveille/G0000095v1nAlxlLA/I0000pjQsaUrGSxY/C0000plgky6s4_tk



- Then you add the Color (225, 5, 5).



- This seems pretty cool. You exit the program and decide to play with it some more. So you load the image of Reveille again. You decide to replicate the colored image you had before, so you add the Color (255, 5, 5) again.



- You notice that there is evidence of other colors besides red whereas last time it was all shades of red. You suspect something may be wrong with the program. You realize that your friend put the image on the heap, so **maybe a shallow copy is the culprit**.
- On top of all that, you find out that **there are memory leaks in the program**. You see a clear() function that could be called from a destructor, but you suspect that there is a problem with it that needs to be fixed.

Requirements

You will only need to edit `Image.h` and `Image.cpp`. You will need to **implement the rule of three for this class**, as well as the `clear()` function.