Information About PPM Files

In graphics, you often want to display something interactively on-screen. However, you often want to instead store your image in a file for viewing later. Therefore, it is necessary to have some accepted file formats to store this data. Common graphics formats include Windows' bitmaps (.BMP), the Joint Photographic Experts Group's format (.JPG), the Graphics Interchange Format (.GIF), and Portable Network Graphics files (.PNG). Unfortunately, these files have built-in compression, complex headers, or odd internal formatting. For this class, we will instead be using portable pixmap (.PPM) files, which are extraordinarily simple.

File Description

As with most graphics file formats, PPM files have two parts: a header (describing the type and amount of data in the file) and the data itself.

The header has three lines of data, but may have additional comment lines beginning with the character '#'.

The first line of the header consists of two characters describing the type of the file. This line must be the first line in the file (a comment line cannot proceed it). For this class, you will only need to use "P3" files, so the rest of this handout will focus on them.

- "P1" images have 1 bit per pixel (black and white) stored in ASCII format.
- "P2" images have 8 bits per pixel (greyscale) stored in ASCII format.
- "P3" images have 24 bits per pixel (8 bits for each red, green, and blue) stored in ASCII format.
- "P4" images have 1 bit per pixel (black and white) stored in binary format.
- "P5" images have 8 bits per pixel (greyscale) stored in binary format.
- "P6" images have 24 bits per pixel (8 bits for each red, green, and blue) stored in binary format.

The second line of the header has two ASCII number, separated by spaces. These two numbers represent the width and height (respectively) of the stored image.

The third line of the header has one ASCII number. This number is the maximum value any color can take. For this class, since you will be using 8-bits per color channel, this value should be "255". This means that a value of $(R, G, B) = (0, 0, 0)$ will be black and $(R, G, B) = (255, 255, 255)$ will be white. If this line instead had the number "137", the value $(R, G, B) = (137, 137, 137)$ would be white, and the value $(R, G, B) = (200, 200, 200)$ would give undefined results.

After the header, the remaining lines of the file store image data. Each pixel is stored as three numbers, separated by whitespace. The first number for each pixel is the red value, the second number is the green value, and the third number is the blue value. You *may* combine multiple pixels on a line, but each line must be no longer than 70 characters. The pixels are specified starting with the upper-left pixel, proceeding from left-to-right, then top-to-bottom (i.e., the same order you read English text).

Simple Example

The following example is an image of width five (5), and height four (4), with a red pixel in the upper-left corner, a green pixel in the upper-right corner, a blue pixel in the lower-left corner, a white pixel in the lower-right pixel, with other pixels black.

```
P3
# Example  5 x 4  image
5 4
255
255 0 0     0 0 0    0 0 0    0 0 0    0 255 0
0 0 0       0 0 0    0 0 0    0 0 0    0 0 0
0 0 0       0 0 0    0 0 0    0 0 0    0 0 0
0 0 255     0 0 0    0 0 0    0 0 0    255 255 255
```