CIM540 Midterm Project Write-Up
Apryl House

For this project I will focus on creating an interactive and playful introduction to my portfolio. As a creative advertising major, an essential part to acquiring and internship and career is having an engaging portfolio. Through the use of frame animation, arrays, counters, millis and a few other techniques, I built a simple yet visually stimulating greeting. Because my code is relatively open for expansion, I can use this project as a foundation that once my coding skills and knowledge have advanced I am able to build off to create a multifaceted greeting for not only my portfolio but website.

I first began my project by deciding what I wanted to accomplish and how I could accomplish it with the knowledge I have. My objective was to create a greeting that appeared on the screen but not all at once. I knew I could achieve this effect using objects on set counters or even text on set counters however each of these methods would not be visually what I desired and as such a time consuming process for a result I did not want. So in order to achieve the look I wanted I knew I had to use images. This revelation led me into the world of frame animation and millis. Through previous codes I had used for homework assignments I was able to develop a preliminary code that ran a set of images through a millis and frame relationship that drew from a preloaded image code and variables (pictured below).

```
var frameAmounts = 9;
var frameArray = [];
var currentFrame = 0;
var interval = 1000;
var pMillis = 0;
var framenumber = 2;

function preload() {
    for (var frames = 0; frames < frameAmounts; frames++) {
        var frameString = "assests2/welcome" + frames + ".png";
        frameArray[frames] = loadImage(frameString);
    }
} //close for preload
```

```
console.log(millis());
image(frameArray[currentFrame], 0, 0); // this must be above everything else in order to loop
if (millis() - pMillis >= interval) {
    currentFrame++;
    pMillis = millis();
}
if (currentFrame == frameArray.length) {
    currentFrame = 0; //this allows it to loop


} //close of draw
```

After achieving a successful code with temporary placement images, I then moved on to creating the actual images I wanted to use. I did this through the use of Adobe Illustrator. Although it is simple to create the images you desire and export them in a stream to an “assets” folder, it took me numerous times to get the correct image size. Ultimately I created eight different image sizes in Illustrator, each time having to export and rename each image in the stream before I finally realize that if I zoom my browser out to the greatest setting, my image showed up completely filling the frame. This was also a

challenging process because I was simultaneously trying to figure out the proper canvas size to reveal my full image. After much configuration I was able to settle on good dimensions.
With my code running properly with the correct images, I wanted a more ascetically pleasing background. This created a new objective and new ideation of how to achieve it. While I could have simply added a background to my images, I decided to use an array on a set counter to flicker the background from color to color (pictured below).

```
var colorArray = ["magenta", "orange", "yellow", "cyan"];
var counter = 0;
```

```
function draw() {
    background(colorArray[counter]);
    counter = counter + 1;
    if (counter > colorArray.length - 1) {
        counter = 0;
    }
```

Now I had achieved a fully functioning code and wished to test myself by adding more embellishments but instead ran into many challenges and failures. The first challenge I encountered was setting the background with the array and the frame animation on the millis on different frame rates. I tried various frame rates and its placement in my code, even using a variable for my frame rate but none of it had an effect of the speed of my array or animation. I also wished to add a twinkling star effect using objects and the bubble code we used in class but that made my code extremely complex and I ultimately could not get it to work. The last challenge I had which was my greatest failure due to the fact that I had the proper code, I just couldn't manage to get it to show up. I had attempted to add a second animation to follow the first. For this I added an interaction that would load and new image set into the millis setting on the press of the mouse (pictured below).

```
function mousePressed() {
    for (var frames = 0; frames < frameAmounts; frames++) {
        var frameString = "assests/thankyou" + frames + ".png";
        frameArray[frames] = loadImage(frameString);
    } //close to mousePressed
```

When I ran this code in Live Preview it did not show up and when I checked Developer Tools for errors none showed. I had a separate file that I was using to brainstorm and test code in before adding it to my official file. When I ran this file in live preview my code worked seamlessly and the new image set showed but never when I tried it in my actual file. It took hours of configuring and going back and forth between files for me to realize that the function was not working because of the array. When the array was commented out, the function would run a new image set but not with the array on. I played around with placement, substituting variables for values and different image formats but nothing I tried allowed the array and new image set to work concurrently. So in the end I abandoned this feat.

Overall I am satisfied with my project and that I was able to use the knowledge and skills I had learned in class to create a piece of code from scratch. I have many more ideas to add on to this code and hope to incorporate this work into my final project.