Media Computation:
More Image Fun!
How Would You Create This?
How is it done?

- What kinds of transforms?
- What image size?
- How did they all end up in the same image?
What About This?
Activities

- Dream up your own collage
- Sketch it on paper
- Create your own method and write your own version of a collage!
- Use your own images!
A quick recap

- Learning the basics:
  - Classes
  - Objects
  - Methods
  - Fields and variables
  - If statements, loops
Allow us to define what data is stored in an object

- Fields

Allow us to define new behaviors for objects

- Methods

Examples: Penny, Cloud, Cliff, Picture, Turtle, Lettuce, Actor, World
Objects

- The entities in the computer’s memory while the program is executing
- Objects **hold the data** our program is working on
- Objects **provide the actions** our program can perform
- Each object is defined by a class that determines its features
Methods

- A series of individual steps we combine into one action
- Examples: act(), changeBrightness(), maxGreen()
- Can use parameters to provide data to a method to customize its behavior
- Example: changeBrightness(-0.05)
Allow us to introduce names for our data

Variables allow us to give names to values used inside a method:

- int x = 0;
- Pixel source = in.getPixel(x, y);

Fields allow us to give names to values held inside an object:
- private int count = 100;
If statements let us choose actions

```java
if (x < this.getWidth() / 3) {
    // do something in the first third
}
else if (x < this.getWidth() * 2 / 3) {
    // do something in the second third
}
else {
    // Do something in the last third
}
```
For-each loops let us repeat over a collection

```java
for (Pixel p : this.getPixels())
{
    // do something with p
}
```
Numeric for loops let us repeat anything

```java
for (int x = 0; x < this.getWidth(); x++)
{
    for (int y = 0; y < this.getHeight(); y++)
    {
        // do something with (x, y)
    }
}
```
Joining Horizontally
Joining Vertically
Scaling

- `this.scale(1.20);`
- `this.scale(0.75);`

Or:

- `this.scale(32, 32);`
Chromakey (or “green screen”)
Basic chromakey recipe ...

- Two images
- Use some pixels from image 1, and some from image 2
- If image 1’s pixel is not blue, use it
- If image 1’s pixel is blue, use image 2’s pixel instead
Alpha Blending
Also called *compositing*
So a pixel color is ...

- A triple: (R, G, B)
- A red value (0-255)
- A green value (0-255)
- A blue value (0-255)
In reality: (R, G, B, A)
Also an alpha value (0-255)
Represents transparency
0 => fully transparent
255 => fully opaque

But there’s more ...
Basic alpha blending recipe ...

- Two images
- Use **all pixels** from both images
- Combine the color of image 1’s pixel with the color of image 2’s pixel
Basic alpha blending recipe ...

- Combine the colors of the two pixels using a weighted average
- Suppose image 1 is 80% transparent
- Combine 20% of image 1’s color with 80% of image 2’s color
For two colors C1 and C2

- Let C1 be the **top** image, and C2 be the **bottom**
- Let $A_1 = \alpha$ of top image (C1)
  
  $A_1/255 = \text{percent } C_1 \text{ is solid (C1’s strength)}$

  $(255 - A_1)/255 = \text{percent } C_1 \text{ is transparent (C2’s strength)}$

- Blended color is:
  
  $$C_1 \ast A_1/255 + C_2 \ast (255 - A_1)/255$$
Activities

- Dream up your own collage
- Sketch it on paper
- Create your own method and write your own version of a collage!
- Use your own images!