Sunny Sun

Devyn Bushelaibi

Professor Mathura Govindarajan

IM-UH 1010

13 May 2022

Splash Pop

Final URL: https://editor.p5js.org/Devynnn/sketches/mHUbrT55I

Our final project will be a shooting game that involves a physical gun built with the arduino kit and a canvas with targets on p5. The targets consist of balloons with different colors of paint that when it shoots it will splash onto a canvas and make an art piece. Starting with P5.js, We'll attempt to make 3 pages, Start game, the game itself, and the ending "showcasing your art piece".

BEFORE:

Originally, here were our first steps to progress with our project:

1. Get the accelerometer working ~
2. Get the accelerometer to move something on p5
3. Get paint splatter visuals
4. Get the button to work
5. Get the button and accelerometer to work together

For p5, so far we have coded a canvas that has a square target. Although originally we were going to use ellipses, the square is easier to code for (this may be a temporary design).

Then, we used the function “splatter” to create a paint splatter effect that replaces the square when the mouse is clicked. This is the code we have for mousePressed for the square

```
if (
  mouseX >= 250 &&
  mouseX <= 350 &&
  mouseY >= 250 &&
  mouseY <= 350 )
```

When we add arduino to the project, we will replace mousePressed with the button on the arduino board clicking when the weapon is pointed at the target (the square), and the paint splatter will signal that the target was shot at. One issue we ran into for the code was the visual of the paint splatter. As of now, we are able to achieve the visuals we wanted only when we press the mouse for a longer duration and move the mouse around, when ideally the visual can be achieved by clicking the mouse once. This is the code I referenced for the splatter effect: https://editor.p5js.org/kwertyops/sketches/V7RIjyVpd.

As for the arduino, we were inspired by carnival guns where the guns would be attached to the table and can be used to shoot targets from afar. This was the hardest part of our project, where the calculations were just confusing for the most part. Until Devyn asked her classmates and professors for alternatives. Our life saver, Professor Aaron Sherwood, suggested to use Brightest point instead. Which seemed more convenient to us. This is the code I referenced for brightest point: https://editor.p5js.org/kylemcdonald/sketches/r17RTIc3Q

AFTER:

We realized that the code to splash the paint was faulty, which made us think of making our own changes to the splatter effect. Instead of using the code, we used images of the drawings that Devyn created specifically for this game.

We needed to reference a code for BrightestPoint, and in class code for handshake which connects the arduino to p5. At first, we had mousePressed for the balloon splatter effect on p5, which we changed to the button value of data = 1 which is the button being pressed that results in the splatter effect.

As for the brightest point, we altered the whole code. Starting by hiding the camera screen. Here's what we came up with:

```
function findBrightest(video) {
 let brightestValue = 0;
 let brightestPosition = createVector(0, 0);
 let pixels = video.pixels;
 let i = 0;
 for (let y = 0; y < h; y++) {
  for (let x = 0; x < w; x++) {
```

```
        let r = pixels[i++];
        let g = pixels[i++];
        let b = pixels[i++];
        i++; // ignore a
        let brightness = r + g + b;
        if (brightness > brightestValue) {
          brightestValue = brightness;
          brightestPosition.set(x, y);
        }
      }
    }
    return brightestPosition;
  }
```

In order for the brightest point to shoot the balloon we wrote a code that consists of :

```
let t = frameCount / 60;
  if (frameCount % 60 === 0 && !purged) {
    balloons.push(new Balloon(random(400, width - 400), height));
  }

  capture.loadPixels();
  if (capture.pixels.length > 0) {
```

```
    // don't forget this!
    brightest = findBrightest(capture);
  }

  let radius = 20;
  noStroke();
  fill(0, 0, 0);
  ellipse(brightest.x, brightest.y, radius, radius);

  if (!purged && endtime < frameCount) {
    purged = true;
    for (let i = 0; i < balloons.length; i++) {
      if (!balloons[i].splatter) {
        balloons[i].hidden = true;
      }
    }
  }
 }
}
```

We wanted to have the crosshair to be visible. Instead of the lines that appeared everywhere the light was visible to, we added a bigger size and changed its color to black

```
 let radius = 20;
   noStroke();
   fill(0, 0, 0);
   ellipse(brightest.x, brightest.y, radius, radius);
```

We added a class for the balloons to be hidden behind the actual illustration.

```
class Balloon {
  constructor(xPos, yPos) {
    this.x = xPos;
    this.y = yPos;
    this.speed = 8;
    this.splatter = false;
    this.deleteAt = -1;
    this.hidden = false;
    this.index = Math.floor(random(0, 5));
  }
  drawBalloon() {
    push();
    imageMode(CENTER);
    if (!this.hidden) {
      if (this.splatter) {
```

```
    //change 70, 100 to make bigger (width, height)
    image(poppedlist[this.index], this.x, this.y, 300, 300);
   } else if (!this.hidden) {
    image(unpoppedlist[this.index], this.x, this.y, 200, 300);
   } //if hidden, display nothing.
  }
  pop();
}
moveBalloon() {
  this.y = this.y - this.speed;
}
splatterN() {
  this.speed = 0;
  if (
    !(
     canvasX[0] < this.x &&
     this.x < canvasX[1] &&
     canvasY[0] < this.y &&
     this.y < canvasY[1]
    )
  ) {
   this.hidden = true;
  } else {
```

```
      this.splatter = true;
    }
  }
}
```

We loaded the images to the p5 :

```
function setup() {
  bg = loadImage("assets/BG.png");
  bwc = loadImage("assets/BWC.png");
  createCanvas(1440, 780);
  for (let i = 0; i < 1; i++) {
    balloons[i] = new Balloon(random(400, width - 400), height);
    bTimes[0] = 0;
    for (let i = 1; i < numBalloons; i++) {
      bTimes[i] = ceil(random(bTimes[i - 1], bTimes[i - 1] + 5));
    }
  }
```

As for the camera, we wanted it to be the same size as the game itself. This allows the player to freely move the target even out of the canvas.

```
 capture = createCapture(
```

```
  {
    audio: false,
    video: {
      width: w,
      height: h,
    },
  },
  function () {
    console.log("capture ready.");
  }
);
capture.elt.setAttribute("playsinline", "");
capture.size(w, h);
createCanvas(w, h);
capture.hide();


 capture.loadPixels();
  if (capture.pixels.length > 0) {
    // don't forget this!
    brightest = findBrightest(capture);
  }
```

We also wanted to make an animation effect where hands would pop out of the screen towards the center of screen 1 with the title of the game and used key space to start the serial port.

```
function draw() {
  background(bg);
  //image(bwc, 325, 0);
  if (screen === 0) {
    push();
    if (hand_pos > height / 6 - 13) {
      translate(0, hand_pos);
      hand_pos -= 3;
    } else {
      translate(0, height / 6 - 13);
    }

  fill(71, 120, 255);
      textSize(40);
      textAlign(CENTER);
      textFont(font);
      text("PRESS ENTER TO START GAME", width / 2, text_1);
      if (!serialActive) {
```

```
      text("Press Space Bar to select Serial Port", width / 2, text_2);
    } else {
      text("Connected", width / 2, text_3);
    }
  } else {
    fill(71, 120, 255);
    textSize(40);
    textAlign(CENTER);
    textFont(font);
    text("PRESS ENTER TO START GAME", width / 2, text_1);
    if (!serialActive) {
      text("Press Space Bar to select Serial Port", width / 2, text_2);
    } else {
      text("Connected", width / 2, text_3);
    }
  }


  pop();
} else if (screen === 1) {
  image(bwc, 325, 0);
```

We did the same with the end screen. Using the enter key, takes the player all the way back to screen 1.

```
fill(71, 120, 255);
    textSize(35);
    textFont(font);
    if (!purged)
      text(`${Math.floor((endtime - frameCount) / 60)} seconds left`, 50, 50);
    else text("GAME OVER, PRESS ENTER TO RESTART", 50, 50);
    translate(width, 0);
    scale(-1, 1);


   //SERIAL CODE- END
    if (!purged && endtime < frameCount) {
      purged = true;
      for (let i = 0; i < balloons.length; i++) {
        if (!balloons[i].splatter) {
          balloons[i].hidden = true;
        }
      }
    }
  }
}
```

```
function restartGame() {
  purged = false;
  balloons = [new Balloon(random(400, width - 400), height)];
  screen = 0;
}


//SERIAL CODE
// code used to connect using spacebar
function keyPressed() {
  if (!serialActive && key == " ") {
    // important to have in order to start the serial connection!!
    setUpSerial();
  }
  if (keyCode == ENTER) {
    if (screen === 0 && serialActive) {
      screen = 1;
      endtime = frameCount + FRAMETIMER;
    } else if (purged && screen === 1) {
      restartGame();
    }
  }
}
//SERIAL CODE- END
```

```
//SERIAL CODE
function readSerial(data) {
  ////////////////////////////////////
  //READ FROM ARDUINO HERE  (BUTTON INFO)
  ////////////////////////////////////

  if (data != null) {
    // make sure there is actually a message
    // split the message
    let buttonInfomation = data;
    print(data);
    if (data == 1) {
      for (let i = 0; i < balloons.length; i++) {
        if (
          dist(
            width - balloons[i].x,
            balloons[i].y,
            width - brightest.x,
            brightest.y
          ) < 100
        ) {
          balloons[i].splatterN();
```

```
    }
  }
```

We also added popped and unpopped balloons to differentiate the colors and the splashes.

```
let unpoppedlist;
let poppedlist;

const canvasX = [350, 1100];
const canvasY = [140, 620];

function preload() {
  unpopped1 = loadImage("./assets/blue.png");
  popped1 = loadImage("./assets/BS.png");
  unpopped2 = loadImage("./assets/yellow.png");
  popped2 = loadImage("./assets/YS.png");
  unpopped3 = loadImage("./assets/green.png");
  popped3 = loadImage("./assets/GS.png");
  unpopped4 = loadImage("./assets/purple.png");
  popped4 = loadImage("./assets/PS.png");
  unpopped5 = loadImage("./assets/red.png");
```

```
  popped5 = loadImage("./assets/RS.png");

  openScreen = loadImage("./assets/openscreen.png");

  music = loadSound("assets/music.ogg");

  font = loadFont("oleo.ttf");

  unpoppedlist = [unpopped1, unpopped2, unpopped3, unpopped4, unpopped5];
  poppedlist = [popped1, popped2, popped3, popped4, popped5];
}
```

For the final touches of our game, we added a timer, replay button, and background music.

```
const TIMER = 60;
const FRAMETIMER = 60 * TIMER;
let endtime;
```

We also wanted the background music to appear in full screen,

```
function mousePressed(){
 if (!musicPlay) {
 music.loop();
  musicPlay = !musicPlay;
 }
```

We then packaged the arduino into a gun and made a box for the computer to shield out light as the BrightPoint code was correlated to the LED light on the end of the gun, and other sources of light made the shooting inaccurate.

Reflections:

Sunny:

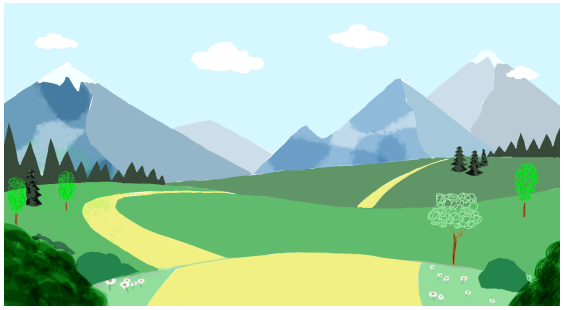
I enjoyed working on this project as it incorporated a lot of creative elements. For my part of the project, I worked on parts of the code in p5 such as creating a class for the balloons, the text for the start and end of the game, and the code for connecting p5 with arduino. For most of the project, Dev and I worked together and troubleshooted together. I enjoyed the coding aspect, and mainly struggled with the BrightestPoint code and the arduino code. The wiring also took a lot of troubleshooting, as we had the button malfunction multiple times. However, the repetition of the wiring allowed me to better understand circuits. I also went over previous knowledge through coding for classes, and other elements in our code such as textFont. I believe

our project could've been better if we soldered before building a casing around the arduino, as well as made a proper end screen. From the showcase, I got helpful feedback that a counter for the balloons that were shot would have been a nice addition.

Devyn:

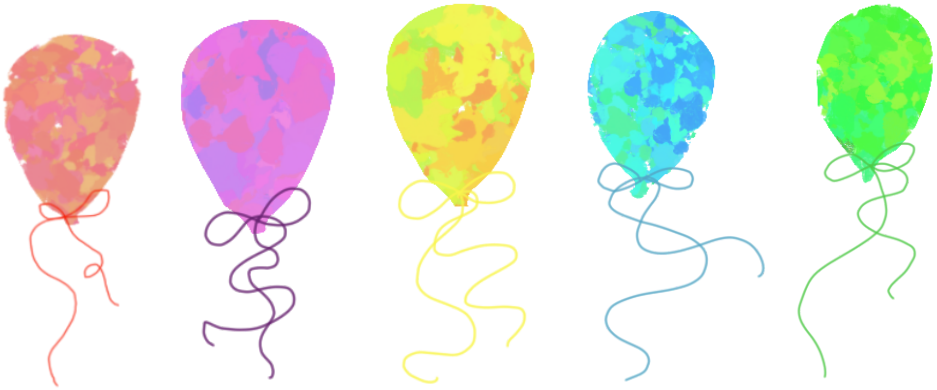
I loved the fact that we coordinated both ideas into one, for instance. She wanted colors indicating voice while I wanted a target game. Which later became a paint splash shooting game. I enjoyed the process of making the background and the balloons and being able to incorporate my artistic skills to my technical skills. I also loved the process of building the gun and the hardships that followed it. For instance, burning my arduino to almost burning my friend's arduino. This taught me how to correctly place the wires and stick them in a proper manner. I believe our project could've been better if we had not procrastinated for a while. Without the help and the suggestions from my peers, I wouldn't have made it here.

Art Images:



background

board



Balloons

Splashes

Heres our arduino code:

```
const int BUTTON = 5;
void setup() {
  Serial.begin(9600);
  pinMode(BUTTON, INPUT);

 //uncommand these lines
  // start the handshake
  //while (Serial.available() <= 0) {
   // Serial.println(0); // send a starting message
    //delay(300);            // wait 1/3 second
  //}
}

void loop() {

        int buttonValue = digitalRead(BUTTON);
      Serial.println(buttonValue);
  // deleted these three lines
  // int buttonValue = digitalRead(BUTTON);
  // delay(1);
  // Serial.println(buttonValue);
  // wait for data from p5 before doing something
//  while (Serial.available()) {
//    if (Serial.read() == '\n') {
//      //currently we are not doing ANYTHING with the infor from p5
//      //lets get button information to send to p5
//
//    // uncommand these lines
//      int buttonValue = digitalRead(BUTTON);
//      Serial.println(buttonValue);
//      delay(1);
//    }
//  }
}
```