HW 3 - Journal

Concept

This project lets you create an ASCII character version of any picture you want as long as you give the width, height, and the picture. There are two ASCII character pictures. The first one is yellow and the second one is blue. At first both pictures are applied on the original picture, but when the viewer moves their mouse to the right, the pictures will unveil themselves one after another. And if you look closely, some characters are changing between their adjacent characters to give this fuzzy look. Additionally, the user can move their mouse and apply the effect however they want.

Generative Rules

Select each pixel and get their brightness value. Since the value is between 0 to 255, each 10 brightness value starting from 0 is used as an index for a character, with the last 5 brightness values (250 to 255) being an exception and getting assigned to the darkest character. Then, a random value between -5 and 5 gets selected to switch between adjacent characters. If the index is out of bounds after the random value is added, then we will use the brightness value without the randomness.

Moreover, if the mouse moves to the right, the pictures will follow the mouse until 2/3 of the window.

Challenges

As I applied the random value, I had to check for the value since it could be out of bounds for the characters array.

Source Code

```
PImage img;
PFont font;
color pixColor;

// Characters representing the different brightness levels. From 0 to 255.
char[] characters= {
    ' ', '`', '.', ',', ':', '-', '~', '=', '+', '*',
    'o', 'O', 'G', 'R', 'Q', 'H', 'X', 'W', 'M', 'B',
```

```
    '8', '&', '%', '@', '#', 'W'
};

//change the path of the picture but remember to change the window size from
setup too.
String path = "Mona_Lisa.jpg";
int photoW = 725; //photo width
int photoH = 1080;  //photo height

void setup(){
  size(2175,1080); // Image size got to have the width times three since we
have 3 pictures
  img = loadImage(path); // Image size 725x1080
  // Have to use a Mono font
  font = createFont("HackNerdFontMono-Regular.ttf", 12);
  textSize(4);
}

void draw(){
  background(0);
  image(img, 0,0);
  for (int y=0; y<photoH; y+=2){
    for (int x=0; x<photoW; x+=2){
       pixColor = get(x, y);     // selecting each pixel
       int b = round(brightness(pixColor));   // getting the brightness
value
       int rand = (int) random(-5, 5);

       // only selecting some pixels to randomize due to index being out of
bounds
       // pixels without the random glitching effect
       if (rand + b/10 > 25 || rand + b/10 < 0){
         fill(247, 223, 5);
         if (mouseX < 725){
           text(characters[b/10], mouseX + x, y);
         } else {
           text(characters[b/10], x + photoW, y);
         }
         fill(50, 88, 168);
         if (mouseX < 1450){
```

```
            text(characters[b/10], mouseX + x, y);
          } else {
            text(characters[b/10], x + 1450, y);
          }
        }
        // pixels with the random glitching effect
        else{
          fill(247, 223, 5);
          if (mouseX < 725){
            text(characters[rand + b/10], mouseX + x, y);
          } else {
            text(characters[rand + b/10], x + photoW, y);
          }
          fill(50, 88, 168);
          if (mouseX < 1450){
            text(characters[rand + b/10], mouseX + x, y);
          } else {
            text(characters[rand + b/10], x + 1450, y);
          }
        }
      }
    }
  }
```

Outcome