

Manipulating Pictures, Arrays, and Loops part 6

Barb Ericson
Georgia Institute of Technology
August 2005

Georgia Institute of Technology

Learning Goals

- Understand at a conceptual and practical level
 - How to change all the colors in a method
 - How to use a for loop
 - How to negate a picture
 - How to turn a picture into grayscale
 - How to adjust the grayscale based on how we perceive colors

Georgia Institute of Technology

Negating an Image

- How would you turn a picture into a negative?

- White should become black

- 255,255,255 becomes 0,0,0

- Black should become white

- 0,0,0 becomes 255,255,255



Georgia Institute of Technology

Negate Algorithm

- Subtract current value from 255 for red, green, and blue
 1. Get the array of pixels from the picture
 2. Declare variables to hold the current pixel and the red, green, and blue values
 3. Loop starting an index at 0 and incrementing by 1 and loop while the index is less than the length of the array
 1. Get the pixel at the current index from the array of pixels
 2. Set the red value to 255 - current red value
 3. Set the blue value to 255 - current blue value
 4. Set the green value to 255 - current green value
 5. Increment the index and go back to step 3

Georgia Institute of Technology

Negate Method

```
/**
 * Method to negate the picture
 */
public void negate()
{
    Pixel[] pixelArray = this.getPixels();
    Pixel pixelObj = null;
    int redValue, blueValue, greenValue = 0;



    // loop through all the pixels
    for (int i = 0; i < pixelArray.length; i++)
    {
        // get the current pixel
        pixelObj = pixelArray[i];

        // get the values
        redValue = pixelObj.getRed();
        greenValue = pixelObj.getGreen();
        blueValue = pixelObj.getBlue();

        // set the pixel's color
        pixelObj.setColor(
            new Color(255 - redValue,
                    255 - greenValue,
                    255 - blueValue));
    }
}
```

Georgia Institute of Technology

Changing to Grayscale

- Grayscale ranges from black to white
 - The red, green, and blue values are the same
- How can we change any color to gray?
 - What number can we use for all three values?
 - The intensity of the color
 - We can average the colors
 - $(\text{red} + \text{green} + \text{blue}) / 3$ 
 - Example
 - $(15 + 25 + 230) / 3 = 90$ 

Georgia Institute of Technology

Grayscale Algorithm

- Set color values to the average of the original values
 - Get the array of pixels from the picture
 - Declare variables to hold the current pixel and the red, green, and blue values
 - Loop starting an index at 0 and incrementing by 1 and loop while the index is less than the length of the array
 - Get the pixel at the current index from the array of pixels
 - Calculate the average of the current values
 - $(\text{redValue} + \text{greenValue} + \text{blueValue}) / 3$
 - Set the red value to the average
 - Set the blue value to the average
 - Set the green value to the average
 - Increment the index and go to step 3

Georgia Institute of Technology

Grayscale Method

```
/**
 * Method to change the picture to
 * gray scale
 */
public void grayscale()
{
    Pixel[] pixelArray = this.getPixels();
    Pixel pixelObj = null;
    int intensity = 0;

    // loop through all the pixels
    for (int i = 0; i < pixelArray.length;
        i++)
    {
        // get the current pixel
        pixelObj = pixelArray[i];

        // compute the average intensity
        intensity =
            (pixelObj.getRed() +
             pixelObj.getGreen() +
             pixelObj.getBlue()) / 3;

        // set the pixel color
        pixelObj.setColor(new
            Color(intensity,intensity,intensity));
    }
}
```

Georgia Institute of Technology

Testing Grayscale

- String file =
 - "c:/intro-prog-java/mediasources/caterpillar.jpg";
- Picture pictureObj = new Picture(file);
- pictureObj.explore();
- pictureObj.grayscale();
- pictureObj.explore();

Georgia Institute of Technology

Grayscale Result



Georgia Institute of Technology

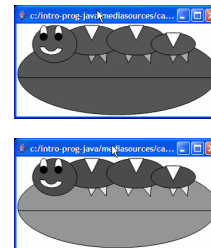
Luminance

- We perceive blue to be darker than red, and green
 - Even when the same amount of light is reflected
- A better grayscale model should take this into account
 - Weight green the highest ($* 0.587$)
 - red less ($* 0.299$) and
 - blue the very least ($* 0.114$)

Georgia Institute of Technology

Grayscale with Luminance Exercise

- Create a new method grayscaleWithLuminance
- Using the new algorithm for calculating intensity
- $\text{intensity} = (\text{int}) (\text{red} * 0.299 + \text{green} * 0.587 + \text{blue} * 0.114)$



Georgia Institute of Technology

Testing Grayscale with Luminance

- String file =
- "c:/intro-prog-java/mediasources/caterpillar.jpg";
- Picture pictureObj = new Picture(file);
- pictureObj.explore();
- pictureObj.grayscaleWithLuminance();
- pictureObj.explore();

Georgia Institute of Technology

Summary

- You can change all the colors in a method
 - By creating a new color object
- You can negate a picture
 - By creating a new color with (255 - red, 255 - green, 255 - blue)
- You can create a grayscale picture
 - By setting the red, green, and blue to the same value
 - You can improve this by weighting the colors based on how we perceive them

Georgia Institute of Technology