Media Computation:
Fun with Images!
What You Might Do
What You Might Do
http://techgirls.cs.vt.edu
download (scenario):
CreativeImages
Starter Scenario
Starter Scenario
Starter Scenario
How Does Color Vision Work?

- Our retinas use rod cells and cone cells
- Rods aren’t sensitive to color
- Cones allow us to see in color
- Three types of cones
In computing, we represent all colors using a combination of red, green, and blue light.

Abbreviated: R G B

White: maximum intensity red, green, and blue

Black: the absence of light = zero red, green, and blue

All other colors are combinations of different intensities of R, G, and B.
How do Digital Cameras Work?

- There are red, green, and blue filters that capture the amount of each color at a position
  - A part of a grid
- There are many positions
  - Picture element or **pixel**
  - 640 x 480 is low resolution
  - 1600 x 1200 is high resolution
- The more pixels, the better the picture
  - Can enlarge it without it looking grainy
for each pixel

```
for (Pixel p : pic.getPixels())
{
    // code applied to every pixel in the image
}
```
Pixel methods

We use numbers from $0 \ldots 255$ to represent intensity

- `getRed()`
- `getGreen()`
- `getBlue()`
- `getAlpha()`
- `setRed(...)`
- `setGreen(...)`
- `setBlue(...)`
- `setAlpha(...)`
First try: changing every pixel!

```java
for (Pixel p : pic.getPixels())
{
    p.setGreen(255);
}
```
Activities

- Create `maxRed()` and `maxBlue()` methods and try them out!
- Create a `grayScale()` method. For each pixel:
  - Calculate an **average** of the red, green and blue values for that pixel
  - Set the red, green and blue values for that pixel to be the average
- Experiment with your own ideas for methods!
Suppose we want to **brighten** a picture by 20%?

- We want to increase the RGB intensity!
- Increase all three color components: R, G, and B
- We can represent the percentage as a decimal fraction
  - new red = old red * (1 + 0.20)
  - new green = old green * (1 + 0.20)
  - new blue = old blue * (1 + 0.20)
What if we want a different percentage?

- To change the brightness by different levels?
- Let’s use a parameter!
Create an invert() method that inverts the red, green, and blue values. For example:

- if the red value is 200, change it to 55
- if the blue value is 20, change it to 235
- if the green value is 150, change it to 105
Media Computation: Fun with Images!
Media Computation:
Fun with Images!
How can we apply different operations to different pixels?
How can we apply different operations to different pixels?
For-Each Loop

- Loops through all of the elements in a sequence
- Syntax
  ```java
  for (type name : sequence)
  ```
- Example
  ```java
  for (Pixel pixelObj : pixelSequence)
  ```
- Each time through the loop this will be a different pixel from the collection of pixels until all of the pixels have been processed
Create a new method called `multicolored()`

How can we apply different operations to different pixels?
For Loop

- Has three areas: initialization, condition, and update
- Syntax
  ```
  for (initialization; condition; update)
  ```
- Example
  ```
  for (int x = 0; x < pic.getWidth(); x++)
  ```
For Loop Sequence of Actions

```plaintext
for (initialization; condition; update)
```

Example
```
for (int x = 0; x < pic.getWidth(); x++)
```

1. Execute **initialization** once
2. Each time through the loop, check the **condition**
3. If **condition** is true execute both the **body** of the loop and then the **update** step
4. Go back to 2. and repeat until **condition** is false
Processing pixels using (x, y) coordinates

```java
for (int x = 0; x < this.getWidth(); x++) {
    for (int y = 0; y < this.getHeight(); y++) {
        // process pixel at (x, y)
    }
}
```
Multicolored!

```java
for (int x = 0; x < this.getWidth(); x++)
{
    for (int y = 0; y < this.getHeight(); y++)
    {
        // ??
    }
}
```

- How do we know when to setGreen() vs. setBlue() vs. setRed()?
public void flipHorizontal()
{
    for (int x = 0; x < this.getWidth(); x++)
    {
        for (int y = 0; y < this.getHeight(); y++)
        {
            this.getPixel(x, y).setColorFrom(this.getPixel(x, y));
        }
    }
}
Flip Vertical

- Make a new `flipVertical()` method
- Use nested loops to create a vertical flip of the image
- Use `flipHorizontal()` as a model
Tiles