Christopher Biddle, Craig Fox

Mr. Patel

1/5/2017

Image Filter Project

Instructions

To run the program, click the green play button at the top left of the Processing window. A window should pop up. Click the “*Load Picture*” button at the top right of the window and select the image “*Patel.jpg*” in the same folder as the program.

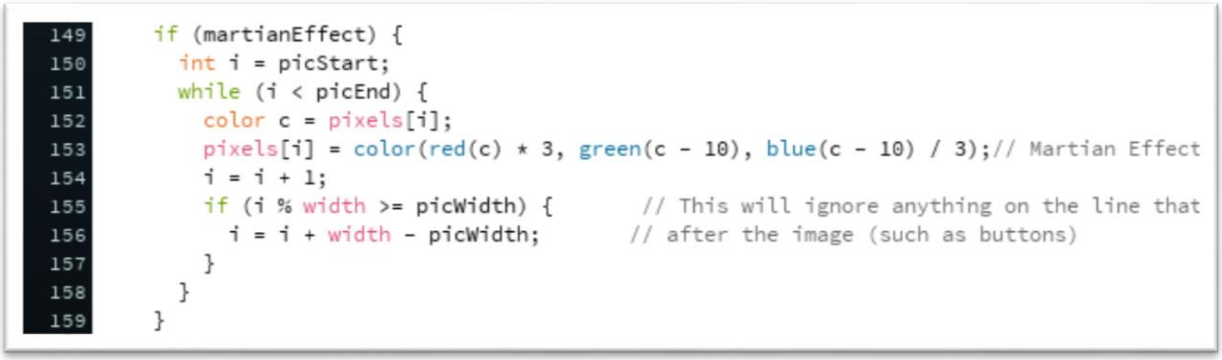
Click any of the “*Filter Effects*” buttons on the right side of the window to alter the image.

To save the image, click the “*Save Picture*” and choose a destination to save the new photo at.

Effects

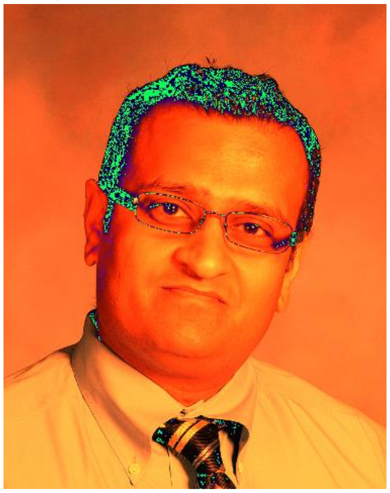
Martian Effect

CODE SNIPPET



```
if (martianEffect) {
  int i = picStart;
  while (i < picEnd) {
    color c = pixels[i];
    pixels[i] = color(red(c) * 3, green(c - 10), blue(c - 10) / 3);// Martian Effect
    i = i + 1;
    if (i % width >= picWidth) {       // This will ignore anything on the line that
      i = i + width - picWidth;       // after the image (such as buttons)
    }
  }
}
```

Output



The martian effect takes each pixel, seperates its RGB value, and applies a certain effect to each set.

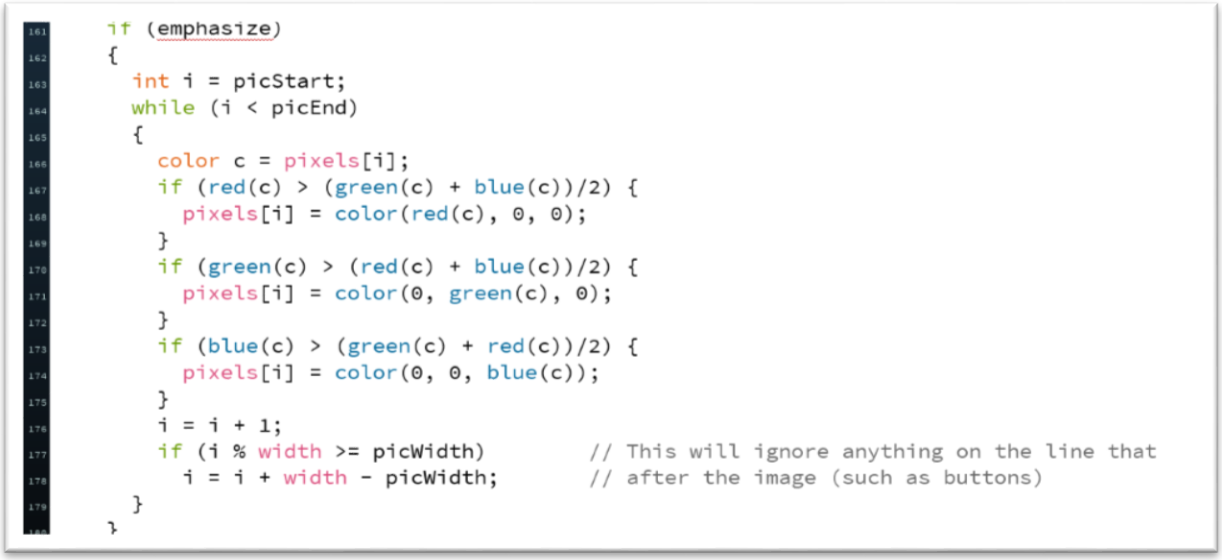
The effect saturates the red part of the pixels by multiplying its value by 3.

It reorders the tenth previous pixel’s green value to be the value of the current pixel. This allows for the blue values in the dark parts of the image to multiply and become a new alien green color in the hair. The effect also changes the background color.

Lastly, the effect reorders the tenth previous pixel’s blue value to be the value of the current pixel. It then divides the blue value by 3. This changes the color of the image by making it a martian orange.

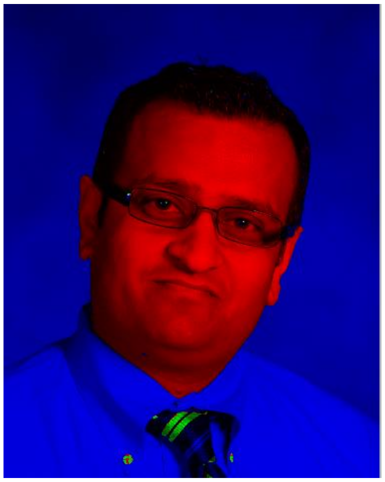
Emphasize Effect

CODE SNIPPET



```
if (emphasize)
{
  int i = picStart;
  while (i < picEnd)
  {
    color c = pixels[i];
    if (red(c) > (green(c) + blue(c))/2) {
      pixels[i] = color(red(c), 0, 0);
    }
    if (green(c) > (red(c) + blue(c))/2) {
      pixels[i] = color(0, green(c), 0);
    }
    if (blue(c) > (green(c) + red(c))/2) {
      pixels[i] = color(0, 0, blue(c));
    }
    i = i + 1;
    if (i % width >= picWidth)        // This will ignore anything on the line that
      i = i + width - picWidth;       // after the image (such as buttons)
  }
}
```

Output



The emphasize effect emphasizes either the red, green, or blue in a color.

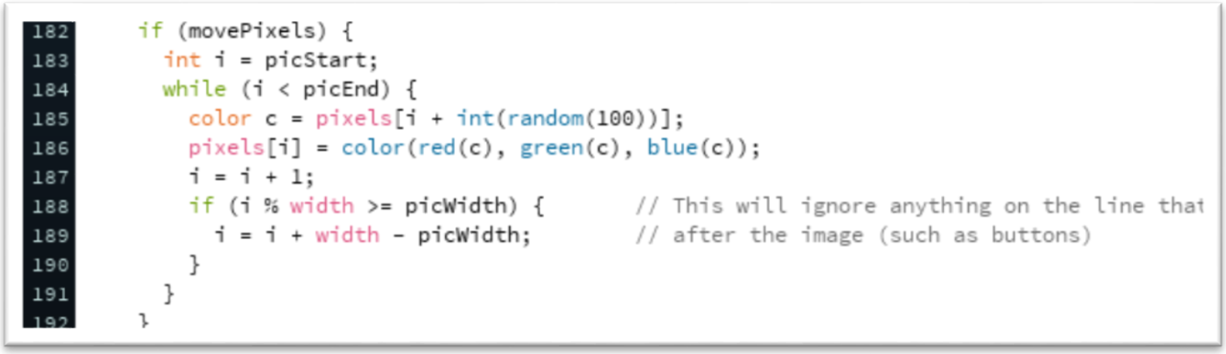
The emphasize effect will check to see if any pixels have a red, green, or blue value greater than the avergit age of the other two colors.

If they are equal, the current pixel will have a value of 0 for the other two colors.

For example if a pixel has a red value of 234, a green value of 34, and a blue value of 76, the pixel would become (234, 0, 0) because 234 is greater than the average of 34 and 76.

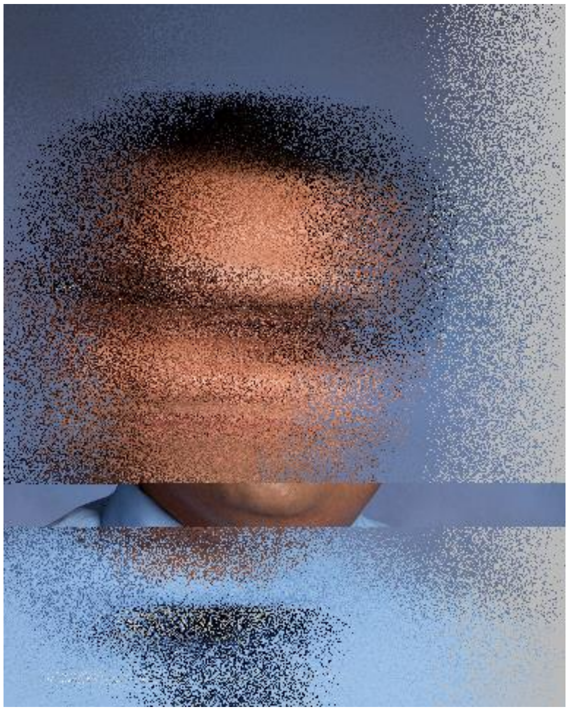
Move Pixels

CODE SNIPPET



```
182     if (movePixels) {
183       int i = picStart;
184       while (i < picEnd) {
185         color c = pixels[i + int(random(100))];
186         pixels[i] = color(red(c), green(c), blue(c));
187         i = i + 1;
188         if (i % width >= picWidth) {       // This will ignore anything on the line that
189           i = i + width - picWidth;        // after the image (such as buttons)
190         }
191       }
192     }
```

Output



This effect randomly moves the pixels zero to ten places forward every screen refresh.

This is accomplished by putting "*i + int(random(10))*" inside "*pixels[]*" on line 185.