CIM 540/640

Section 4F/4H

Assignment Wk 14

Zhengrong Hu

Progress Report #2:

Project Name: IN & OUT

Location: Miami River

Description: Public Swimming pool lighting installation which trying to evoke community attention to water quality and sea level rise.

Checklist:

- ✓ Simulation for LED Matrix

  Bridging Arduino data to web

Introduction:

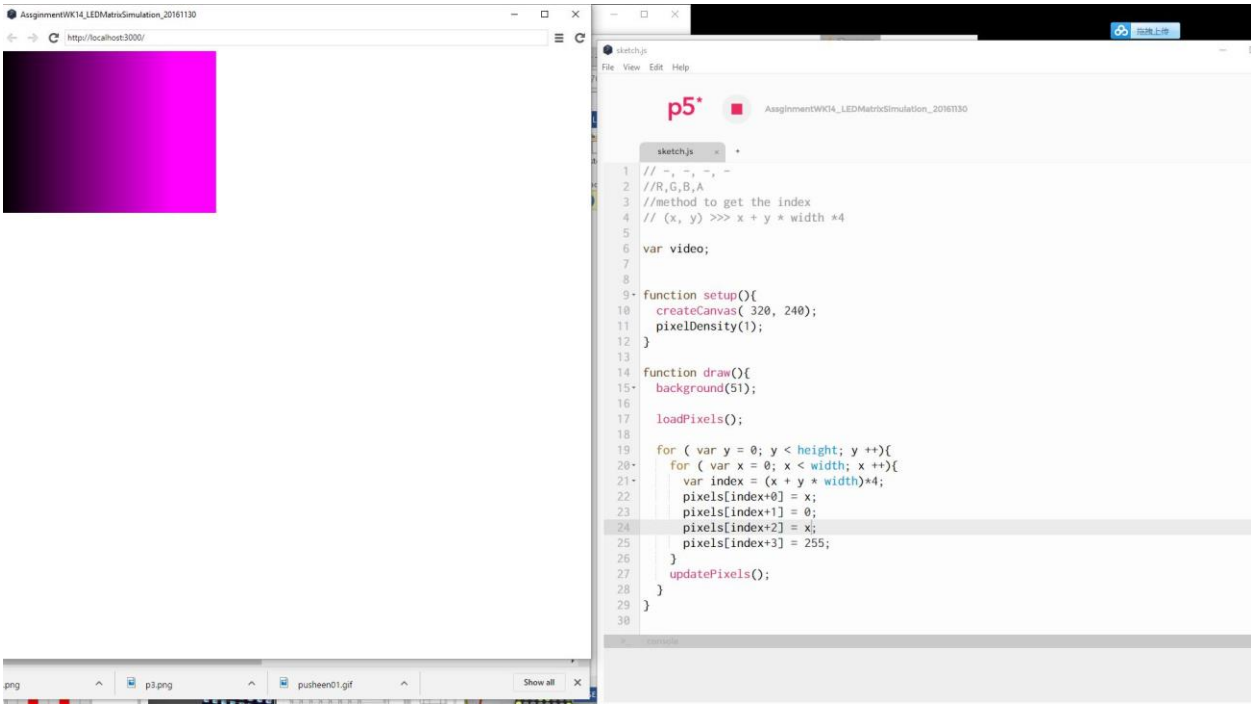
This project is meant to use web application as a data visualization tool to address people's attention about our environment, especially those information that rarely being captured by humans. In this case, I am proposing building a lighting installation at the Lummus Park near the Miami River. The installation has three components:

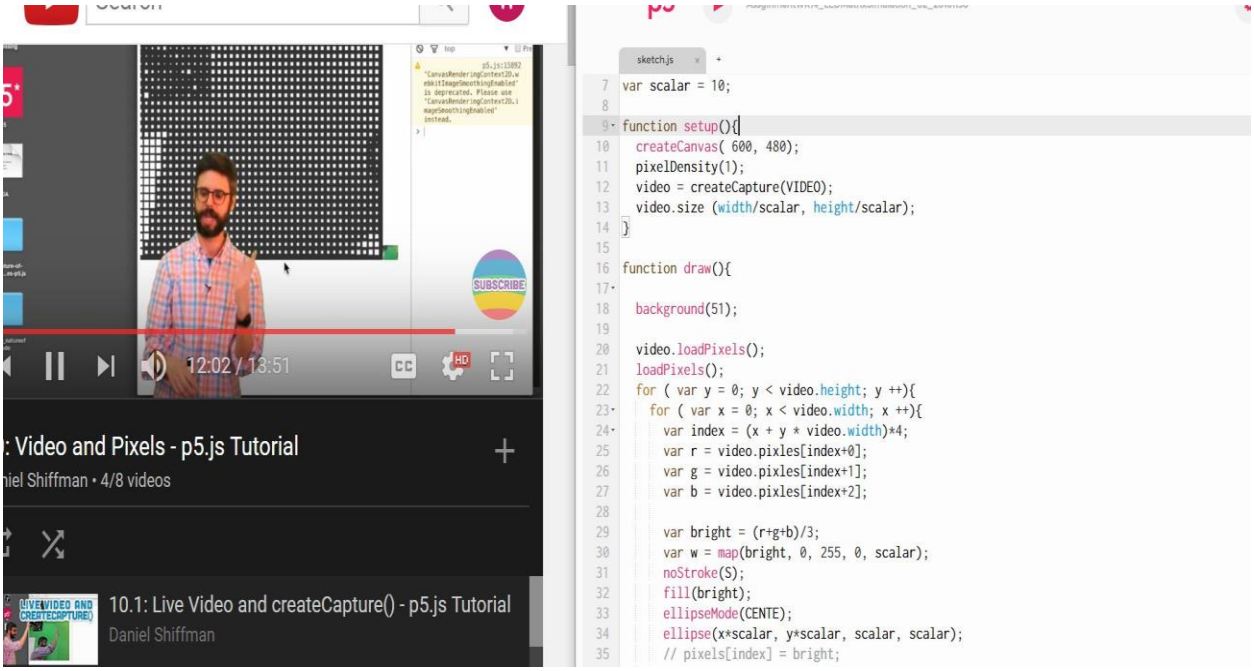
1: fish sensor (inside the Miami River)

2: LED Matrix Installation (near the swimming pool)

3: Website to bridge the data input and data output (internet)

The fish sensors will collecting real-time data about the fish movement inside the Miami River and sending those data to website. The website will receive, store and sending those parsed data to LED Matrix Installation and display the visual interpretation of those data.


AssginmentWK14_LEDMatrixSimulation_20161130
http://localhost:3000/
p3.png
pusheen01.gif
Show all
sketch.js
File View Edit Help
p5*
AssginmentWK14_LEDMatrixSimulation_20161130
sketch.js
// -, -, -, -
//R,G,B,A
//method to get the index
// (x, y) >>> x + y * width *4
var video;
function setup(){
createCanvas(320, 240);
pixelDensity(1);
}
function draw(){
background(51);
loadPixels();
for (var y = 0; y < height; y ++){
for (var x = 0; x < width; x ++){
var index = (x + y * width)*4;
pixels[index+0] = x;
pixels[index+1] = 0;
pixels[index+2] = x;
pixels[index+3] = 255;
}
updatePixels();
}
}


SUBSCRIBE
12:02 / 13:51
: Video and Pixels - p5.js Tutorial
iel Shiffman • 4/8 videos
10.1: Live Video and createCapture() - p5.js Tutorial
Daniel Shiffman
sketch.js
var scalar = 10;
function setup(){
createCanvas(600, 480);
pixelDensity(1);
video = createCapture(VIDEO);
video.size (width/scalar, height/scalar);
}
function draw(){
background(51);
video.loadPixels();
loadPixels();
for (var y = 0; y < video.height; y ++){
for (var x = 0; x < video.width; x ++){
var index = (x + y * video.width)*4;
var r = video.pixles[index+0];
var g = video.pixles[index+1];
var b = video.pixles[index+2];
var bright = (r+g+b)/3;
var w = map(bright, 0, 255, 0, scalar);
noStroke(S);
fill(bright);
ellipseMode(CENTE);
ellipse(x*scalar, y*scalar, scalar, scalar);
// pixels[index] = bright;

LED Matrix Simulation Code Plan:

1: Preload Video

2: Create Pixels Arrays

Divide canvas to Grid (R,G, B, A)

>>> nested for loop

>>>Assign RGB and A Values

>>> draw circle at the center of the pixels

>>>Map the brightness of the display to the brightness of the video

3: Display LED according to the Video