CS430 Computer Graphics

Project 1 - Images

In this project you will write code to read and write images in PPM format. Your code should be able to read and write PPM files in P3 and P6 format.

Your program should be resistant to errors and should not segfault or produce undefined behavior. If an error occurs it should print a message to stderr with “Error:” prefixed to a descriptive error message before returning a non-zero error code. I have a test suite designed to test the robustness of your program.

Your program (ppmrw) should have this usage pattern to convert a P3 or P6 image to P3:

```
ppmrw 3 input.ppm output.ppm
```

Your program (ppmrw) should have this usage pattern to convert a P3 or P6 image to P6:

```
ppmrw 6 input.ppm output.ppm
```

You should print an error if the image would not be 8 bit per channel.

Technical Objectives

Technical objectives describe the organizational or code-related features that are a required part of your game and will be evaluated in the technical objective rubric for this project. In grading technical objectives, we will ask the question “How well does this project provide evidence of the objective?” For example, a single Git commit probably does **not** represent outstanding “use of git”.

- Ability to read P3 formatted images into memory
- Ability to write to P3 formatted images from memory
- Ability to read P6 formatted images into memory
- Ability to write to P6 formatted images from memory
- Use of C programming language
- Use of consistent coding style and commenting
- Use of Git

Creative Objectives

Creative objectives mirror the technical objectives but involve subjective creative features of your project. In this project you will need a demonstration image. Use a paint or illustration program to compose this image. For this project the creative objective has very little weight but I still encourage you to consider these objectives:

- Uses a range of colors
- Makes the correct orientation of the image obvious (which way is up, left, right, etc.)
- Visually interesting

What do I turn in?

In BBLearn you should turn in a report (entered with the BBLearn editor) with the following format:

> Project # and Title
> Your Name
> Your NAU User ID
>
> Link to Github Repository

Your Github repository should contain all the code for your repository. It should also include a README.md file that describes your application, usage, and communicates any special notes to the grader. A Makefile which can be used to build your project. Your program should compile with gcc without any special libraries (libc and libm are ok).

The grading rubric is posted in BBLearn.

Graduate Student Extension (CS599)

If you are a graduate student, you should also implement P7 such that your program (ppmrw) supports this usage pattern to convert a P3, P6, or P7 image to P7:

```
ppmrw 7 input.ppm output.pam
```

You should print an error if the image would not be 8 bit per channel or not RGB(A). You should be able to support RGBA.