

Intro

Period 2

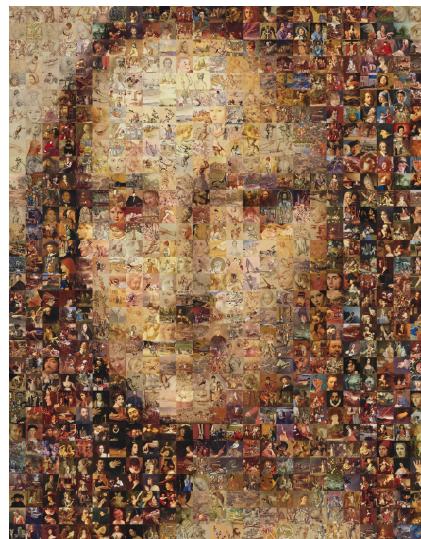
Felice Li

Felice Li

Photomosaics with Emojis

Description

My project aims to create photomosaics with emojis. Photomosaics are images made up of other tiny images. For example, in the image below, the overall image, the Mona Lisa, is made up of many tiny images of other works of art. My project would receive a file from the user, then process it to create a photomosaic. In my case, the tiny images would be emojis.



Functionalities:

There are two parts to this project. First, we must process emojis.

- Download a database of emoji photos and convert it to a list of emojis in the form of PImages
- Run through the emojis and find their average color by processing them pixel by pixel and averaging their RGB values
- Save their average color to a new list, where the index of the average color is also the index of the emoji in the emoji list

Then, we must process the overall image we want to recreate:

- User file input
- Divide the input image into regions

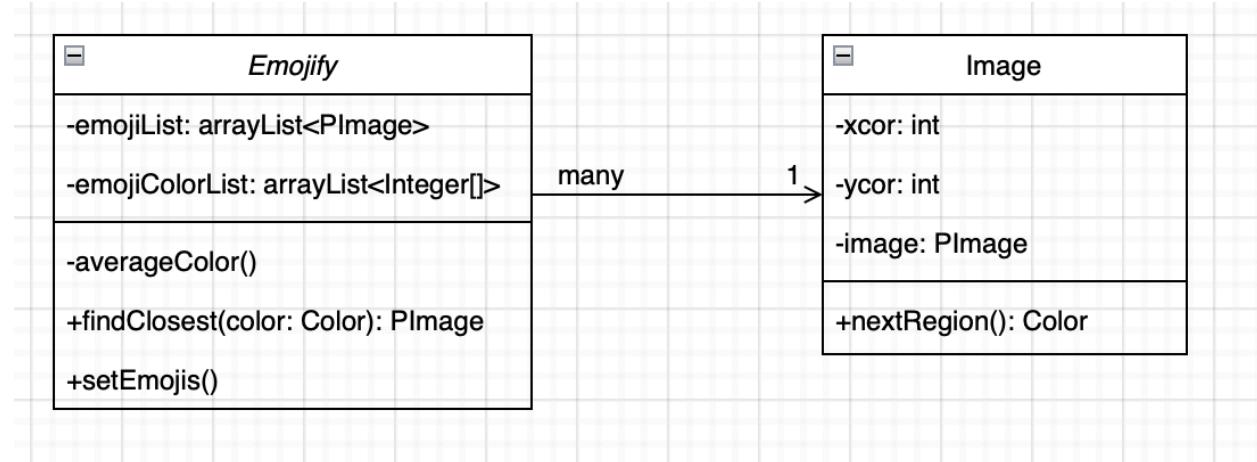
- Average the color of each region

Lastly, we put the two parts together:

- For each average color of a region, find the closest corresponding average color for an emoji.
Replace that region with the emoji.

Libraries: N/A

UML Diagram



How does it work?

Upon opening the program, the program will prompt the user to click the spacebar to start. Then, it will ask for a file input (there is a `selectInput()` method on Processing that does this). The program will run, processing the image and making a photomosaic from that image. The final output will be the original and rendered image side by side, with the original image on the left and the photomosaic on the right. Clicking space again will allow the user to upload another image to make into a photomosaic, and the above repeats.

Functionalities & Issues

Currently, I have the majority of the **Emojify** class done. A lot of it is processing images to first fit it into an `arrayList` (done in constructor), and then processing each image to find the average color and adding that to another `arrayList` (`averageColor()`). I also finished the `findClosest()` method, which is meant to find the color from the `arrayList` that is closest to the inputted color, utilizing the 3D distance formula. Lastly, I have my database of emojis! I had to decide whether to process it from a `.csv` file, which I realized it's tricky because I would have to extract images from a spreadsheet. Thus, I downloaded some images and

added them to a folder. For the Image class, I did start the nextRegion() method, which returns the averageColor of a region in the user-uploaded image. Lastly, I created a main class but ran into troubles (see last paragraph).

For the next meeting, I plan to finish the main class to piece everything together. Additionally, I plan to finish the setEmojis() method for Emojify.

I found a couple of issues. First, I realized that I couldn't make an arrayList where each element is a color, as a color is a primitive value. So, I changed color to an array of Integers, where it would house 3 integers, one to represent r, another to represent g, and the last one to represent b. Additionally, I realized I didn't know how to access a folder of files. Thus, I researched for a bit and found that .listFiles() returns an array of files! Additionally, I realized that edge cases for the nextRegion() method posed issues, so I created a isDone() method to counter that. Currently, I still have some unresolved issues. It seems like I cannot refer to another class's public methods for some reason. Also, I'm getting "The class 'Image' cannot have the same name as your sketch or its enclosing class" for Image's class declaration and I do not understand why. This is the main obstacle preventing me from starting the main() class.