Documentation for Final Project - Justin Lin:

My project was to create a webcam to use “images” to display what it was seeing by approximating the color it saw with a color palette it had.

The project was difficult because originally, I thought that I would be able to enable webcam and use pixel processing in order to do what I want. The problem is that pixel processing on a live image is extremely slow because the amount of information that is constantly being fed in and then having to be manipulated is too much.

Thus, I was forced to go into a webGL route, which is basically a more efficient way to process graphics. What I kind-of wanted to play with was the idea of visualization and the way we have been representing things over time, especially digitally. I think especially being able to see yourself in some other medium, such as on the 4th floor of 370 Jay St. MetroTech, there is a kinematic art piece that flips tiles based on the people in front of it is a powerful and fun experience and I wanted to play with that.

I decided, to go down a route of first representing a person with 1s and 0s the most basic representation in computers. Then from there I worked my way up to a RGB color palette. I added in a small pop-culture reference afterwards, before finally hitting the modern age, where it can feel like everyone has been using abbreviated messages and emojis in order to try to capture a greater complexity of feelings. Good or bad, is unknown, but it definitely is working out way into our culture.

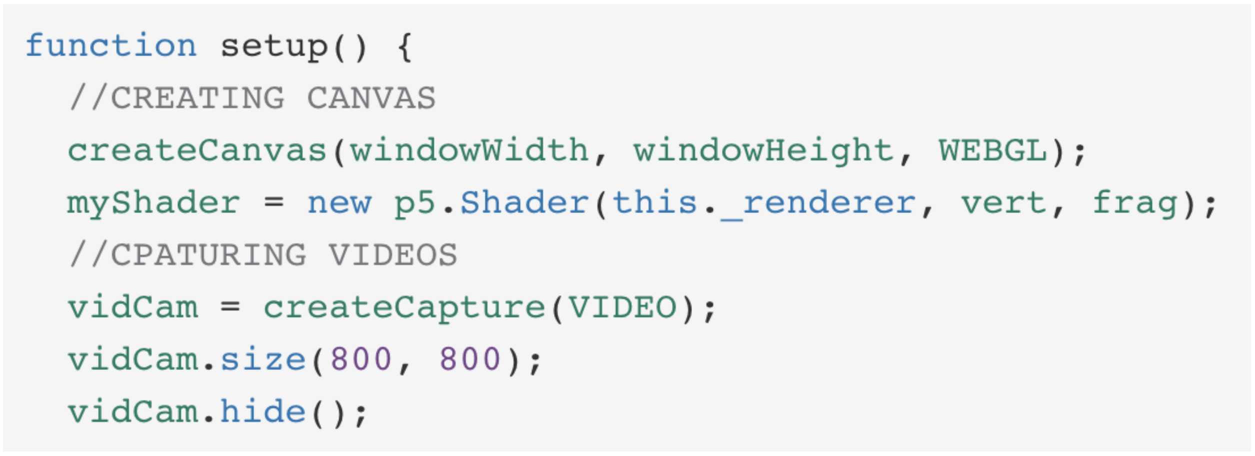
Definitely the largest part of this project, was exploring into frag and vert files for webGL. But they work pretty similar in that you write functions in the file, it basically seems to bound it into a variable or object, and manipulates those files as parameter inputs.

But overall, with enough research and messing around, with hitting bugs and errors, I was able to figure out it out and get it working.

P.S: Tried to get a video of the sketch, but screen-capping it made the sketch run unbelievably slow and impossible to capture or interact with

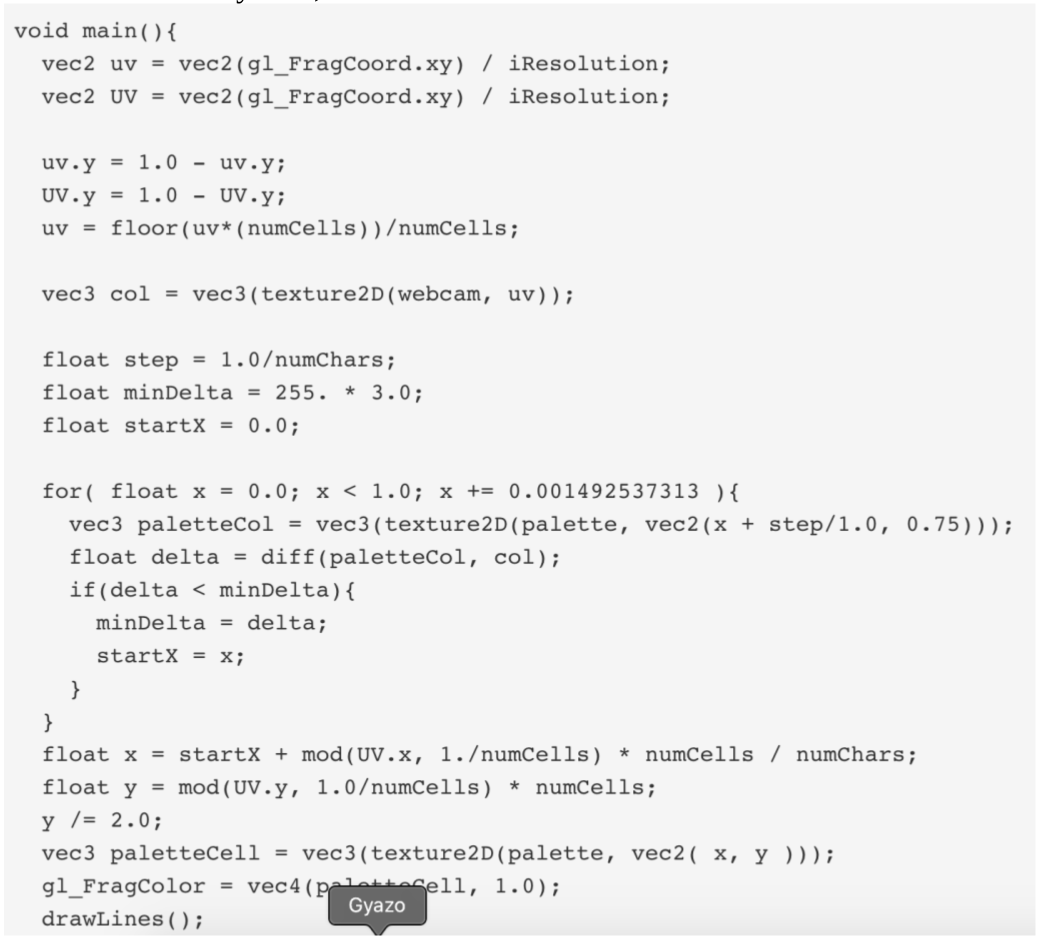
Screen-Shots

Code Proud of b/c of condensing and figuring out how to enable webcam capture, and passing the proper files to the setup. Couldn't seem to get the webcam to go more than the specified parameter of 800, 800, once I did, it seemed to bleed off into corrupted pixels, making me think that there is just a size limitation to the amount of pixels that the webcam sends to p5js directly.



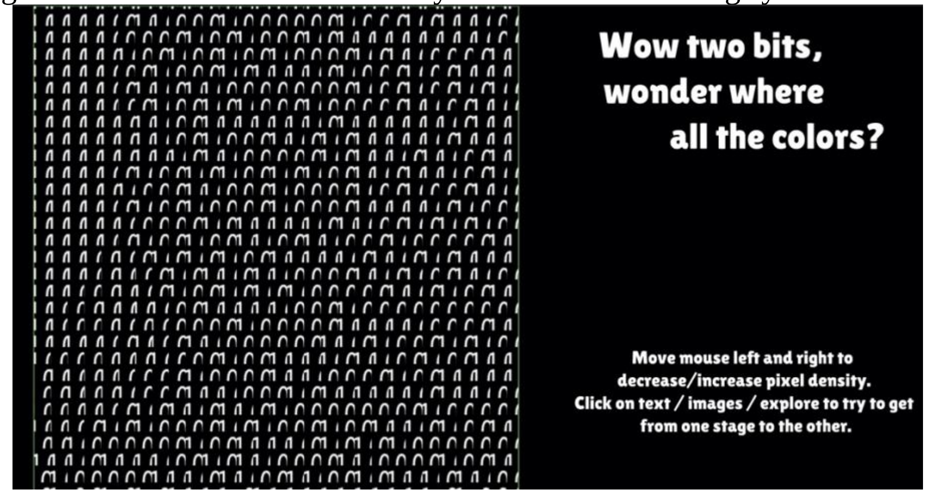
```
function setup() {
  //CREATING CANVAS
  createCanvas(windowWidth, windowHeight, WEBGL);
  myShader = new p5.Shader(this._renderer, vert, frag);
  //CPATURING VIDEOS
  vidCam = createCapture(VIDEO);
  vidCam.size(800, 800);
  vidCam.hide();
```

Writing Frag code that is able to scan through colors in images and insert color into color palette. The for-loop basically loops through every image, compares the difference of that color with what it currently sees, and then sets a cell based on the smallest difference.

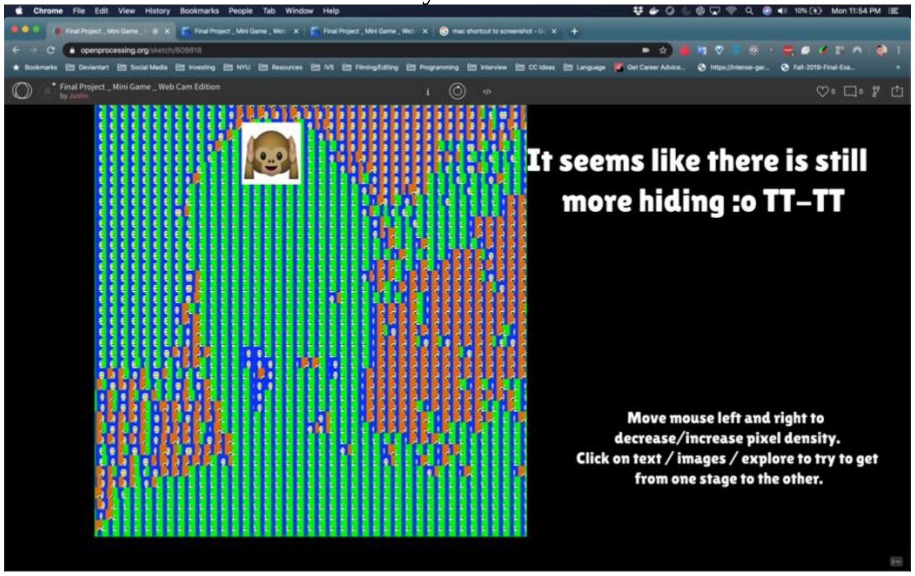


```
void main(){
  vec2 uv = vec2(gl_FragCoord.xy) / iResolution;
  vec2 UV = vec2(gl_FragCoord.xy) / iResolution;

  uv.y = 1.0 - uv.y;
  UV.y = 1.0 - UV.y;
  uv = floor(uv*(numCells))/numCells;

  vec3 col = vec3(texture2D(webcam, uv));

  float step = 1.0/numChars;
  float minDelta = 255. * 3.0;
  float startX = 0.0;

  for( float x = 0.0; x < 1.0; x += 0.001492537313 ){
    vec3 paletteCol = vec3(texture2D(palette, vec2(x + step/1.0, 0.75)));
    float delta = diff(paletteCol, col);
    if(delta < minDelta){
      minDelta = delta;
      startX = x;
    }
  }
  float x = startX + mod(UV.x, 1./numCells) * numCells / numChars;
  float y = mod(UV.y, 1.0/numCells) * numCells;
  y /= 2.0;
  vec3 paletteCell = vec3(texture2D(palette, vec2( x, y )));
  gl_FragColor = vec4(paletteCell, 1.0);
  drawLines();
```

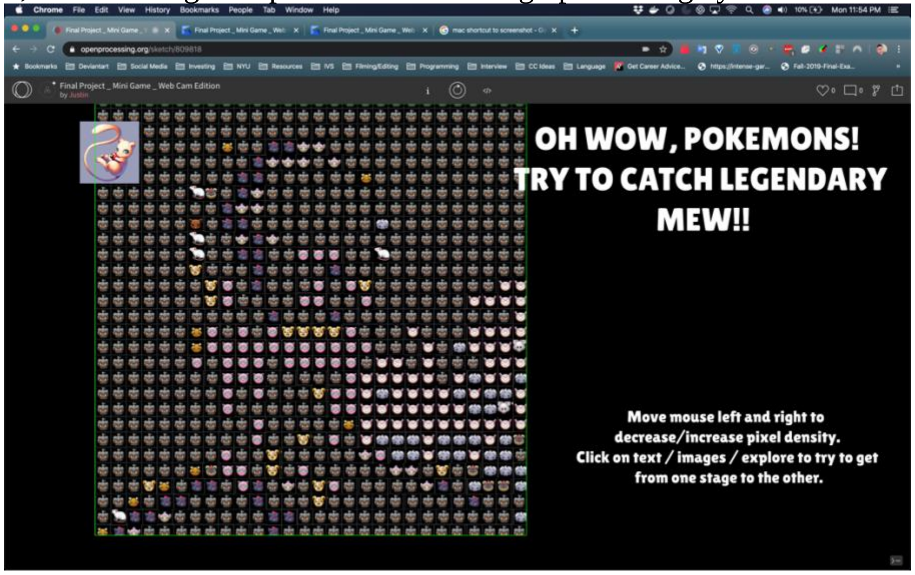
Beginning Screen of a 2-Bit World. When you click on color brings you to the next screen.


Wow two bits,
wonder where
all the colors?
Move mouse left and right to
decrease/increase pixel density.
Click on text / images / explore to try to get
from one stage to the other.

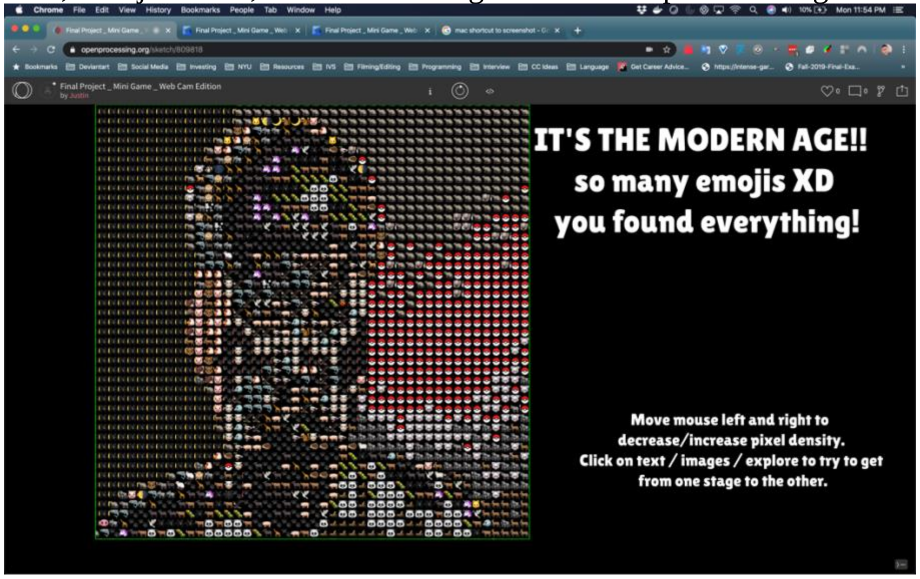
Secondary Screen with Image to Click on Revealed. If you hover on that part of the screen the monkey will show.


Final Project _ Mini Game _ Web Cam Edition
It seems like there is still
more hiding :o TT-TT
Move mouse left and right to
decrease/increase pixel density.
Click on text / images / explore to try to get
from one stage to the other.

Third Screen (Pop Culture Reference) with Hidden Image Revealed. If you move your mouse to the left, it will enlarge the pixels and also bring up the image you need to click on.


Final Project _ Mini Game _ Web Cam Edition
OH WOW, POKEMONS!
TRY TO CATCH LEGENDARY
MEW!!
Move mouse left and right to
decrease/increase pixel density.
Click on text / images / explore to try to get
from one stage to the other.

Last Screen, Emoji Screen, or the modern age where we represent things with emojis.


Final Project _ Mini Game _ Web Cam Edition
IT'S THE MODERN AGE!!
so many emojis XD
you found everything!
Move mouse left and right to
decrease/increase pixel density.
Click on text / images / explore to try to get
from one stage to the other.