Project Implementation Progress:

Learning the basics of PJ5:

<https://p5js.org/reference/>

Base Code GUI for Building Block Activity:

let value = 0;

let shapes = 0;

function setup() {

createCanvas(400, 400);

textSize(22);

textAlign(CENTER, CENTER);

}

function mouseClicked() {

if (value === 0) {

value = 1;

} else {

value = 1;

}

}

function draw() {

background(220);

fill('blue');

rect(0,0,400,10);

rect(0,0,10,400);

rect(390,0,10,400);

rect(0,390,400,10);

fill('light grey');

rect(150,200,100,30);

fill('black');

text('Begin',30,30,340,370);

if (value == 1){

clear();

background(220);

fill('blue');

rect(0,0,400,10);

rect(0,0,10,400);

rect(390,0,10,400);

rect(0,390,400,10);

//Start Building Block Game Here

if (shapes == 0) {

fill('red')

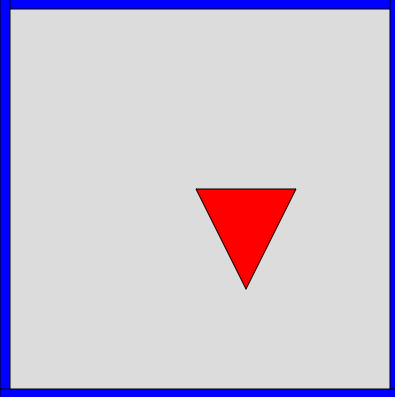
triangle(mouseX, mouseY + 100, mouseX - 50, mouseY , mouseX + 50, mouseY);

translate(mouseX, mouseY);

}

}

}



Spent a majority of time figuring out what PJ5 can do and it’s properties.

So far the program:

Creates the template

Prompts the user to click to begin

A triangle is placed on the cursor to begin the matching game

------------------------------------------------------------------------------------------------------

const letters = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'

function setup() {

createCanvas(400, 400);

textSize(50);

}

function draw() {

background(220);

const rand = int(random(0, letters.length - 1));

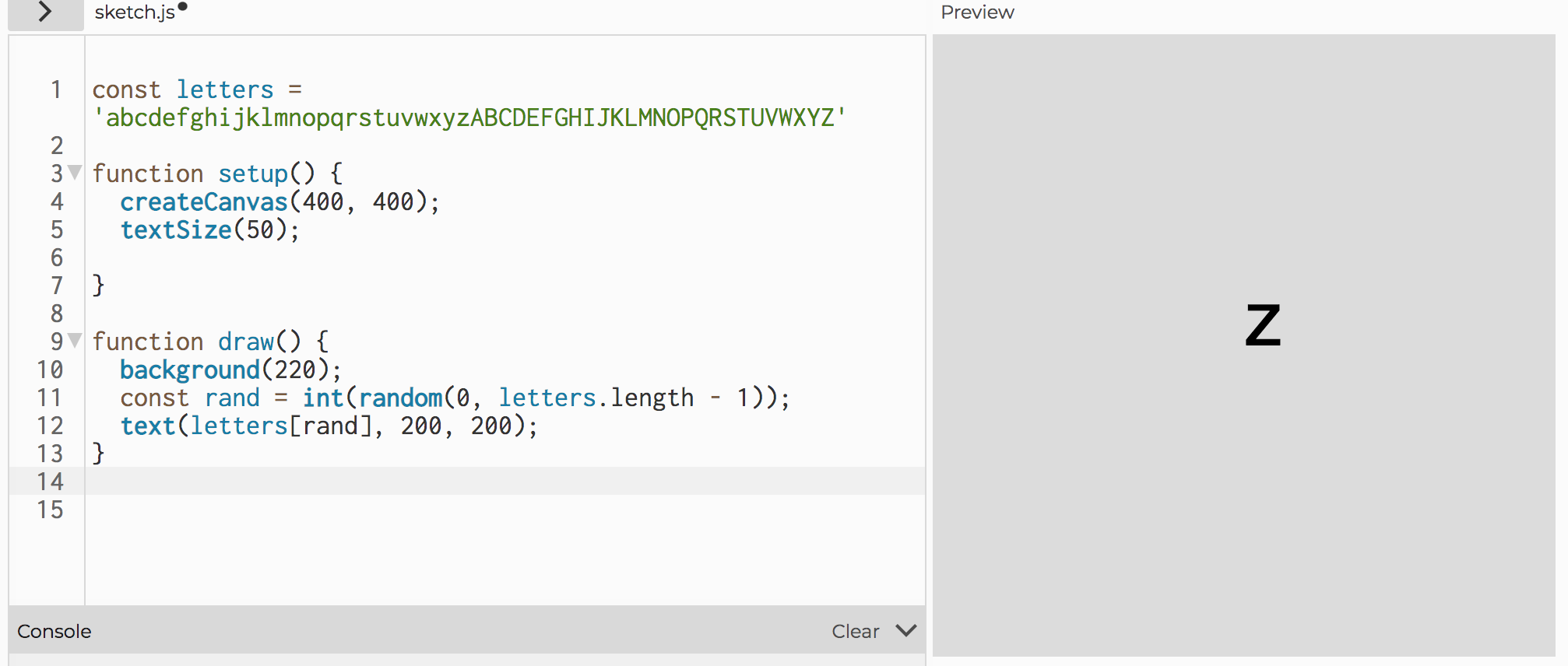
text(letters[rand], 200, 200);

}

I spent my time trying to figure out a way to display a singular, random letter to be displayed onto the screen for the toddler to trace.

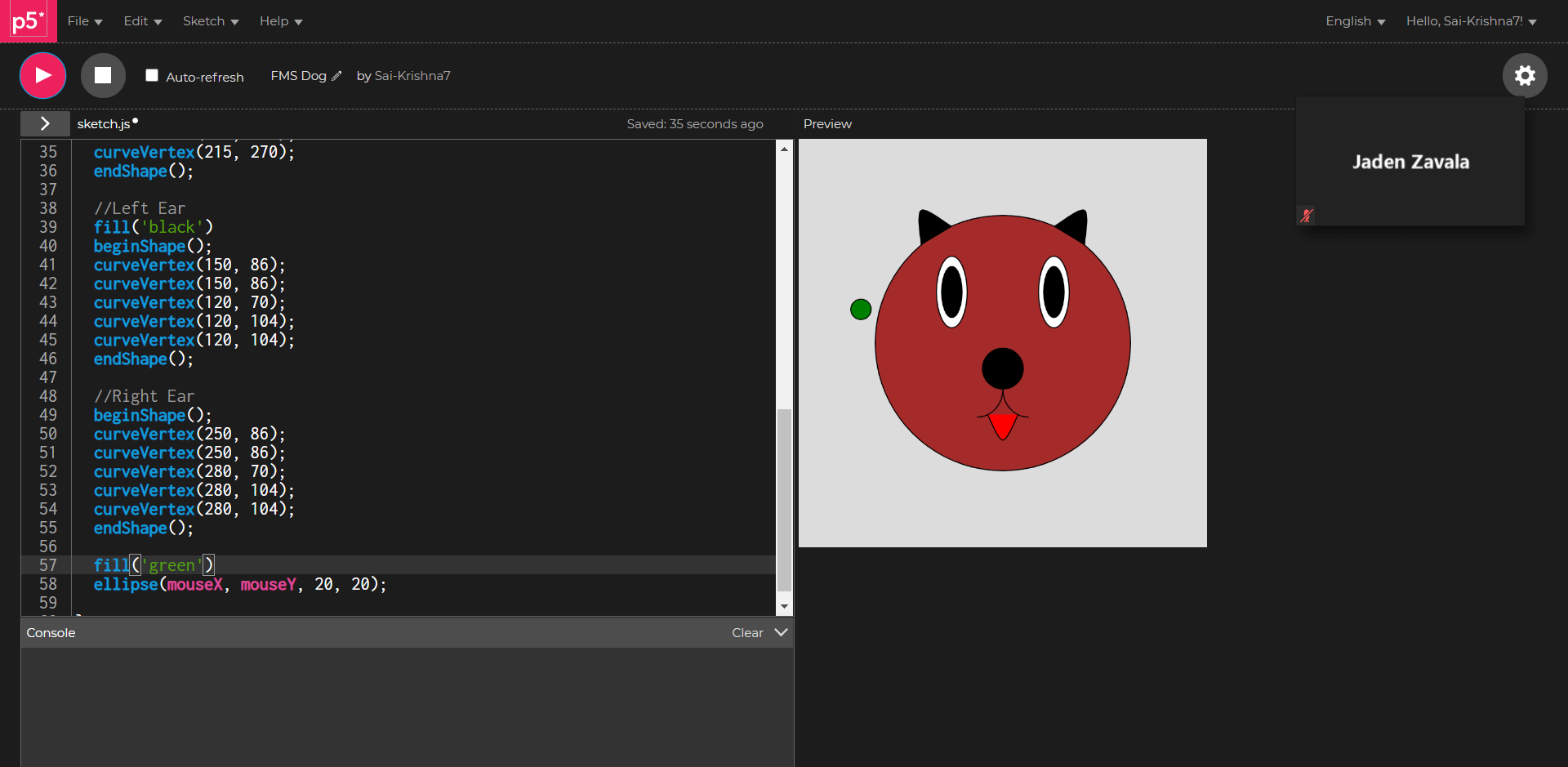
So far the program:

Displays a series of letters in the middle of the screen rather than one



-------------------------------------------------------------------------------------------------------------------------------

**UI Interaction For Tracing the Pattern**

**function setup() {** 

**createCanvas(400, 400);**

**}**

**function draw() {**

**background(220);**

**fill('brown')**

**circle(200, 200, 250);**

**fill('black')**

**circle(200, 225, 40);**

**fill('white')**

**ellipse(150, 150, 30, 70);**

**ellipse(250, 150, 30, 70);**

**fill('black')**

**ellipse(150, 150, 20, 50);**

**ellipse(250, 150, 20, 50);**

**noFill();**

**arc(175, 245, 50, 55, 0, HALF\_PI);**

**arc(225, 245, 50, 55, HALF\_PI, PI);**

**//Tongue**

**fill('red')**

**beginShape();**

**curveVertex(185, 270);**

**curveVertex(185, 270);**

**curveVertex(200, 295);**

**curveVertex(215, 270);**

**curveVertex(215, 270);**

**endShape();**

**//Left Ear**

**fill('black')**

**beginShape();**

**curveVertex(150, 86);**

**curveVertex(150, 86);**

**curveVertex(120, 70);**

**curveVertex(120, 104);**

**curveVertex(120, 104);**

**endShape();**

**//Right Ear**

**beginShape();**

**curveVertex(250, 86);**

**curveVertex(250, 86);**

**curveVertex(280, 70);**

**curveVertex(280, 104);**

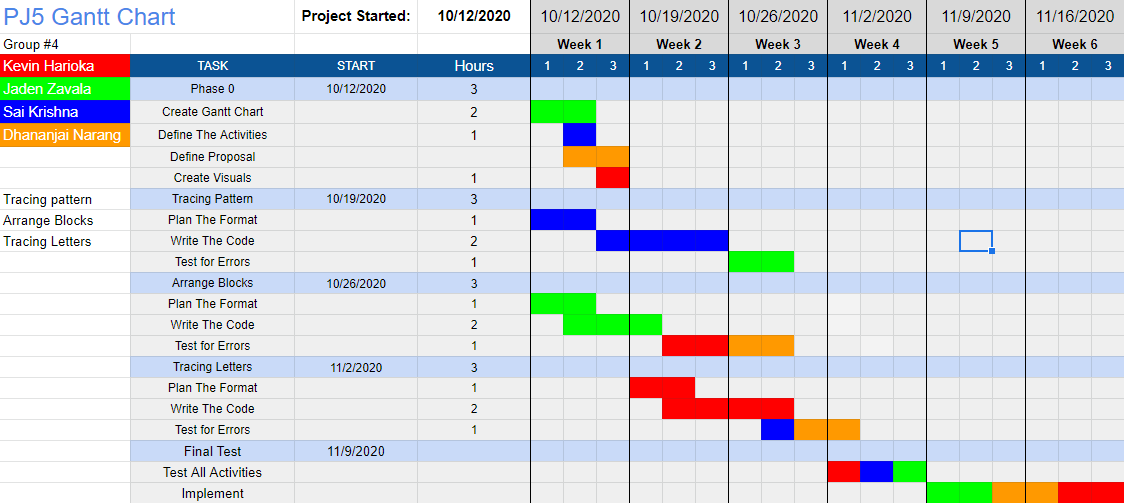
**curveVertex(280, 104);**

**endShape();**

**fill('green')**

**ellipse(mouseX, mouseY, 20, 20);**

**}**

****