SASTA OLIPHANT SCIENCE AWARDS

SOUTH AUSTRALIAN SCIENCE TEACHERS ASSOCIATION

Highly Commended

Programming, Apps & Robotics

Year 5-6

Charles Tang

East Marden Primary School

Intelligent Bin

Isn't it annoying when you need to sort out your rubbish every single day? Well, that's where 'Intelligent Bin' comes in. Intelligent bin is a smart bin that uses artificial intelligence and machine learning to segregate different kinds of rubbish. Rubbish that isn't being sorted properly is an increasingly large issue that must be stopped. If all your rubbish is dumped into landfill, then as it decomposes it will release methane, which, coincidentally, is the most harmful greenhouse gas and a major cause of global warming and climate change. By accurately sorting out rubbish instead of dumping it all together it not only takes up less space, but it also makes the world a more sustainable place and reduces our ecological footprint, as well as being better for public health and the environment.

Intelligent Bin is a quick and reliable solution to fix this problem. By sorting and separating rubbish like plastics, metals, and papers, the rubbish will take up less space, pollution will be substantially reduced, and reusable materials won't be wasted. If rubbish is dumped into landfill, then will eventually be dumped into the ocean, becoming a part of a floating garbage patch it the ocean. Most of the rubbish in garbage patches could've been recycled to save money, but it would now take hundreds of million dollars to remove and is not a pleasant habitat for animals.

The goal of my project is to reduce the amount of rubbish carelessly tossed into landfill by sorting it beforehand, ultimately reducing the amount of greenhouse gases released into the atmosphere. Greenhouse gases, when released into the atmosphere, damage the ozone layer, thus allowing more UV rays to pass, which are harmful to animals and humans and lead to skin cancer. My project is intended to both efficiently and accurately sort out the rubbish it receives. Ideally, it would be used at both public places like parks, food courts and around the city and more industrial places like garbage depots or even on the garbage trucks themselves. In the future I would also like to make it as energy efficient as possible in order to further reduce its carbon footprint

My project uses AI and machine learning to classify a variety of rubbish. It has been trained utilizing 'Teachable Machine', a program designed by Google to quickly and easily train an accurate Tensorflow model on a set of images captured. When the model has finished training it is exported to the cloud where a link is generated for it that can be accessed on a webpage that I designed. I then programmed an Arduino Leonardo to connect to a p5.js sketch that I wrote and move a servo motor attached to deposit the rubbish when it recognizes something.

To run my project, you must first upload the Arduino Code to the Arduino Leonardo. Then you train the 'Teachable Machine' model before entering the link onto the p5.js webpage. When the webpage is running you can also connect the Arduino Leonardo to it via the UsbWeb library. Finally, give the webpage webcam access and put objects in the sorter for the AI to classify and segregate.

Code and Explanation

This Arduino code tells a servo motor to move the rubbish either right or left depending on the serial signal that it receives from the Artificial Intelligence

```
#include <WebUSB.h>
#include <Servo.h>

//TODO: fix this url hinting
WebUSB WebUSBSerial(1 /* https:// */, "webusb.github.io/arduino/demos/rgb");

#define Serial WebUSBSerial
Servo myservo;

const int redPin = 9;
const int greenPin = 10;
const int bluePin = 11;
int pos = 0;    // variable to store the servo position

int color[3];
int colorIndex;

void setup() {
  while (!Serial) {
    ;
  }
  Serial.begin(9600);
  Serial.write("Sketch begins.\r\n");
  Serial.flush();
  colorIndex = 0;
  myservo.attach(9);
  myservo.write(110);
}

void loop() {

  if (Serial && Serial.available()) {
    color[colorIndex++] = Serial.read();
    if (colorIndex == 1) {

      //       Serial.flush();
      //      analogWrite(redPin, color[0]);
      if (color[0] == 1) {
        myservo.write(30);
        delay(2000);
        for (pos = 0; pos <= 110; pos += 1) { // goes from 0 degrees to 180 degrees
          // in steps of 1 degree
          myservo.write(pos);              // tell servo to go to position in variable 'pos'
          delay(5);                       // waits 15ms for the servo to reach the position
        }
        delay(1000);
        Serial.write("compostable landfill detected.\r\n");
      }

      else if (color[0] == 2) {
        myservo.write(180);
        delay(2000);
        for (pos = 180; pos <= 110; pos -= 1) { // goes from 0 degrees to 180 degrees
          // in steps of 1 degree
          myservo.write(pos);              // tell servo to go to position in variable 'pos'
          delay(20);                       // waits 15ms for the servo to reach the position
        }
        delay(1000);
        Serial.write("mixed recycling detected.\r\n");
      }
      while (Serial.available()) {
        int throwaway = Serial.read();
        Serial.write("Throwing away.\r\n");
      }

      Serial.flush();
      colorIndex = 0;
    }
  }
  else {
    //    Serial.println("asdf");
    for (pos = 105; pos <= 115; pos += 1) { // goes from 0 degrees to 180 degrees
      // in steps of 1 degree
      myservo.write(pos);              // tell servo to go to position in variable 'pos'
      delay(3);                       // waits 15ms for the servo to reach the position
    }
    for (pos = 115; pos >= 105; pos -= 1) { // goes from 180 degrees to 0 degrees
      myservo.write(pos);              // tell servo to go to position in variable 'pos'
      delay(3);                       // waits 15ms for the servo to reach the position
    }
//    delay(200);
  }


}
```

Here I import the necessary libraries. WebUSB allows the Arduino Leonardo to connect to the webpage. Then I tell the Arduino what pin the servo motor is on

This is where I begin the serial monitor and tell it to print either '1' or '2' depending on what kind or rubbish it detects.

If the Arduino receives a serial signal of '1' then the servo motor will move to deposit the rubbish into compost, whereas if it receives '2' then it will sort the waste into mixed recycling

Here I state that if nothing is detected then the servo motor will slightly move back and forth to prompt the user to throw some rubbish into there

This p5.js sketch is written on the p5.js JavaScript IDE and sets up and creates the webpage that Intelligent Bin is used on.

```
class PhotoGrid {
  constructor(isLeft) {
    this.images = [];
    console.log(this.images.length);
    if (isLeft) {
      this.x = width / 2 - 480;
    } else {
      this.x = width / 2 + 300;
    }

    this.y = height / 2.5;
    this.numRows = 3;
    this.numCols = 2;
    this.imageSize = 120;
    this.padding = 20;
  }

  addImage(img) {
    this.images.push(img);
    if (this.images.length > 9) {
      this.images.shift();

    }
  }

  render() {

    for (let i = 0; i < this.images.length; i++) {

      let currImage = this.images[i];
      let row = i % 3;
      let col = int(i / 3);
      fill(255);
      noStroke();
      rectMode(CORNER);
      rect(this.x + (this.imageSize + this.padding) * col, this.y + (this.imageSize + this.padding) * row,
        this.imageSize, this.imageSize, 3, 3, 3, 3);

      image(currImage, this.x + (this.imageSize + this.padding) * col + 5, this.y + (this.imageSize + this.padding) * row + 5,
        this.imageSize - 10, this.imageSize - 10);
    }
  }


}


class DropDown {
  constructor(isLeft, classList) {
    this.isClicked = false;
    this.width =
      this.height = 3;
    this.x = 0;
    this.y = 0;

    this.class = null;
  }

  render() {
```

The class 'PhotoGrid' is where the webcam will go and its proportions

The 'render()' function here tells the browser to show the output from the webcam.

Here I create the borders for the webcam

The class 'DropDown' tells the browser to ask for permission to access the webcam. Once accepted it will start streaming video from the webcam.

```
60.
61.   }
62.
63.
64. }
65.
66. function debounce(func, wait, immediate) {
67.   var timeout;
68.   return function() {
69.     var context = this,
70.       args = arguments;
71.     var later = function() {
72.       timeout = null;
73.       if (!immediate) func.apply(context, args);
74.     };
75.     var callNow = immediate && !timeout;
76.     clearTimeout(timeout);
77.     timeout = setTimeout(later, wait);
78.     if (callNow) func.apply(context, args);
79.   };
80. }
81.
82.
83.
84. class Splash {
85.
86.   constructor(isLeft) {
87.     if (isLeft) {
88.       this.x = width / 2 + 314;
89.     } else {
90.       this.x = width / 2 - 314
91.     }
92.     this.y = height / 3.3;
93.     this.color = color(147, 229, 21);
94.     this.isExploding = false;
95.     this.isInbetweenUpdates = false;
96.     this.explosionRadius = 100;
97.     this.explosionIndex = 0;
98.     this.numRadius = 4;
99.     this.radiusOffset = 10;
100.          this.width = 243;
101.          this.height = 53;
102.        }
103.
104.        updatePosition(x, y) {
105.          this.x = x;
106.          this.y = y;
107.        }
108.
109.
110.        trigger() {
111.          this.isExploding = true;
112.        }
113.
114.        updateIndex() {
115.          this.explosionIndex++;
116.          this.isInbetweenUpdates = false;
117.        }
118.
119.        render() {
120.          if (!this.isExploding) {
121.            fill(this.color);
122.            // rect(this.x, this.y, this.width, this.height);
123.          } else {
124.            noFill();
125.            strokeWeight(3);
```

This function tells the webpage to turn the webcam on as soon as webcam access has been granted by the used

Lines 84 to 107 tells the browser what colour the text under class 'Splash' will be and where they are positioned.

Lines 110 to 144 basically states when and how often the 'Splash' text and the webcam will be updated.

```
126.             stroke(this.color);
127.             rect(this.x, this.y, this.width + (this.radiusOffset * this.explosionI
    ndex),
128.               this.height + (this.radiusOffset * this.explosionIndex), 9, 9, 9, 9)
    ;
129.           }
130.
131.           if (this.isExploding && !this.isInbetweenUpdates) {
132.             setTimeout(() => {
133.               this.updateIndex()
134.             }, 100);
135.             this.isInbetweenUpdates = true;  A
136.           }
137.
138.           if (this.explosionIndex >= this.numRadius) {
139.             this.isExploding = false;
140.             this.isInbetweenUpdates = false;
141.             this.explosionIndex = 0;
142.           }
143.
144.         }
145.
146.
147.
148.       }
149.       class ClassificationBar {
150.         constructor() {
151.           this.width = min(width / 4, 341);
152.           this.height = 28;
153.           this.x = width / 2;
154.           this.y = height / 3.3;
155.           this.radius = 5;
156.
157.           this.classificationLeft = 0;
158.           this.classificationMaxWidth = this.width / 2;
159.           this.classificationRight = 0.0;
160.           this.hasSetTimeout = false;
161.         }
162.
163.
164.         updateClassification(results) {
165.           // console.log(results);
166.           const class1 = results.filter(objs => {
167.             if (objs.label === labels[0]) {
168.               return objs;
169.             }
170.           });
171.
172.           const class2 = results.filter(objs => {
173.
174.             if (objs.label === labels[1]) {
175.               return objs;
176.             }
177.           });
178.
179.
180.           this.classificationLeft = map(class1[0].confidence, 0, 1.0, 0, this.clas
    sificationMaxWidth);
181.           this.classificationRight = map(class2[0].confidence, 0, 1.0, 0, this.cla
    ssificationMaxWidth);
182.
183.           let view = new Uint8Array(1);
184.
185.           if (class1[0].confidence > 0.90) {
186.             view[0] = 1;
187.             try {
```

Lines 149 to 155 tells the webpage how big the classification bar is and whereabouts it is located on the page.

Lines 164 to 177 update the classification bar constantly according to what is streamed from the webcam. The browser will send out either [0] or [1] according to what it detects and give a confidence rating of how confident it is of what it sees.

Here it is saying that if the confidence rating for either class1 or class2 is higher than 90% it will send out a signal of either 1 or 2 to the serial monitor that will be received by the Arduino Leonardo.

```
188.               port.send(view);
189.               shouldFreezeFrame = true;
190.               splashLeft.trigger();
191.
192.               isLeftPic = false;
193.             } catch (e) {}
194.           } else if (class2[0].confidence > 0.90) {
195.             view[0] = 2;
196.             try {
197.               port.send(view);
198.               shouldFreezeFrame = true;
199.               splashRight.trigger();
200.               isLeftPic = true;
201.             } catch (e) {}
202.
203.
204.           }
205.         }
206.
207.         render() {
208.           //Draw Background rectangle
209.           rectMode(CENTER);
210.           fill('rgba(174, 203, 250, 0.4)');
211.           stroke(255);
212.           strokeWeight(5);
213.           rect(this.x, this.y, this.width, this.height, this.radius, this.radius,
    this.radius, this.radius);
214.           noStroke();
215.
216.
217.           fill('#19ce1f');
218.           rect(this.x + this.classificationLeft / 2, this.y, this.classificationLe
    ft, this.height, this.radius, this.radius, this.radius, this.radius);
219.           rect(this.x - this.classificationRight / 2, this.y, this.classificationR
    ight, this.height, this.radius, this.radius, this.radius, this.radius);
220.           stroke(0);
221.           strokeWeight(7);
222.           strokeCap(ROUND);
223.           line(this.x, this.y - this.height / 2, this.x, this.y + this.height / 2)
    ;
224.
225.         }
226.       }
227.
228.       class ClassInput {
229.
230.         constructor(isLeft) {
231.           this.width = 243;
232.           this.height = 53;
233.           this.radius = 9;
234.           this.textLineOffset = 40;
235.           this.isLeft = isLeft;
236.           this.hoverOne = false;
237.           this.hoverTwo = false;
238.           this.hoverThree = true;
239.           if (isLeft === true) {
240.             this.x = width / 2 + 314;
241.           } else {
242.             this.x = width / 2 - 314;
243.           }
244.
245.           this.y = height / 3.3;
246.           this.isActive = false;
247.           this.currentValue = null;
248.
249.         }
```

Lines 207 to 226 tell the webpage what colour the background should be and draws it up. The hex colour and the RGB colour are the colour of the background

```
250.
251.         onClick(x, y) {
252.
253.           const leftBound = this.x - this.width / 2;
254.           const rightBound = this.x + this.width / 2;
255.           const bottomBound = this.y + this.height / 2;
256.           const topBound = this.y - this.height / 2;
257.           const isInside = (x >= leftBound && x <= rightBound && y <= bottomBound
    && y >= topBound);
258.
259.           if (isInside) {
260.             this.isActive = !this.isActive;
261.
262.           }
263.
264.         }
265.
266.         onHover(x, y) {
267.           this.detectZone(x, y);
268.
269.         }
270.         detectZone(x, y) {
271.           const leftBound = this.x - this.width / 2;
272.           const rightBound = this.x + this.width / 2;
273.
274.           const zoneOneBottom = this.y + this.height / 2;
275.           const zoneOneTop = this.y - this.height / 2;
276.
277.           if (x >= leftBound && x <= rightBound && y <= zoneOneTop && y >= zoneOne
    Bottom) {
278.             this.hoverOne = true;
279.             this.hoverTwo = false;
280.             this.hoverThree = false;
281.             return 1;
282.           }
283.         }
284.         render() {
285.           if (isModelLoaded) {
286.             fill(255);
287.             rectMode(CENTER);
288.             noStroke();
289.             textFont(poppinsBold);
290.             textSize(24);
291.             // if (!this.isActive) {
292.
293.               rect(this.x, this.y, this.width, this.height, this.radius, this.radi
    us, this.radius, this.radius);
294.
295.             // } else {
296.             //   rect(this.x, this.y + this.textLineOffset, this.width, this.heigh
    t + this.textLineOffset * 2, this.radius, this.radius, this.radius, this.radius);
297.             //   if (this.hoverOne) {
298.             //     fill('rgba(154,160,166, 0.2)');
299.             //     rect(this.x, this.y, this.width, this.height, this.radius, this
    .radius, 0, 0);
300.             //   } else if (this.hoverTwo) {
301.             //     fill('rgba(154, 160, 166, 0.2)');
302.             //     rect(this.x, this.y + this.textLineOffset, this.width, this.hei
    ght - 10, 0, 0, 0, 0);
303.             //   } else if (this.hoverThree) {
304.             //     fill('rgba(154, 160, 166, 0.2)');
305.             //     rect(this.x, this.y + this.textLineOffset * 2 + 6, this.width,
    this.height - 11, 0, 0, this.radius, this.radius);
306.             //   }
307.             // }
308.             if (labels.length >= 2) {
```

Here I check if the button has been clicked or not, and tells the webpage what the button will say, what colour it is and what font it is in.

```
309.               fill('#19ce1f');
310.               if (this.isLeft) {
311.                 textAlign(LEFT, CENTER);
312.                 text(labels[0], this.x - this.width / 2 + 10, this.y - 4);
313.       //           if (this.isActive) {
314.       //             text(labels[1], this.x - this.width / 2 + 10, this.y + this.t
    extLineOffset);
315.       //             text(labels[2], this.x - this.width / 2 + 10, this.y + this.t
    extLineOffset * 2);
316.
317.       //           }
318.                 image(pencil, this.x - this.width / 2 + 200, this.y - this.height
    / 2 + 10, pencil.width / 2, pencil.height / 2);
319.               } else {
320.                 textAlign(RIGHT, CENTER);
321.                 // if (labels.length > 2) {
322.                   text(labels[1], this.x + this.width / 2 - 13, this.y - 4);
323.                 // }
324.
325.       //           if (this.isActive) {
326.       //             text(labels[1], this.x + this.width / 2 - 13, this.y +AaW thi
    s.textLineOffset);
327.       //             text(labels[2], this.x + this.width / 2 - 13, this.y + this.t
    extLineOffset * 2);
328.       //           }
329.                 image(pencil, this.x + this.width / 2 - 235, this.y - this.height
    / 2 + 10, pencil.width / 2, pencil.height / 2);
330.               }
331.             }
332.
333.
334.           }
335.         }
336.       }
337.
338.       // Classifier Variable
339.       let classifier;
340.       let input;
341.       // Model URL
342.       let imageModel = 'https://teachablemachine.withgoogle.com/models/9L4-
    MDs0/';
343.
344.       // Video
345.       let video;
346.       let videoSize;
347.       let classificationIndicator;
348.
349.       let leftGrid;
350.       let leftAdd
351.       let rightGrid;
352.       let rightAdd;
353.
354.       let isLeftPic;
355.
356.       let leftClassSelector;
357.       let rightClassSelector;
358.
359.       let cameraBorder;
360.       let title;
361.       let splashLeft;
362.       let splashRight;
363.       let selectPic;
364.
365.       let editCode;
366.       let connect;
367.       let group;
```

This is the model URL to the trained model where the machine learning model will be hosted. It stores the weights and the .json files for the object detector.

Lines 345 to 388 define the let functions. The let functions declare all of the different variables. It starts off with declaring all the terms for getting the video running before declaring the background colours and fonts.

```
368.       let pencil;
369.
370.       // Darker BG
371.       // let bgColor = '#63e446';
372.       // Lighter Bg
373.       let bgColor = '#bce446';
374.       let port;
375.       let shouldFreezeFrame;
376.       let modeInput;
377.       let loadModel;
378.       let labels = [];
379.       let isLeftClassSelected = false;
380.       let isRightClassSelected = false;
381.
382.       let poppinsRegular;
383.       let poppinsBold;
384.       let hasSetPauseTimer;
385.       // To store the classification
386.       let label = "";
387.       let isModelLoaded = false;
388.       let enteredText = ""
389.       // // Load the model first
390.       // function preload() {
391.       //   classifier = ml5.imageClassifier(imageModel + 'model.json');
392.       // }
393.
394.       function myInputEvent() {
395.         enteredText = this.value();
396.       }
397.
398.       function setup() {
399.         createCanvas(window.innerWidth, window.innerHeight);
400.         // Create the video
401.         videoSize = 250;
402.         video = createCapture(VIDEO);
403.         video.hide();
404.
405.         cameraBorder = loadImage('camera_border.png');
406.         title = loadImage('title.png');
407.         group = loadImage('Group 61.png');
408.
409.         loadModel = new Clickable();
410.
411.         loadModel.resize(145, 40);
412.
413.         loadModel.locate(300, 15);
414.         loadModel.strokeWeight = 0;
415.         loadModel.color = bgColor;
416.         loadModel.text = 'LOAD MODEL';
417.         loadModel.textSize = 18;
418.         loadModel.textColor = '#19ce1f';
419.         loadModel.onPress = () => {
420.           try {
421.             console.log(enteredText + 'metadata.json');
422.             classifier = ml5.imageClassifier(enteredText + 'model.json');
423.
424.             httpGet(enteredText + 'metadata.json', 'json', false, (response) => {
425.               if (response.labels.length <= 2) {
426.                 alert("Train a model with at least three classes: one for each typ
    e of object you want to sort, and one for the empty sorter");
427.               } else {
428.                 labels = response.labels;
429.                 isModelLoaded = true;
430.                 classifyVideo();
431.               }
```

Here the webpage loads the model and starts up the video. The video dimensions are determined as well before some pictures are imported from the sketch files.

The Clickable() function works with the p5.clickable.min.js file to create the 'LOAD MODEL' button. It is dimensioned, sized and coloured before being told what it will do one clicked. Here it says that if there are 2 or less different classes it will come up with an alert. Otherwise, it will load the model and start

```
432.
433.             }, (error) => alert("invalid TM2 url"));
434.           } catch (e) {
435.             loadModel.text = 'INVALID URL';
436.           }
437.           if (labels.length > 1) {
438.             loadModel.text = 'MODEL LOADED';
439.             setTimeout(() => {
440.               loadModel.text = 'REFRESH MODEL'
441.             }, 3000);
442.           }
443.         }
444.
445.         leftGrid = new PhotoGrid(true);
446.         pencil = loadImage('pencil_icon.png');
447.         classificationIndicator = new ClassificationBar();
448.         leftClassSelector = new ClassInput(true);
449.         rightClassSelector = new ClassInput(false);
450.         splashRight = new Splash(false);
451.         splashLeft = new Splash(true);
452.         rightGrid = new PhotoGrid(false);
453.         poppinsRegular = loadFont('Poppins-Regular.ttf');
454.         poppinsBold = loadFont('Poppins-Bold.ttf');
455.
456.         loadModel.textFont = poppinsRegular;
457.         shouldFeezeFrame = false;
458.         hasSetPauseTimer = false;
459.         var serial = {};
460.
461.         modelInput = createInput();
462.         modelInput.input(myInputEvent);
463.         // modelInput.style('position', 'absolute');
464.         // modelInput.style('z-index', 10);
465.         modelInput.position(20, 20);
466.         modelInput.style('height', '35px');
467.         modelInput.style('width', '267px');
468.         modelInput.style('border-width', '0px');
469.         modelInput.style('border-radius', '4px 4px 0px 0px');
470.         modelInput.style('border-bottom', '2px solid #19ce1f');
471.         modelInput.style('font-family', 'Poppins');
472.         modelInput.style('font-size', '16px');
473.         modelInput.style('padding-left', '5px');
474.         modelInput.style('color', '#19ce1f');
475.         modelInput.attribute('placeholder', "Paste model link here");
476.
477.
478.         connect = createButton('CONNECT ARDUINO');
479.         connect.position(width - 200, 20);
480.         connect.id("connect");
481.         connect.style('height', '40px');
482.         connect.style('border-width', '0px');
483.         connect.style('background-color', bgColor);
484.         connect.style('font-family', 'Poppins');
485.         connect.style('font-size', '18px');
486.         connect.style('width', '200px');
487.         connect.style('color', '#19ce1f');
488.         leftAdd = debounce(() => {
489.           leftGrid.addImage(selectPic)
490.         }, 500, true);
491.         rightAdd = debounce(() => {
492.           rightGrid.addImage(selectPic)
493.         }, 500, true);
494.
495.         editCode = createA('https://editor.p5js.org/ctang21/sketches/ttnyj7hOV', '
    EDIT CODE', '_blank');
496.         editCode.position(width - 110, height - 40);
```

Here it will come up with an alert if the trained model link is incorrect if it doesn't work. If it is fine, it will say 'MODEL LOADED' and give the option to 'REFRESH MODEL' in case you updated it.

The 'pencil_icon.png' image is loaded onto the classification bar and the text, fonts and colours are also imported.

This section of the code creates the textbox where you can paste the trained model's link. After that a button is created to go in the top right corner of the page that says 'CONNECT ARDUINO'. It is also coloured and sized.

Below that a button that says 'EDIT CODE' is created. If clicked, it will take you to the p5.js sketch where you can edit the code.

```
497.          editCode.style('height', '40px');
498.          editCode.style('border-width', '0px');
499.          editCode.style('background-color', bgColor);
500.          editCode.style('font-family', 'Poppins');
501.          editCode.style('font-size', '18px');
502.          editCode.style('width', '200px');
503.          editCode.style('color', '#19ce1f');
504.
505.          // Start classifying
506.          if (isModelLoaded) {
507.            classifyVideo();
508.          }
509.        }
510.
511.        function draw() {
512.          //   Darker BG
513.          if (width > 700) {
514.            background(bgColor);
515.            video.get();
516.            //   Darker BG
517.            // background('#e8f0fe');
518.            if (shouldFreezeFrame && !hasSetPauseTimer) {
519.              video.pause();
520.              selectPic = video.get(150, 0, videoSize / 1.6, videoSize / 1.6);
521.              if (isLeftPic) {
522.                leftAdd();
523.              } else {
524.                rightAdd();
525.              }
526.              setTimeout(() => {
527.
528.                video.play();
529.                hasSetPauseTimer = false;
530.                shouldFreezeFrame = false
531.
532.              }, 2000);
533.            }
534.
535.            noStroke();
536.            textFont(poppinsBold);
537.              textAlign(CENTER, CENTER);
538.              textSize(14);
539.              text("enable webcam access", width / 2, height / 1.6);
540.              text("and refresh page to use", width / 2, height / 1.5);
541.            image(title, width / 2 - title.width / 5, 0, title.width / 2.5, title.he
    ight / 2.5);
542.            image(video, width / 2 - videoSize / 2, height / 1.6 - videoSize / 2, vi
    deoSize, videoSize, 150, 0, videoSize * 1.5, videoSize * 1.5);
543.            image(cameraBorder, width / 2 - videoSize / 2 - 3, height / 1.6 - videoS
    ize / 2 - 3, videoSize + 6, videoSize + 6);
544.
545.            // // image(connect, width - connect.width - 20, 20);
546.            // image(group, 20, 20);
547.
548.            // rectMode(CENTER);
549.            // noFill();
550.            // stroke(255);
551.            // strokeWeight(6);
552.            // rect(width / 2, height / 2, videoSize, videoSize);
553.            leftGrid.render();
554.            rightGrid.render();
555.            rectMode(CORNER);
556.            loadModel.draw();
557.            classificationIndicator.render();
558.            leftClassSelector.render();
559.            rightClassSelector.render();
```

This portion of the code tells the pictures taken from the webcam stream once they have been classified what should happen to them and where to go.

Here it says what will happen if webcam access isn't granted; the webcam will ask for control of the webpage before it tells you to refresh the page to see the webcam

```
560.          splashLeft.render();
561.          splashRight.render();
562.        } else {
563.          noStroke();
564.
565.          text("expand page or ", width / 2, height / 1.6);
566.          text("load on a computer to use", width / 2, height / 1.5);
567.        }
568.      }
569.
570.      // Get a prediction for the current video frame
571.      function classifyVideo() {
572.        classifier.classify(video, gotResult);
573.        // classifier.classify(video, () => {});
574.      }
575.
576.      // When we get a result
577.      function gotResult(error, results) {
578.        // If there is an error
579.        if (error) {
580.          console.error(error);
581.          return;
582.        }
583.        // The results are in an array ordered by confidence.
584.        // console.log(results[0]);
585.        classificationIndicator.updateClassification(results);
586.
587.        label = results[0].label;
588.        // Classifiy again!
589.        classifyVideo();
590.      }
591.
592.      function windowResized() {
593.        resizeCanvas(windowWidth, windowHeight);
594.        const leftPhotos = leftGrid.images;
595.        const rightPhotos = rightGrid.images;
596.        leftGrid = new PhotoGrid(true);
597.        rightGrid = new PhotoGrid(false);
598.        leftGrid.images = leftPhotos;
599.        rightGrid.images = rightPhotos;
600.        classificationIndicator = new ClassificationBar();
601.        leftClassSelector = new ClassInput(true);
602.        rightClassSelector = new ClassInput(false);
603.        splashRight = new Splash(false);
604.        splashLeft = new Splash(true);
605.        loadModel = new Clickable();
606.        connect.position(width - 200, 20);
607.        loadModel.resize(145, 40);
608.        loadModel.locate(300, 15);
609.        loadModel.strokeWeight = 0;
610.        loadModel.color = '#19ce1f';
611.        loadModel.text = 'LOAD MODEL';
612.        loadModel.textSize = 18;
613.        loadModel.textColor = '#19ce1f';
614.        loadModel.onPress = () => {
615.          loadModel.text = 'MODEL LOADED';
616.          setTimeout(() => {
617.            loadModel.text = 'REFRESH MODEL'
618.          }, 3000);
619.        }
620.        // connect.textFont = poppinsRegular;
621.        loadModel.textFont = poppinsRegular;
622.      }
623.
624.      function mousePressed() {
625.        leftClassSelector.onClick(mouseX, mouseY);
```

If the webpage is too small the camera will not be accessed, and so here it tells you to expand the page.

This is where the video is finally classified. It will make a prediction of what it is and give a percentage of certainty before telling you what it classified it as. Once you receive a result the webcam will start classifying again.

The windowResized() function tells the webpage how big everything should be if the browser is extended. This is so that the ratio of the size of the text/buttons/shapes to the size of the webpage stays the same. It also makes it so that the text/buttons/shapes are evenly distributed throughout the page. Here it updates the video, the classification bar and the buttons.

This part of the code creates the 'load model' button. It gives the browser the x and y coordinates of the button and tells it how big they should be. It also creates the space for you to paste the trained model's link. Then it will send out a signal once it has been clicked.

```
626.         rightClassSelector.onClick(mouseX, mouseY);
627.       }
628.
629.       function mouseMoved() {
630.         leftClassSelector.onHover(mouseX, mouseY);
631.         rightClassSelector.onHover(mouseX, mouseY);
632.       }
```

This next script is a html script that imports all of the libraries that my webpage needs to run.

```
1.  <!DOCTYPE html>
2.  <html>
3.
4.  <head>
5.    <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.9.0/p5.js"></script>
6.    <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.9.0/addons/p5.dom.min.js"></script>
7.    <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.9.0/addons/p5.sound.min.js"></script>
8.    <script src="https://unpkg.com/ml5@0.4.1/dist/ml5.min.js"></script>
9.    <link href="https://fonts.googleapis.com/css?family=Poppins&display=swap" rel="stylesheet">
10.
11.   <link rel="stylesheet" type="text/css" href="style.css">
12.   <link rel="stylesheet" type="text/css" href="purejscarousel.css">
13.   <meta charset="utf-8" />
14.
15.   <script src="CanvasInput.min.js"></script>
16.   <script src="p5.clickable.js"></script>
17. </head>
18.
19. <body>
20.
21.   <script src="sketch.js"></script>
22.   <script src="serial.js"></script>
23.
24. </body>
25.
26. </html>
```

The first library allows p5.js to get access to the webcam. The next one lets p5.js classify images and then the third one gives the webpage servo motor control. The next two libraries import the font that I used and then 'CanvasInput.min.js' and 'p5.clickable.js' are pre-written scripts that I made to give my webpage html5 access and the clickable buttons.

Sketch.js is the script before that tells the browser what the webpage looks like, and serial.js gives the webpage serial access to talk to the Arduino.

This JavaScript sketch allows me to use buttons in my webpage

```
1.  //Determines if the mouse was pressed on the previous frame
2.  var cl_mouseWasPressed = false;
3.  //Last hovered button
4.  var cl_lastHovered = null;
5.  //Last pressed button
6.  var cl_lastClicked = null;
7.  //All created buttons
8.  var cl_clickables = [];
```

I didn't write the code for this script; it is a JavaScript library created by the developers of p5.js to add buttons to the webpage.

```
9.
10. //This function is what makes the magic happen and should be ran after
11. //each draw cycle.
12. p5.prototype.runGUI = function(){
13.     for(i = 0; i < cl_clickables.length; ++i){
14.         if(cl_lastHovered != cl_clickables[i])
15.             cl_clickables[i].onOutside();
16.     }
17.     if(cl_lastHovered != null){
18.         if(cl_lastClicked != cl_lastHovered){
19.             cl_lastHovered.onHover();
20.         }
21.     }
22.     if(!cl_mouseWasPressed && cl_lastClicked != null){
23.         cl_lastClicked.onPress();
24.     }
25.     if(cl_mouseWasPressed && !mouseIsPressed && cl_lastClicked != null){
26.         if(cl_lastClicked == cl_lastHovered){
27.             cl_lastClicked.onRelease();
28.         }
29.         cl_lastClicked = null;
30.     }
31.     cl_lastHovered = null;
32.     cl_mouseWasPressed = mouseIsPressed;
33. }
34.
35. p5.prototype.registerMethod('post', p5.prototype.runGUI);
36.
37. //Button Class
38. function Clickable(){
39.     this.x = 0;         //X position of the clickable
40.     this.y = 0;         //Y position of the clickable
41.     this.width = 100;       //Width of the clickable
42.     this.height = 50;       //Height of the clickable
43.     this.color = "#FFFFFF";     //Background color of the clickable
44.     this.cornerRadius = 10;     //Corner radius of the clickable
45.     this.strokeWeight = 2;      //Stroke width of the clickable
46.     this.stroke = "#000000";    //Border color of the clickable
47.     this.text = "Press Me";     //Text of the clickable
48.     this.textColor = "#000000"; //Color for the text shown
49.     this.textSize = 12;     //Size for the text shown
50.     this.textFont = "sans-serif";   //Font for the text shown
51.
52.     this.onHover = function(){
53.         //This function is ran when the clickable is hovered but not
54.         //pressed.
55.     }
56.
57.     this.onOutside = function(){
58.         //This function is ran when the clickable is NOT hovered.
59.     }
60.
61.     this.onPress = function(){
62.         //This fucking is ran when the clickable is pressed.
63.     }
64.
65.     this.onRelease = function(){
66.         //This funcion is ran when the cursor was pressed and then
67.         //released inside the clickable. If it was pressed inside and
68.         //then released outside this won't work.
69.     }
70.
71.     this.locate = function(x, y){
72.         this.x = x;
73.         this.y = y;
74.     }
```

```
75.
76.     this.resize = function(w, h){
77.         this.width = w;
78.         this.height = h;
79.     }
80.
81.     this.draw = function(){
82.         fill(this.color);
83.         stroke(this.stroke);
84.         strokeWeight(this.strokeWeight);
85.         rect(this.x, this.y, this.width, this.height, this.cornerRadius);
86.         fill(this.textColor);
87.         noStroke();
88.         textAlign(CENTER, CENTER);
89.         textSize(this.textSize);
90.         textFont(this.textFont);
91.         text(this.text, this.x+1, this.y+1, this.width, this.height);
92.         if(mouseX >= this.x && mouseY >= this.y
93.            && mouseX < this.x+this.width && mouseY < this.y+this.height){
94.             cl_lastHovered = this;
95.             if(mouseIsPressed && !cl_mouseWasPressed)
96.                 cl_lastClicked = this;
97.         }
98.     }
99.
100.            cl_clickables.push(this);
101.       }
```

I also didn't write this script, but essentially it sets up the canvas for the webpage to run on. It gives p5.js HTML5 input, meaning that I can use commands like render(), canvas(), fontSize, fontFamily, etc.

```
1.  /*!
2.   *  CanvasInput v1.2.7
3.   *  http://goldfirestudios.com/blog/108/CanvasInput-HTML5-Canvas-Text-Input
4.   *
5.   *  (c) 2013-2017, James Simpson of GoldFire Studios
6.   *  goldfirestudios.com
7.   *
8.   *  MIT License
9.   */
10. !function(){var e=[],t=window.CanvasInput=function(t){var n=this;t=t?t:{},n._canvas
    =t.canvas||null,n._ctx=n._canvas?n._canvas.getContext("2d"):null,n._x=t.x||0,n._y=t
    .y||0,n._extraX=t.extraX||0,n._extraY=t.extraY||0,n._fontSize=t.fontSize||14,n._fon
    tFamily=t.fontFamily||"Arial",n._fontColor=t.fontColor||"#000",n._placeHolderColor=
    t.placeHolderColor||"#bfbebd",n._fontWeight=t.fontWeight||"normal",n._fontStyle=t.f
    ontStyle||"normal",n._fontShadowColor=t.fontShadowColor||"",n._fontShadowBlur=t.fon
    tShadowBlur||0,n._fontShadowOffsetX=t.fontShadowOffsetX||0,n._fontShadowOffsetY=t.f
    ontShadowOffsetY||0,n._readonly=t.readonly||!1,n._maxlength=t.maxlength||null,n._wi
    dth=t.width||150,n._height=t.height||n._fontSize,n._padding=t.padding>=0?t.padding:
    5,n._borderWidth=t.borderWidth>=0?t.borderWidth:1,n._borderColor=t.borderColor||"#9
    59595",n._borderRadius=t.borderRadius>=0?t.borderRadius:3,n._backgroundImage=t.back
    groundImage||"",n._boxShadow=t.boxShadow||"1px 1px 0px rgba(255, 255, 255, 1)",n._i
    nnerShadow=t.innerShadow||"0px 0px 4px rgba(0, 0, 0, 0.4)",n._selectionColor=t.sele
    ctionColor||"rgba(179, 212, 253, 0.8)",n._placeHolder=t.placeHolder||"",n._value=(t
    .value||n._placeHolder)+"",n._onsubmit=t.onsubmit||function(){},n._onkeydown=t.onke
    ydown||function(){},n._onkeyup=t.onkeyup||function(){},n._onfocus=t.onfocus||functi
    on(){},n._onblur=t.onblur||function(){},n._cursor=!1,n._cursorPos=0,n._hasFocus=!1,
    n._selection=[0,0],n._wasOver=!1,n.boxShadow(n._boxShadow,!0),n._calcWH(),n._render
    Canvas=document.createElement("canvas"),n._renderCanvas.setAttribute("width",n.oute
    rW),n._renderCanvas.setAttribute("height",n.outerH),n._renderCtx=n._renderCanvas.ge
    tContext("2d"),n._shadowCanvas=document.createElement("canvas"),n._shadowCanvas.set
```

```
Attribute("width",n._width+2*n._padding),n._shadowCanvas.setAttribute("height",n._h
eight+2*n._padding),n._shadowCtx=n._shadowCanvas.getContext("2d"),"undefined"!=type
of t.backgroundGradient?(n._backgroundColor=n._renderCtx.createLinearGradient(0,0,0
,n.outerH),n._backgroundColor.addColorStop(0,t.backgroundGradient[0]),n._background
Color.addColorStop(1,t.backgroundGradient[1])):n._backgroundColor=t.backgroundColor
||"#fff",n._canvas&&(n._canvas.addEventListener("mousemove",function(e){e=e||window
.event,n.mousemove(e,n)},!1),n._canvas.addEventListener("mousedown",function(e){e=e
||window.event,n.mousedown(e,n)},!1),n._canvas.addEventListener("mouseup",function(
e){e=e||window.event,n.mouseup(e,n)},!1));var o=function(e){e=e||window.event,n._ha
sFocus&&!n._mouseDown&&n.blur()};window.addEventListener("mouseup",o,!0),window.add
EventListener("touchend",o,!0),n._hiddenInput=document.createElement("input"),n._hi
ddenInput.type="text",n._hiddenInput.style.position="absolute",n._hiddenInput.style
.opacity=0,n._hiddenInput.style.pointerEvents="none",n._hiddenInput.style.zIndex=0,
n._hiddenInput.style.transform="scale(0)",n._updateHiddenInput(),n._maxlength&&(n._
hiddenInput.maxLength=n._maxlength),document.body.appendChild(n._hiddenInput),n._hi
ddenInput.value=n._value,n._hiddenInput.addEventListener("keydown",function(e){e=e|
|window.event,n._hasFocus&&(window.focus(),n._hiddenInput.focus(),n.keydown(e,n))})
,n._hiddenInput.addEventListener("keyup",function(e){e=e||window.event,n._value=n._
hiddenInput.value,n._cursorPos=n._hiddenInput.selectionStart,n._selection=[n._hidde
nInput.selectionStart,n._hiddenInput.selectionEnd],n.render(),n._hasFocus&&n._onkey
up(e,n)}),e.push(n),n._inputsIndex=e.length-
1,n.render()};t.prototype={canvas:function(e){var t=this;return"undefined"!=typeof
e?(t._canvas=e,t._ctx=t._canvas.getContext("2d"),t.render()):t._canvas},x:function(
e){var t=this;return"undefined"!=typeof e?(t._x=e,t._updateHiddenInput(),t.render()
):t._x},y:function(e){var t=this;return"undefined"!=typeof e?(t._y=e,t._updateHidde
nInput(),t.render()):t._y},extraX:function(e){var t=this;return"undefined"!=typeof
e?(t._extraX=e,t._updateHiddenInput(),t.render()):t._extraX},extraY:function(e){var
 t=this;return"undefined"!=typeof e?(t._extraY=e,t._updateHiddenInput(),t.render())
:t._extraY},fontSize:function(e){var t=this;return"undefined"!=typeof e?(t._fontSiz
e=e,t.render()):t._fontSize},fontFamily:function(e){var t=this;return"undefined"!=t
ypeof e?(t._fontFamily=e,t.render()):t._fontFamily},fontColor:function(e){var t=thi
s;return"undefined"!=typeof e?(t._fontColor=e,t.render()):t._fontColor},placeHolder
Color:function(e){var t=this;return"undefined"!=typeof e?(t._placeHolderColor=e,t.r
ender()):t._placeHolderColor},fontWeight:function(e){var t=this;return"undefined"!=
typeof e?(t._fontWeight=e,t.render()):t._fontWeight},fontStyle:function(e){var t=th
is;return"undefined"!=typeof e?(t._fontStyle=e,t.render()):t._fontStyle},fontShadow
Color:function(e){var t=this;return"undefined"!=typeof e?(t._fontShadowColor=e,t.re
nder()):t._fontShadowColor},fontShadowBlur:function(e){var t=this;return"undefined"
!=typeof e?(t._fontShadowBlur=e,t.render()):t._fontShadowBlur},fontShadowOffsetX:fu
nction(e){var t=this;return"undefined"!=typeof e?(t._fontShadowOffsetX=e,t.render()
):t._fontShadowOffsetX},fontShadowOffsetY:function(e){var t=this;return"undefined"!
=typeof e?(t._fontShadowOffsetY=e,t.render()):t._fontShadowOffsetY},width:function(
e){var t=this;return"undefined"!=typeof e?(t._width=e,t._calcWH(),t._updateCanvasWH
(),t._updateHiddenInput(),t.render()):t._width},height:function(e){var t=this;retur
n"undefined"!=typeof e?(t._height=e,t._calcWH(),t._updateCanvasWH(),t._updateHidden
Input(),t.render()):t._height},padding:function(e){var t=this;return"undefined"!=ty
peof e?(t._padding=e,t._calcWH(),t._updateCanvasWH(),t.render()):t._padding},border
Width:function(e){var t=this;return"undefined"!=typeof e?(t._borderWidth=e,t._calcW
H(),t._updateCanvasWH(),t.render()):t._borderWidth},borderColor:function(e){var t=t
his;return"undefined"!=typeof e?(t._borderColor=e,t.render()):t._borderColor},borde
rRadius:function(e){var t=this;return"undefined"!=typeof e?(t._borderRadius=e,t.ren
der()):t._borderRadius},backgroundColor:function(e){var t=this;return"undefined"!=t
ypeof e?(t._backgroundColor=e,t.render()):t._backgroundColor},backgroundGradient:fu
nction(e){var t=this;return"undefined"!=typeof e?(t._backgroundColor=t._renderCtx.c
reateLinearGradient(0,0,0,t.outerH),t._backgroundColor.addColorStop(0,e[0]),t._back
groundColor.addColorStop(1,e[1]),t.render()):t._backgroundColor},boxShadow:function
(e,t){var n=this;if("undefined"==typeof e)return n._boxShadow;var o=e.split("px ");
return n._boxShadow={x:"none"===n._boxShadow?0:parseInt(o[0],10),y:"none"===n._boxS
hadow?0:parseInt(o[1],10),blur:"none"===n._boxShadow?0:parseInt(o[2],10),color:"non
e"===n._boxShadow?"":o[3]},n._boxShadow.x<0?(n.shadowL=Math.abs(n._boxShadow.x)+n._
boxShadow.blur,n.shadowR=n._boxShadow.blur+n._boxShadow.x):(n.shadowL=Math.abs(n._b
oxShadow.blur-
n._boxShadow.x),n.shadowR=n._boxShadow.blur+n._boxShadow.x),n._boxShadow.y<0?(n.sha
dowT=Math.abs(n._boxShadow.y)+n._boxShadow.blur,n.shadowB=n._boxShadow.blur+n._boxS
hadow.y):(n.shadowT=Math.abs(n._boxShadow.blur-
n._boxShadow.y),n.shadowB=n._boxShadow.blur+n._boxShadow.y),n.shadowW=n.shadowL+n.s
```

```
hadowR,n.shadowH=n.shadowT+n.shadowB,n._calcWH(),t?void 0:(n._updateCanvasWH(),n.re
nder())},innerShadow:function(e){var t=this;return"undefined"!=typeof e?(t._innerSh
adow=e,t.render()):t._innerShadow},selectionColor:function(e){var t=this;return"und
efined"!=typeof e?(t._selectionColor=e,t.render()):t._selectionColor},placeHolder:f
unction(e){var t=this;return"undefined"!=typeof e?(t._placeHolder=e,t.render()):t._
placeHolder},value:function(e){var t=this;return"undefined"!=typeof e?(t._value=e+"
",t._hiddenInput.value=e+"",t._cursorPos=t._clipText().length,t.render(),t):t._valu
e===t._placeHolder?"":t._value},onsubmit:function(e){var t=this;return"undefined"!=
typeof e?(t._onsubmit=e,t):void t._onsubmit()},onkeydown:function(e){var t=this;ret
urn"undefined"!=typeof e?(t._onkeydown=e,t):void t._onkeydown()},onkeyup:function(e
){var t=this;return"undefined"!=typeof e?(t._onkeyup=e,t):void t._onkeyup()},focus:
function(t){var n=this;if(!n._hasFocus){n._onfocus(n);for(var o=0;o<e.length;o++)e[
o]._hasFocus&&e[o].blur()}n._selectionUpdated?delete n._selectionUpdated:n._selecti
on=[0,0],n._hasFocus=!0,n._readonly?n._hiddenInput.readOnly=!0:(n._hiddenInput.read
Only=!1,n._cursorPos="number"==typeof t?t:n._clipText().length,n._placeHolder===n._
value&&(n._value="",n._hiddenInput.value=""),n._cursor=!0,n._cursorInterval&&clearI
nterval(n._cursorInterval),n._cursorInterval=setInterval(function(){n._cursor=!n._c
ursor,n.render()},500));var r=n._selection[0]>0||n._selection[1]>0;return n._hidden
Input.focus(),n._hiddenInput.selectionStart=r?n._selection[0]:n._cursorPos,n._hidde
nInput.selectionEnd=r?n._selection[1]:n._cursorPos,n.render()},blur:function(e){var
 t=e||this;return t._onblur(t),t._cursorInterval&&clearInterval(t._cursorInterval),
t._hasFocus=!1,t._cursor=!1,t._selection=[0,0],t._hiddenInput.blur(),""===t._value&
&(t._value=t._placeHolder),t.render()},keydown:function(t,n){var o=t.which;t.shiftK
ey;if(!n._readonly&&n._hasFocus){if(n._onkeydown(t,n),65===o&&(t.ctrlKey||t.metaKey
))return n.selectText(),t.preventDefault(),n.render();if(17==o||t.metaKey||t.ctrlK
ey)return n;if(13===o)t.preventDefault(),n._onsubmit(t,n);else if(9===o&&(t.prevent
Default(),e.length>1)){var r=e[n._inputsIndex+1]?n._inputsIndex+1:0;n.blur(),setTim
eout(function(){e[r].focus()},10)}return n._value=n._hiddenInput.value,n._cursorPos
=n._hiddenInput.selectionStart,n._selection=[0,0],n.render()}},click:function(e,t){
var n=t._mousePos(e),o=n.x,r=n.y;return t._endSelection?(delete t._endSelection,voi
d delete t._selectionUpdated):t._canvas&&t._overInput(o,r)||!t._canvas?t._mouseDown
?(t._mouseDown=!1,t.click(e,t),t.focus(t._clickPos(o,r))):void 0:t.blur()},mousemov
e:function(e,t){var n=t._mousePos(e),o=n.x,r=n.y,d=t._overInput(o,r);if(d&&t._canva
s?(t._canvas.style.cursor="text",t._wasOver=!0):t._wasOver&&t._canvas&&(t._canvas.s
tyle.cursor="default",t._wasOver=!1),t._hasFocus&&t._selectionStart>=0){var a=t._cl
ickPos(o,r),i=Math.min(t._selectionStart,a),_=Math.max(t._selectionStart,a);if(!d)r
eturn t._selectionUpdated=!0,t._endSelection=!0,delete t._selectionStart,void t.ren
der();t._selection[0]===i&&t._selection[1]===_||(t._selection=[i,_],t.render())}},m
ousedown:function(e,t){var n=t._mousePos(e),o=n.x,r=n.y,d=t._overInput(o,r);t._mous
eDown=d,t._hasFocus&&d&&(t._selectionStart=t._clickPos(o,r))},mouseup:function(e,t)
{var n=t._mousePos(e),o=n.x,r=n.y,d=t._clickPos(o,r)!==t._selectionStart;t._hasFocu
s&&t._selectionStart>=0&&t._overInput(o,r)&&d?(t._selectionUpdated=!0,delete t._sel
ectionStart,t.render()):delete t._selectionStart,t.click(e,t)},selectText:function(
e){var t=this,e=e||[0,t._value.length];return setTimeout(function(){t._selection=[e
[0],e[1]],t._hiddenInput.selectionStart=e[0],t._hiddenInput.selectionEnd=e[1],t.ren
der()},1),t},renderCanvas:function(){return this._renderCanvas},render:function(){v
ar e=this,t=e._renderCtx,n=e.outerW,o=e.outerH,r=e._borderRadius,d=e._borderWidth,a
=e.shadowW,i=e.shadowH;t&&(t.clearRect(0,0,t.canvas.width,t.canvas.height),t.shadow
OffsetX=e._boxShadow.x,t.shadowOffsetY=e._boxShadow.y,t.shadowBlur=e._boxShadow.blu
r,t.shadowColor=e._boxShadow.color,e._borderWidth>0&&(t.fillStyle=e._borderColor,e.
_roundedRect(t,e.shadowL,e.shadowT,n-a,o-
i,r),t.fill(),t.shadowOffsetX=0,t.shadowOffsetY=0,t.shadowBlur=0),e._drawTextBox(fu
nction(){t.shadowOffsetX=0,t.shadowOffsetY=0,t.shadowBlur=0;var _=e._clipText(),u=e
._padding+e._borderWidth+e.shadowT;if(e._selection[1]>0){var s=e._textWidth(_.subst
ring(0,e._selection[0])),l=e._textWidth(_.substring(e._selection[0],e._selection[1]
));t.fillStyle=e._selectionColor,t.fillRect(u+s,u,l,e._height)}if(e._cursor){var h=
e._textWidth(_.substring(0,e._cursorPos));t.fillStyle=e._fontColor,t.fillRect(u+h,u
,1,e._height)}var c=e._padding+e._borderWidth+e.shadowL,f=Math.round(u+e._height/2)
;_=""===_&&e._placeHolder?e._placeHolder:_,t.fillStyle=""!==e._value&&e._value!==e.
_placeHolder?e._fontColor:e._placeHolderColor,t.font=e._fontStyle+" "+e._fontWeight
+" "+e._fontSize+"px "+e._fontFamily,t.shadowColor=e._fontShadowColor,t.shadowBlur=
e._fontShadowBlur,t.shadowOffsetX=e._fontShadowOffsetX,t.shadowOffsetY=e._fontShado
wOffsetY,t.textAlign="left",t.textBaseline="middle",t.fillText(_,c,f);var p=e._inne
rShadow.split("px "),v="none"===e._innerShadow?0:parseInt(p[0],10),w="none"===e._in
nerShadow?0:parseInt(p[1],10),x="none"===e._innerShadow?0:parseInt(p[2],10),b="none
"===e._innerShadow?"":p[3];if(x>0){var g=e._shadowCtx,y=g.canvas.width,S=g.canvas.h
```

```
eight;g.clearRect(0,0,y,S),g.shadowBlur=x,g.shadowColor=b,g.shadowOffsetX=0,g.shado
wOffsetY=w,g.fillRect(-1*n,-
100,3*n,100),g.shadowOffsetX=v,g.shadowOffsetY=0,g.fillRect(y,-
1*o,100,3*o),g.shadowOffsetX=0,g.shadowOffsetY=w,g.fillRect(-
1*n,S,3*n,100),g.shadowOffsetX=v,g.shadowOffsetY=0,g.fillRect(-100,-
1*o,100,3*o),e._roundedRect(t,d+e.shadowL,d+e.shadowT,n-2*d-a,o-2*d-
i,r),t.clip(),t.drawImage(e._shadowCanvas,0,0,y,S,d+e.shadowL,d+e.shadowT,y,S)}retu
rn e._ctx&&(e._ctx.clearRect(e._x,e._y,t.canvas.width,t.canvas.height),e._ctx.drawI
mage(e._renderCanvas,e._x,e._y)),e}))},destroy:function(){var t=this,n=e.indexOf(t)
;-
1!=n&&e.splice(n,1),t._hasFocus&&t.blur(),document.body.removeChild(t._hiddenInput)
,t._renderCanvas=null,t._shadowCanvas=null,t._renderCtx=null},_drawTextBox:function
(e){var t=this,n=t._renderCtx,o=t.outerW,r=t.outerH,d=t._borderRadius,a=t._borderWi
dth,i=t.shadowW,_=t.shadowH;if(""===t._backgroundImage)n.fillStyle=t._backgroundCol
or,t._roundedRect(n,a+t.shadowL,a+t.shadowT,o-2*a-i,r-2*a-
_,d),n.fill(),e();else{var u=new Image;u.src=t._backgroundImage,u.onload=function()
{n.drawImage(u,0,0,u.width,u.height,a+t.shadowL,a+t.shadowT,o,r),e()}},_clearSelec
tion:function(){var e=this;if(e._selection[1]>0){var t=e._selection[0],n=e._selecti
on[1];return e._value=e._value.substr(0,t)+e._value.substr(n),e._cursorPos=t,e._cur
sorPos=e._cursorPos<0?0:e._cursorPos,e._selection=[0,0],!0}return!1},_clipText:func
tion(e){var t=this;e="undefined"==typeof e?t._value:e;var n=t._textWidth(e),o=n/(t.
_width-t._padding),r=o>1?e.substr(-
1*Math.floor(e.length/o)):e;return r+""},_textWidth:function(e){var t=this,n=t._ren
derCtx;return n.font=t._fontStyle+" "+t._fontWeight+" "+t._fontSize+"px "+t._fontFa
mily,n.textAlign="left",n.measureText(e).width},_calcWH:function(){var e=this;e.out
erW=e._width+2*e._padding+2*e._borderWidth+e.shadowW,e.outerH=e._height+2*e._paddin
g+2*e._borderWidth+e.shadowH},_updateCanvasWH:function(){var e=this,t=e._renderCanv
as.width,n=e._renderCanvas.height;e._renderCanvas.setAttribute("width",e.outerW),e.
_renderCanvas.setAttribute("height",e.outerH),e._shadowCanvas.setAttribute("width",
e._width+2*e._padding),e._shadowCanvas.setAttribute("height",e._height+2*e._padding
),e._ctx&&e._ctx.clearRect(e._x,e._y,t,n)},_updateHiddenInput:function(){var e=this
;e._hiddenInput.style.left=e._x+e._extraX+(e._canvas?e._canvas.offsetLeft:0)+"px",e
._hiddenInput.style.top=e._y+e._extraY+(e._canvas?e._canvas.offsetTop:0)+"px",e._hi
ddenInput.style.width=e._width+2*e._padding+"px",e._hiddenInput.style.height=e._hei
ght+2*e._padding+"px"},_roundedRect:function(e,t,n,o,r,d){2*d>o&&(d=o/2),2*d>r&&(d=
r/2),e.beginPath(),e.moveTo(t+d,n),e.lineTo(t+o-
d,n),e.quadraticCurveTo(t+o,n,t+o,n+d),e.lineTo(t+o,n+r-
d),e.quadraticCurveTo(t+o,n+r,t+o-
d,n+r),e.lineTo(t+d,n+r),e.quadraticCurveTo(t,n+r,t,n+r-
d),e.lineTo(t,n+d),e.quadraticCurveTo(t,n,t+d,n),e.closePath()},_overInput:function
(e,t){var n=this,o=e>=n._x+n._extraX,r=e<=n._x+n._extraX+n._width+2*n._padding,d=t>
=n._y+n._extraY,a=t<=n._y+n._extraY+n._height+2*n._padding;return o&&r&&d&&a},_clic
kPos:function(e){var t=this,n=t._value;t._value===t._placeHolder&&(n="");var o=t._c
lipText(n),r=0,d=o.length;if(e-
(t._x+t._extraX)<t._textWidth(o))for(var a=0;a<o.length;a++)if(r+=t._textWidth(o[a]
),r>=e-
(t._x+t._extraX)){d=a;break}return d},_mousePos:function(e){var t=e.target,n=e.page
X,o=e.pageY;e.touches&&e.touches.length?(t=e.touches[0].target,n=e.touches[0].pageX
,o=e.touches[0].pageY):e.changedTouches&&e.changedTouches.length&&(t=e.changedTouch
es[0].target,n=e.changedTouches[0].pageX,o=e.changedTouches[0].pageY);var r=documen
t.defaultView.getComputedStyle(t,void 0),d=parseInt(r.paddingLeft,10)||0,a=parseInt
(r.paddingLeft,10)||0,i=parseInt(r.borderLeftWidth,10)||0,_=parseInt(r.borderLeftWi
dth,10)||0,u=document.body.parentNode.offsetTop||0,s=document.body.parentNode.offse
tLeft||0,l=0,h=0;if("undefined"!=typeof t.offsetParent)do l+=t.offsetLeft,h+=t.offs
etTop;while(t=t.offsetParent);return l+=d+i+s,h+=a+_+u,{x:n-l,y:o-h}}}}();
```

This script gives the Arduino Leonardo a serial connection to the p5.js sketch

```
1.  (function() {
2.    'use strict';
3.
4.    document.addEventListener('DOMNodeInserted', event => {
```

```
5.      let connectButton = document.querySelector("#connect");
6.
7.
8.
9.      function connect() {
10.       port.connect().then(() => {
11.
12.         connectButton.textContent = 'DISCONNECT';
13.
14.         port.onReceive = data => {
15.           let textDecoder = new TextDecoder();
16.           console.log(textDecoder.decode(data));
17.         }
18.         port.onReceiveError = error => {
19.           console.error(error);
20.         };
21.       }, error => {
22.
23.       });
24.     }
25.     try {
26.     connectButton.addEventListener('click', function() {
27.       if (port) {
28.         port.disconnect();
29.         connectButton.textContent = 'CONNECT ARDUINO';
30.
31.         port = null;
32.       } else {
33.         serial.requestPort().then(selectedPort => {
34.           port = selectedPort;
35.           connect();
36.         }).catch(error => {
37.
38.         });
39.       }
40.     });
41.
42.     }
43.     catch (e) {
44.
45.       console.log("p5 sketch not loaded yet: ", e);
46.     }
47.
48. //     serial.getPorts().then(ports => {
49. //       if (ports.length == 0) {
50.
51. //       } else {
52.
53. //         port = ports[0];
54. //         connect();
55. //       }
56. //     });
57.   });
58. })();
59.
60.
61. // From https://github.com/webusb/arduino/blob/gh-pages/demos/serial.js
62. var serial = {};
63.
64. (function() {
65.   'use strict';
66.
67.   serial.getPorts = function() {
68.     return navigator.usb.getDevices().then(devices => {
69.       console.log(devices);
70.       return devices.map(device => new serial.Port(device));
```

The first function tries to pair the two by finding which port the Arduino is plugged into and requesting to connect.

Once the Arduino is connected the 'CONNECT ARDUINO' button will change to say 'DISCONNECT'. The catch(e) statement allows you to handle the error of 'p5 sketch not loaded yet' in the Google Chrome console if you aren't running the sketch in editing mode. This means that you won't need to have two tabs open at once while running the serial connection.

```
71.     });
72.   };
73.
74.   serial.requestPort = function() {
75.     const filters = [
76.       { 'vendorId': 0x2341, 'productId': 0x8036 },
77.       { 'vendorId': 0x2341, 'productId': 0x8037 },
78.       { 'vendorId': 0x2341, 'productId': 0x804d },
79.       { 'vendorId': 0x2341, 'productId': 0x804e },
80.       { 'vendorId': 0x2341, 'productId': 0x804f },
81.       { 'vendorId': 0x2341, 'productId': 0x8050 },
82.     ];
83.     return navigator.usb.requestDevice({ 'filters': filters }).then(
84.       device => new serial.Port(device)
85.     );
86.   }
87.
88.   serial.Port = function(device) {
89.     this.device_ = device;
90.   };
91.
92.   serial.Port.prototype.connect = function() {
93.     let readLoop = () => {
94.       this.device_.transferIn(5, 64).then(result => {
95.         this.onReceive(result.data);
96.         readLoop();
97.       }, error => {
98.         this.onReceiveError(error);
99.       });
100.          };
101.
102.          return this.device_.open()
103.              .then(() => {
104.                if (this.device_.configuration === null) {
105.                  return this.device_.selectConfiguration(1);
106.                }
107.              })
108.              .then(() => this.device_.claimInterface(2))
109.              .then(() => this.device_.selectAlternateInterface(2, 0))
110.              .then(() => this.device_.controlTransferOut({
111.                  'requestType': 'class',
112.                  'recipient': 'interface',
113.                  'request': 0x22,
114.                  'value': 0x01,
115.                  'index': 0x02}))
116.              .then(() => {
117.                readLoop();
118.              });
119.        };
120.
121.        serial.Port.prototype.disconnect = function() {
122.          return this.device_.controlTransferOut({
123.                  'requestType': 'class',
124.                  'recipient': 'interface',
125.                  'request': 0x22,
126.                  'value': 0x00,
127.                  'index': 0x02})
128.              .then(() => this.device_.close());
129.        };
130.
131.        serial.Port.prototype.send = function(data) {
132.          return this.device_.transferOut(4, data);
133.        };
134.      })();
```

I didn't write the rest of this code below, I got it from github, but what it does is it asks the Arduino to connect or disconnect.

Acknowledgements

I would like to thank the numerous online forums that I used, including Arduino, GitHub and Stack Overflow, my brother for helping me debug bits of my code and of course my mum for buying me all of the parts that I needed. This allowed me to start and finish my project.

References

How recycling is actually sorted, and why Australia is quite bad at it, 2021)
The Conversation. 2021. *How recycling is actually sorted, and why Australia is quite bad at it*. [online] Available at: <https://theconversation.com/how-recycling-is-actually-sorted-and-why-australia-is-quite-bad-at-it-121120> [Accessed 21 July 2021].

Renewable Resources Coalition. (2019). *Pollution Causes & Effects: How Does Wide Scale Pollution Happen?*. [online] Available at: https://www.renewableresourcescoalition.org/pollution-causes-effects/ [Accessed 21 Apr. 2021].

Sciencing. 2021. *Negative Effects of Pollution*. [online] Available at: <https://sciencing.com/negative-effects-pollution-5268664.html> [Accessed 22 July 2021].

World Wildlife Fund. (2021). *Pollution | Threats | WWF*. [online] Available at: https://www.worldwildlife.org/threats/pollution [Accessed 21 Jul. 2021].

Department of Agriculture, Water and the Environment. 2021. *Department Of Agriculture, Water And The Environment*. [online] Available at: <https://www.environment.gov.au/protection/ozone/ozone-science/ozone-layert> [Accessed 21 July 2020].

The Great Pacific Garbage Patch | The Ocean Cleanup, 2021)
The Ocean Cleanup. 2021. *The Great Pacific Garbage Patch | The Ocean Cleanup*. [online] Available at: <https://theoceancleanup.com/great-pacific-garbage-patch/?utm_source=google_cpc&utm_medium=ad_grant&utm_campaign=awareness&gclid=EAIaIQobChMI6tv10-L48QIVSCUrCh1Scw-1EAAYASAAEgKr2PD_BwE> [Accessed 22 July 2021].

Cleanaway. 2021. *Sorting recycling at the source | Cleanaway*. [online] Available at: <https://www.cleanaway.com.au/sustainable-future/all-sorts-to-recycle/> [Accessed 22 July 2021].

Green Choices. 2021. *Environmental impacts*. [online] Available at: <https://www.greenchoices.org/green-living/waste-recycling/environmental-impacts> [Accessed 22 July 2021].