NANYANG TECHNOLOGICAL UNIVERSITY
SEMESTER 2 (2019)
DD1010 — Experimental Interaction

Final — Provocative Objects
CREATIVE PROCESS JOURNAL

Ashley Tan
Eugene Ho
G05

Table of Contents

INTRODUCTION
—
IDEATION AND RESEARCH
—
FLOW OF INTERACTION
—
PROCESS (PHYSICAL SET-UP)
—
PROCESS (CODING)
—
PROCESS (VIDEO)
—
IMAGES
—
INSIGHTS AND REFLECTION

INTRODUCTION

Bellwire is a provocative object that aims to create awareness about the use of bell curve grading system that is being adopted in all schools, and to protest against this unfair grading system.

Through a game of buzz wire which mimics the shape of the bell curve, loop wand that resembles a pen, together with books surrounding the set-up, players get to experience first-hand on how it feels like to be treated under this system in an academic setting.

There is a projection with videos along with the process of the whole gameplay which guides the player through with instructions, time limit, as well as revealing to them the purpose behind this whole project.

Ultimately, this initiative is part of an effort to save our grades in school, which is being drowned out by the use of the bell curve system.

IDEATION AND RESEARCH (RESEARCH)

Aim: To protest against the bell curve grading system used in schools

What is bell curve grading system

- Follows the normal distribution graph which looks like a symmetric curve that is pronounced in the middle and tapered off at the edges. As such, the middle portion under the curve contains more area than either of the ends.

- Two ways teachers can grade through the bell curve grading system:
  1. The “scaled to 100” method which takes the difference between the highest score and the highest possible score, E.g. Total score is 100, and if the highest grade is an 85, then they add 15 points to everyone’s score.
  2. The “average C” method sets an average point (e.g. 75 is a common point). The professor takes the average score and scales it to the set point, then adds (or subtracts) points based on the difference. E.g. If there’s a 60, a 90, and a 60, the average is 70. It takes 5 points to get the average to a 75. Thus the two 60s become 65s and the 90 becomes a 95.

Pros of bell curve grading system

- Grading on a curve automatically factors in the difficulty of the tests and/or assignments, and because instructors assign grades according to relative performance, students can still earn a good grade in a class that they find extremely difficult. E.g. if a student scores 50% on their exams and the class average is 30%, then they will likely be in the ‘A’ range because they scored higher than their classmates. Also, for the students who scored 30%, instead of getting an F grade, the curve will place them nicely within the ‘B’ range.

- The curve encourages healthy competition. When graded on a curve, many people like to compete amongst their friends to see who will come out with the highest mark when all is said and done. This gives people an extra incentive to study hard and aim for success.

IDEATION AND RESEARCH (RESEARCH)

Aim: To protest against the bell curve grading system used in schools

Cons of bell curve grading system

- Students who perform slightly above or below the average can often miss out on a better grade. So, it is possible for a student who earned a straight B to end up with a C as a final curved score. Students may also feel less in control of how well they do in class, which can cause anxiety and stress levels to spike before tests and exams.

- For professors, curved grading might make it more difficult to compare groups of students with each other in terms of performance. For instance, if all students perform poorly, even the highest-scoring students may fail to meet class standards.

- The curve is not good to use for small classes (less than 100 students). There is simply too much room for error. There could be a case where 30 out of the 100 students are 'extremely' proficient with the course material. Even though these students would all score in a similar range on exams, only a small percentage of them will be guaranteed to get a top grade. This essentially makes it 'very' hard to get a top mark. Furthermore, the cut-off grades may be very close to each other; it may be a mere 4% difference that separates an A+ from a B.

- The curve has the potential to foster unhealthy competition. With the knowledge that they 'must' outperform their peers, some students may adopt a rather, disingenuous attitude towards their fellow classmates. They may not be willing to provide help, or worse yet, may deliberately feed others misinformation in order to get ahead.

- High class averages are not always favourable. For example, the class average for the exam scores is 85%. The students who score the average score of 85% will most likely get a 'B' grade, and a student who scores 70% will most likely be in the 'C' range or lower instead.

IDEATION AND RESEARCH (iDEATION)

In relation to school

Studied hard in school

Knowing how to do the exam quesitons/
Fulfill criteria for assignment

Expecting to get an A

Not getting the A

Due to the limited amount of ‘A’s given to student following the adjusted bell curve

In relation to bellwire

Effort made to play the game

Fulfilling targeted time

Expecting to receive a prize

Not receiving a prize

Due to the limited amount of prizes given out following the adjusted bell curve

IDEATION AND RESEARCH (RESEARCH)

Aim: To protest against the bell curve grading system used in schools

Internet Activism

Internet activism (also known as web activism, online activism, digital campaigning, digital activism, online organizing, electronic advocacy and e-activism) is the use of electronic communication technologies such as social media, e-mail, and podcasts for various forms of activism to enable faster and more effective communication by citizen movements, the delivery of particular information to large and specific audiences as well as coordination.

A digital activism campaign is “an organized public effort, making collective claims on a target authority, in which civic initiators or supporters use digital media.” Research has started to address specifically how activist/advocacy groups in the U.S. and Canada are using social media to achieve digital activism objectives.

Hashtag Activism

Hashtag activism is the use of hashtags for fighting or supporting a cause through the usage of social media outlets. Its use has been associated with the 2014 Chibok kidnapping, with hopes that it would help keep the story in the news and raise international attention. The hashtag itself has received 2 million retweets.

One example of the powerful rise of hashtag activism can be seen in the black feminist movement’s use of hashtags to convey their cause. The famous hashtag “IamJada” was an internet backlash to the mocking “#Jadapose” that went viral, ensuing after a sixteen-year-old girl Jada Smith was photographed following her gang rape In this instance, a hashtag was employed to convey a powerful anti-rape message.

IDEATION AND RESEARCH (IDEATION)

Combination of Internet & Hashtag Activism

After researching on both internet activism and hashtag activism, this was where we came up with the final call to action idea in Bellwire, where we encourage our participants to protest online through the use of hashtag. However, we do not want them to post for the sake of posting. We want our participants to experience first-hand the unfairness of the bell curve grading system and go spread awareness online. We feel that it is the most effective way to spread our message given that social media is part of our lives these days.

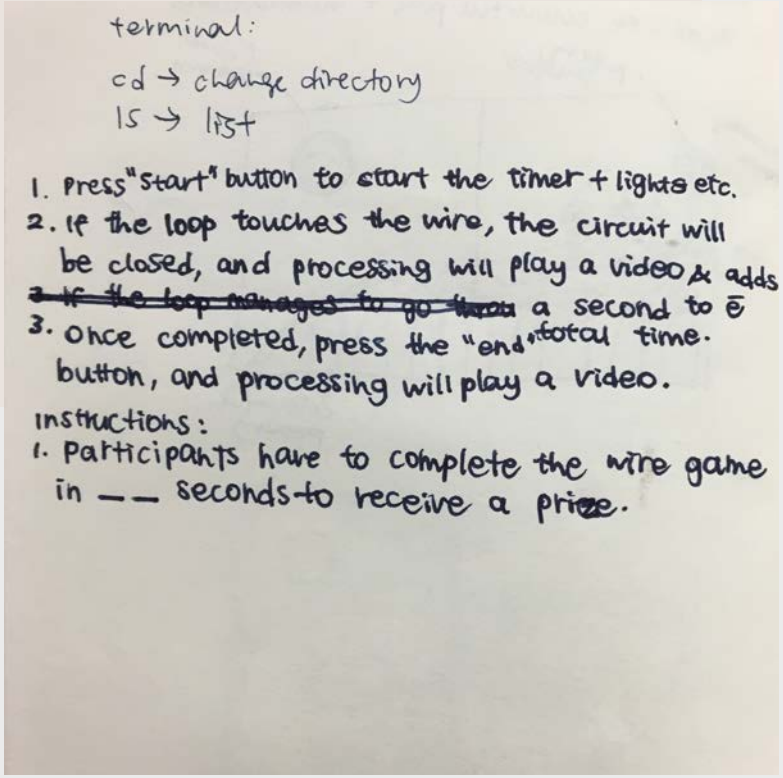
FLOW OF INTERACTION (v1)

1. Press “Start” button to start the timer + lights etc.

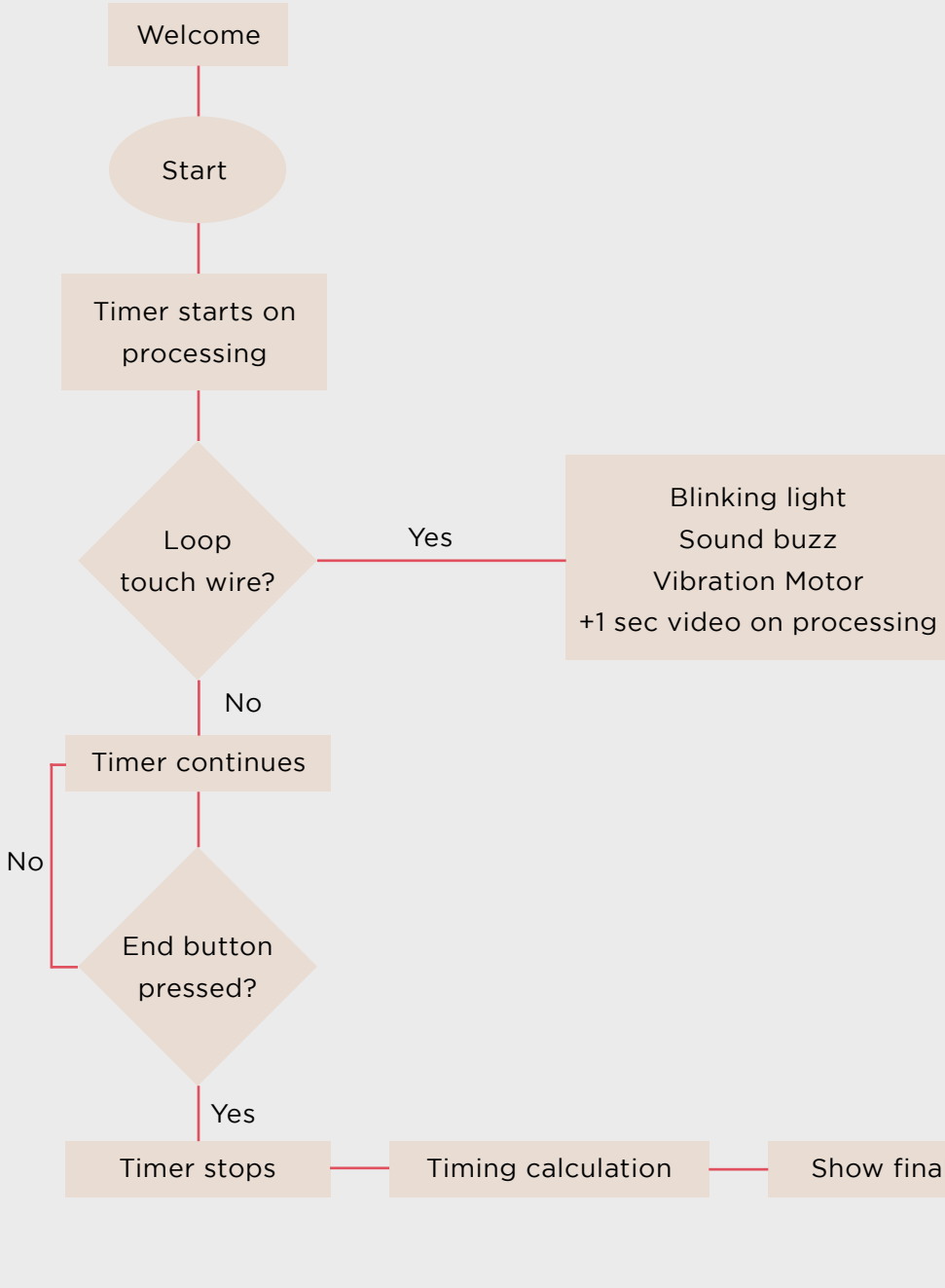
2. If the loop touches the wire, the circuit will be closed, and processing will play a video and adds one second to the total time.

3. Once completed, press “end” button, and processing will play a video.

Instructions:
1. Participants who complete the wire game in xx seconds will receive a prize.
(However, the twist is that at the end after bell curve grading system is applied to the criteria, the xx seconds which was achievable at first is now unacheivable.)



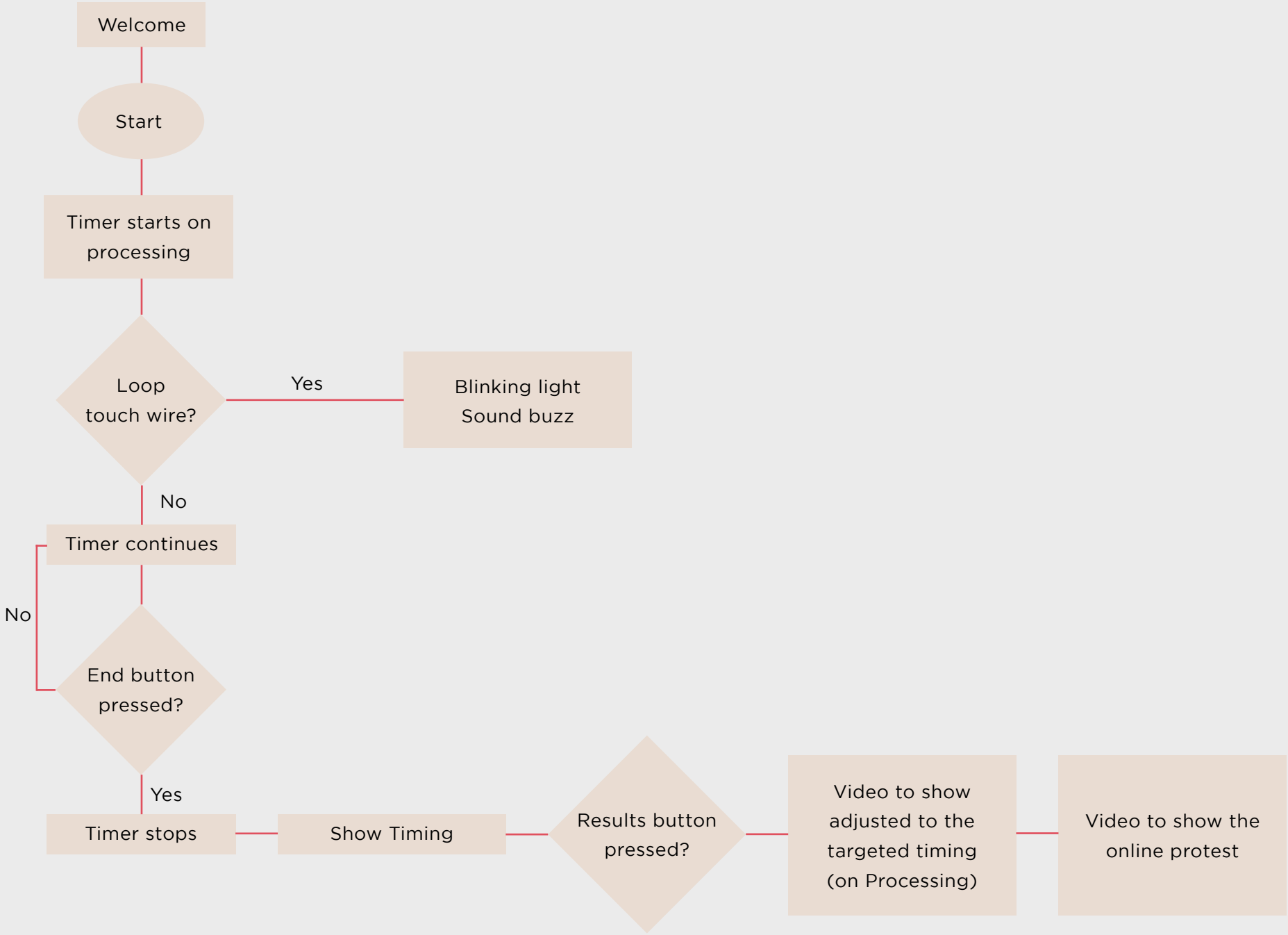
FLOW OF INTERACTION (v1)



Storyline framework introduced by Lei.

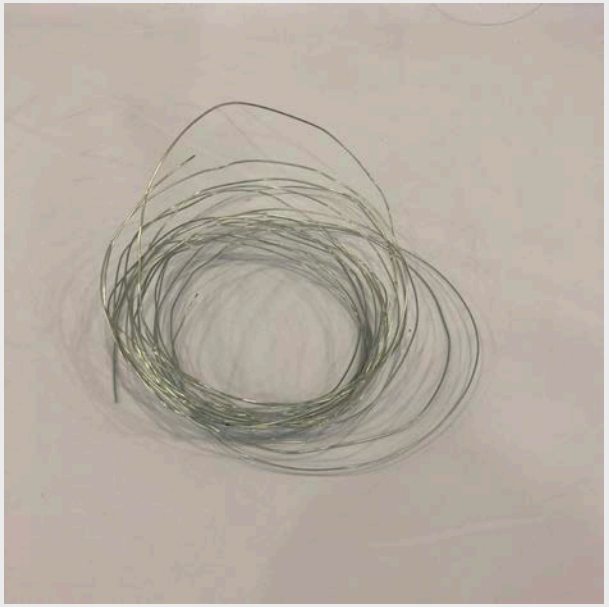
Looks neater & easier to read.

FLOW OF INTERACTION (FINAL)

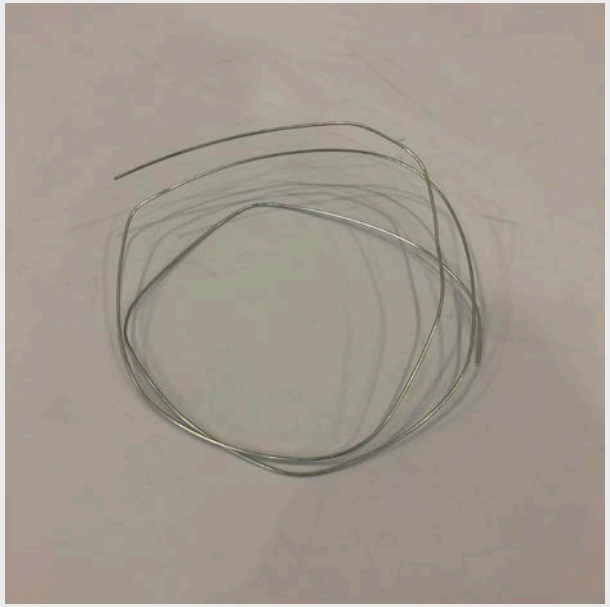


PROCESS OF PHYSICAL SET-UP (Materials Pt. 1)

6x Plywood

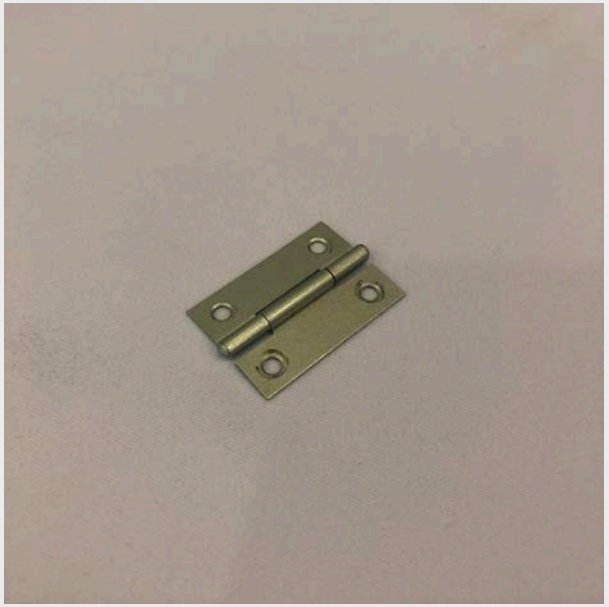


1m Thin steel wire



50cm Thick steel wire

2x Cork



2x Steel hinge

8x Screws

PROCESS OF PHYSICAL SET-UP (Materials Pt. 2)

Drill

White acrylic paint

White string

Pen

PROCESS OF PHYSICAL SET-UP (Building Pt. 1)

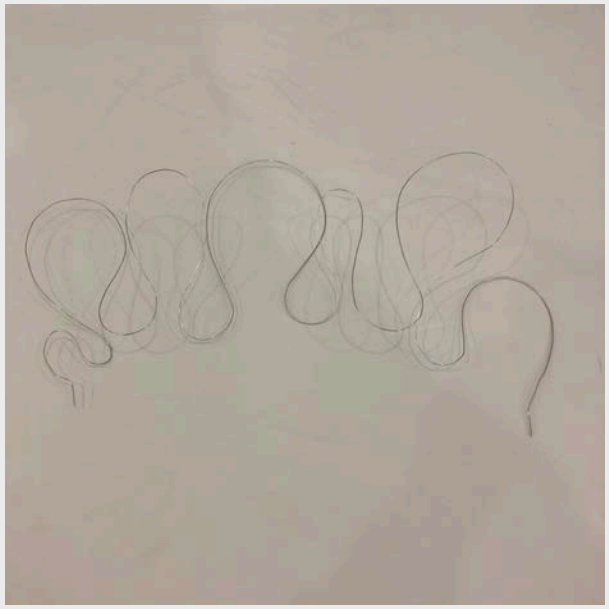
Box from plywood

Box from plywood (painted)

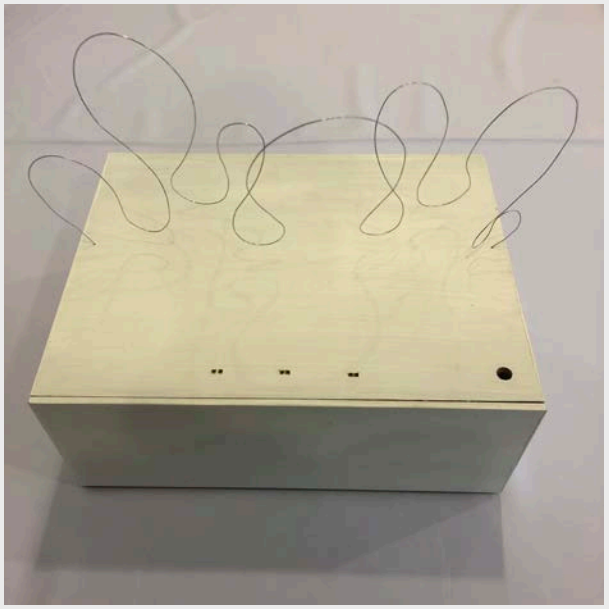
Wand from thick wire

Wand placed inside the pen

PROCESS OF PHYSICAL SET-UP (Building Pt. 2)

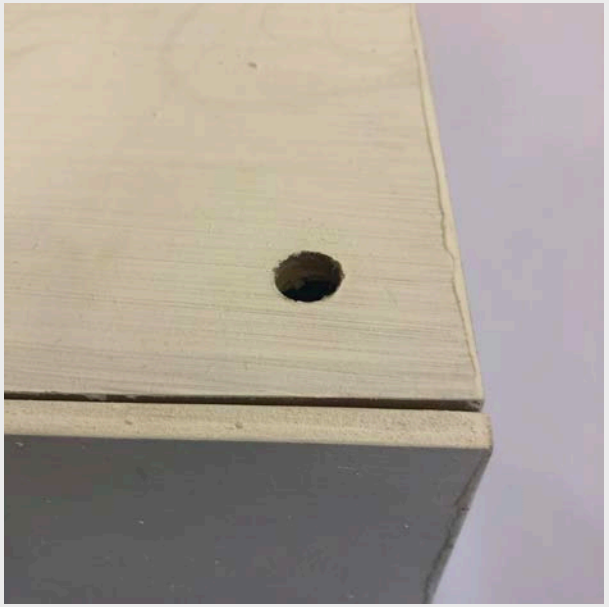


Buzz wire from thin wire

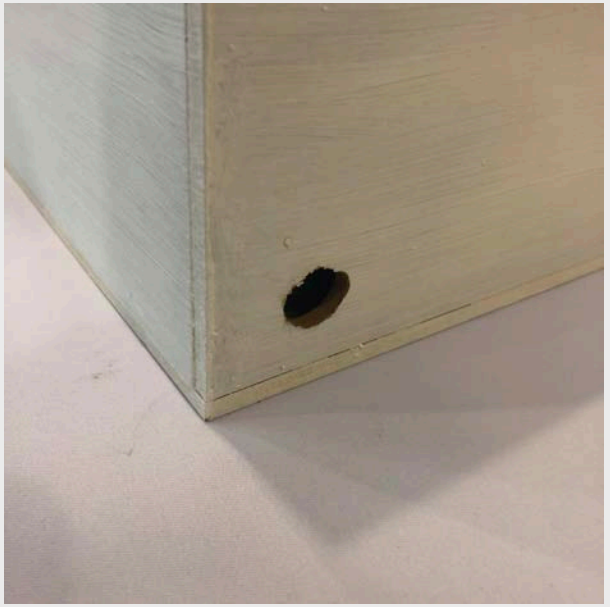


Buzz wire on the box

Holes drilled for buttons



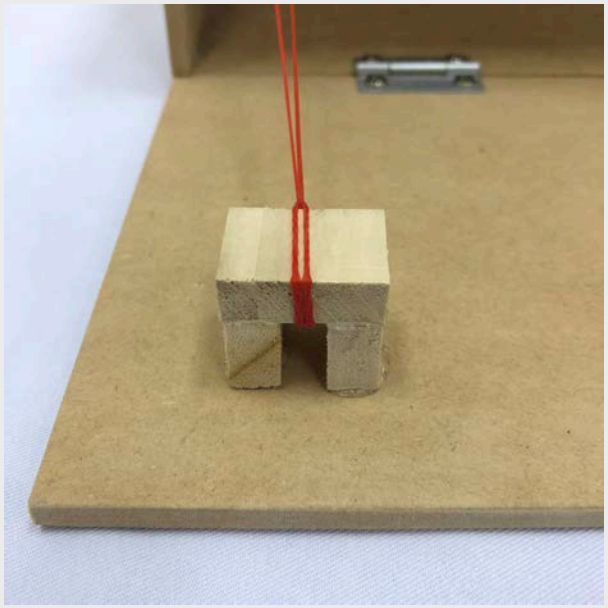
Hole drilled for piezo buzzer



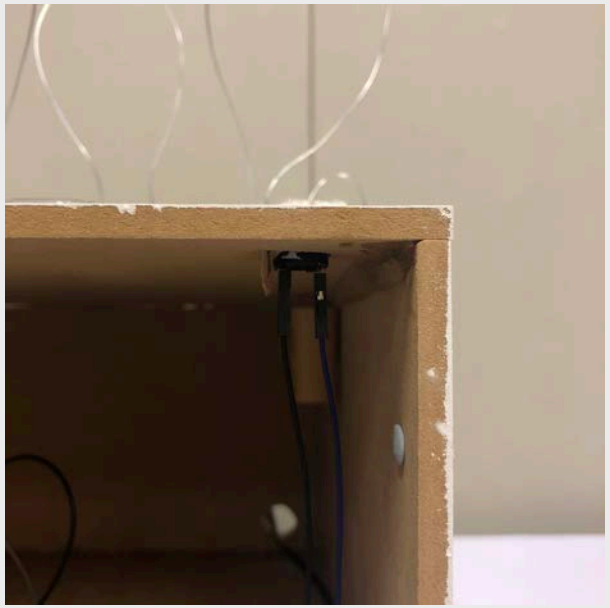
Hole drilled for USB cable

PROCESS OF PHYSICAL SET-UP (Detail Shots)

Hinge built for box to open

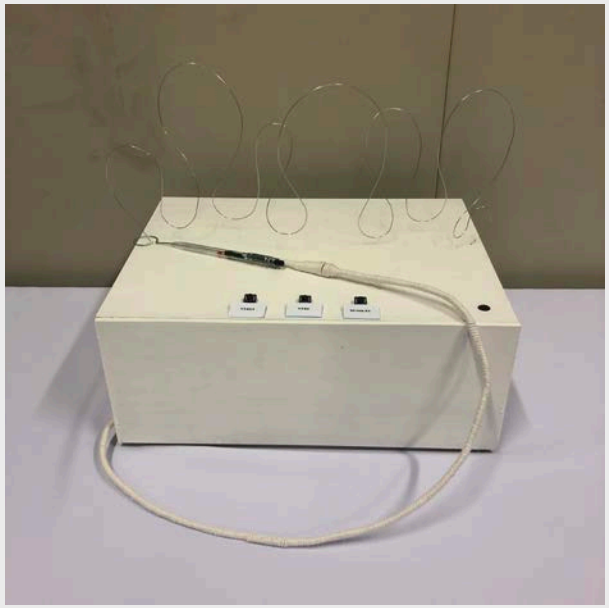


Wood hook to close box

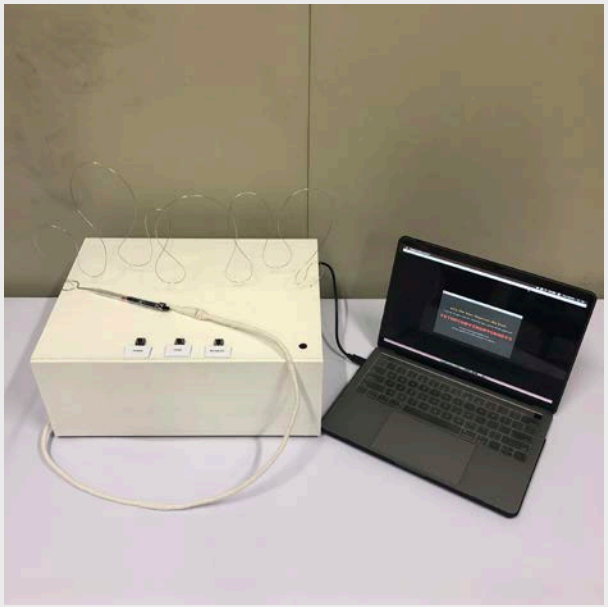


Cork to stabilize buzz wire

PROCESS OF PHYSICAL SET-UP (Set-up Pt. 1)

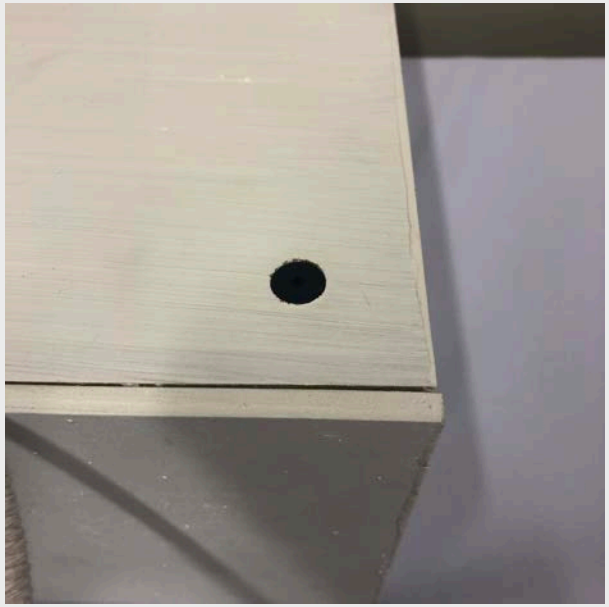


Configuration

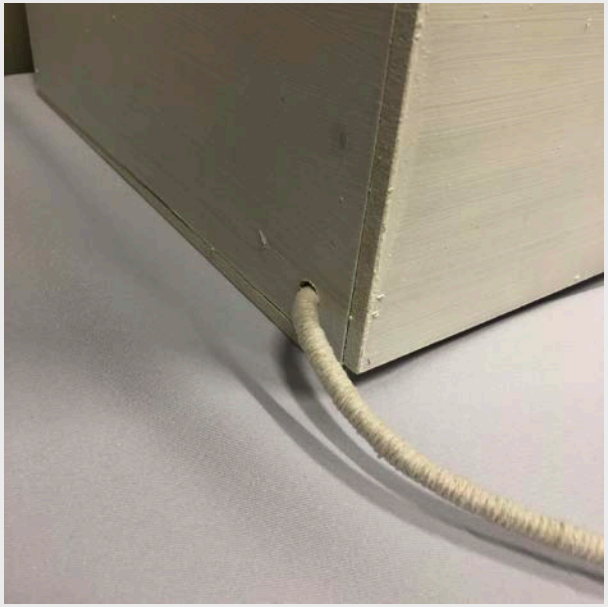


Configuration (with screen)

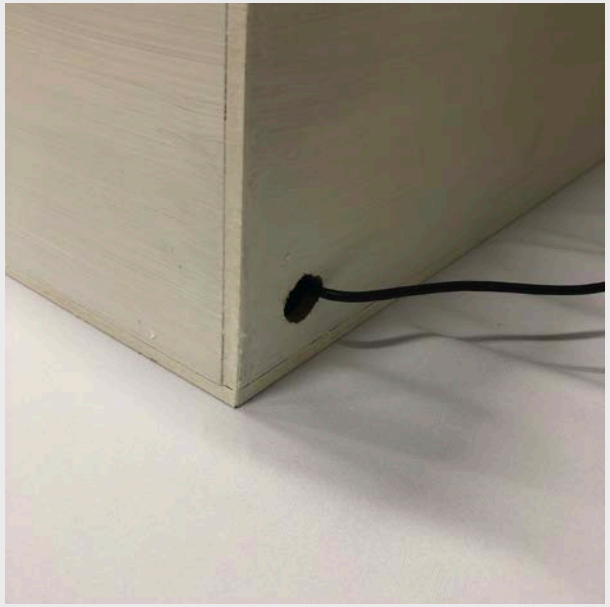
Buttons located in front



Piezo buzzer on the right

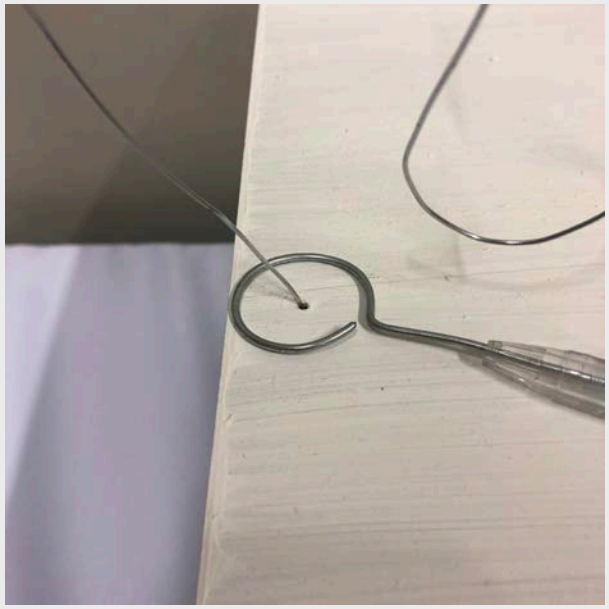


Wand wires on the left

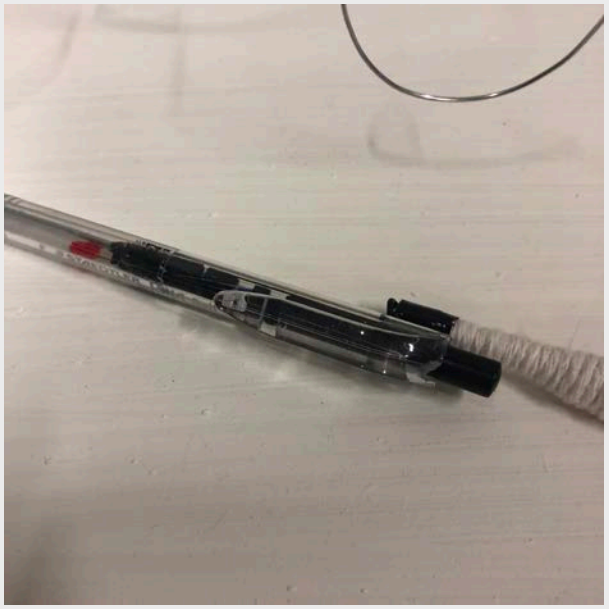


USB cab from the back

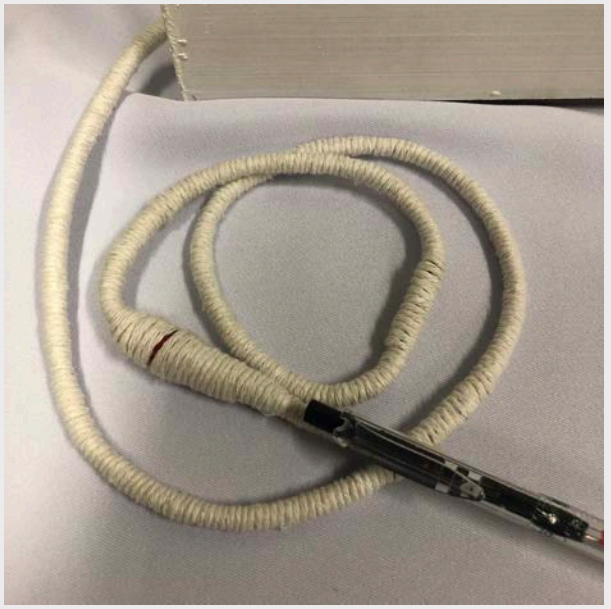
PROCESS OF PHYSICAL SET-UP (Set-up Pt. 2)



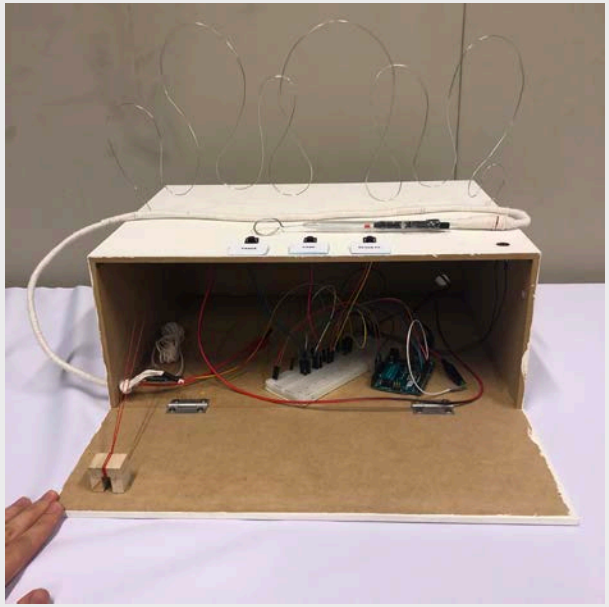
Buzz wire through the box



LED light embedded inside pen (with wand)



String to wrap the wires (for both LED light & wand)



Internal view (Arduino and breadboard kept inside)

PROCESS OF PHYSICAL SET-UP (Reasoning)

How each elements in our physical set-up links to our message behind the gameplay:

1. The bends of the buzz wire resembles the shape of the bell curve graph.

2. The wand is a physical pen, which relates the gameplay to “writing answers in an exam”.

3. The stopwatch timing which represents the time limit given in an exam or any submissions related to school.

4. The books surrounding the box represents academic and educational related institutions who are the ones behind this bell curve grading system.

PROCESS (CODING - ARDUINO) v1

```
int circuitPin = 9;
int speakerPin = 10;
int ledPin = 13
int vibPin = 14
int startPin = 2;
int endPin = 3;

void setup() {
      pinMode(circuitPin, INPUT);    // setup circuit
      pinMode(speakerPin, OUTPUT);   // setup buzzer
      pinMode(ledPin, OUTPUT);       // setup led
      pinMode(vibPin, OUTPUT);       // setup vibrator
      pinMode(startPin, INPUT);      // setup start button
      pinMode(endPin, INPUT);        // setup end button
      Serial.begin(9600);
      // 0: welcome screen
      // 1: timer start
      // 2: +1 sec
      // 3: timer end
     }

void loop() {
  // play welcome video on loop on processing
  // to start game, if start button is pressed:
  if (digitalRead(startPin) == HIGH) {
     Serial.write(1);  // send 1 to processing to start timer
     digitalWrite(ledPin, HIGH)           // light once
     delay(50);
     digitalWrite(ledPin, LOW)
     digitalWrite(vibPin, HIGH);          // vibrate once
     delay(50);
     digitalWrite(vibPin, LOW);
     tone(speakerPin, 440);               // buzz
     delay(50);
     noTone(speakerPin);
```

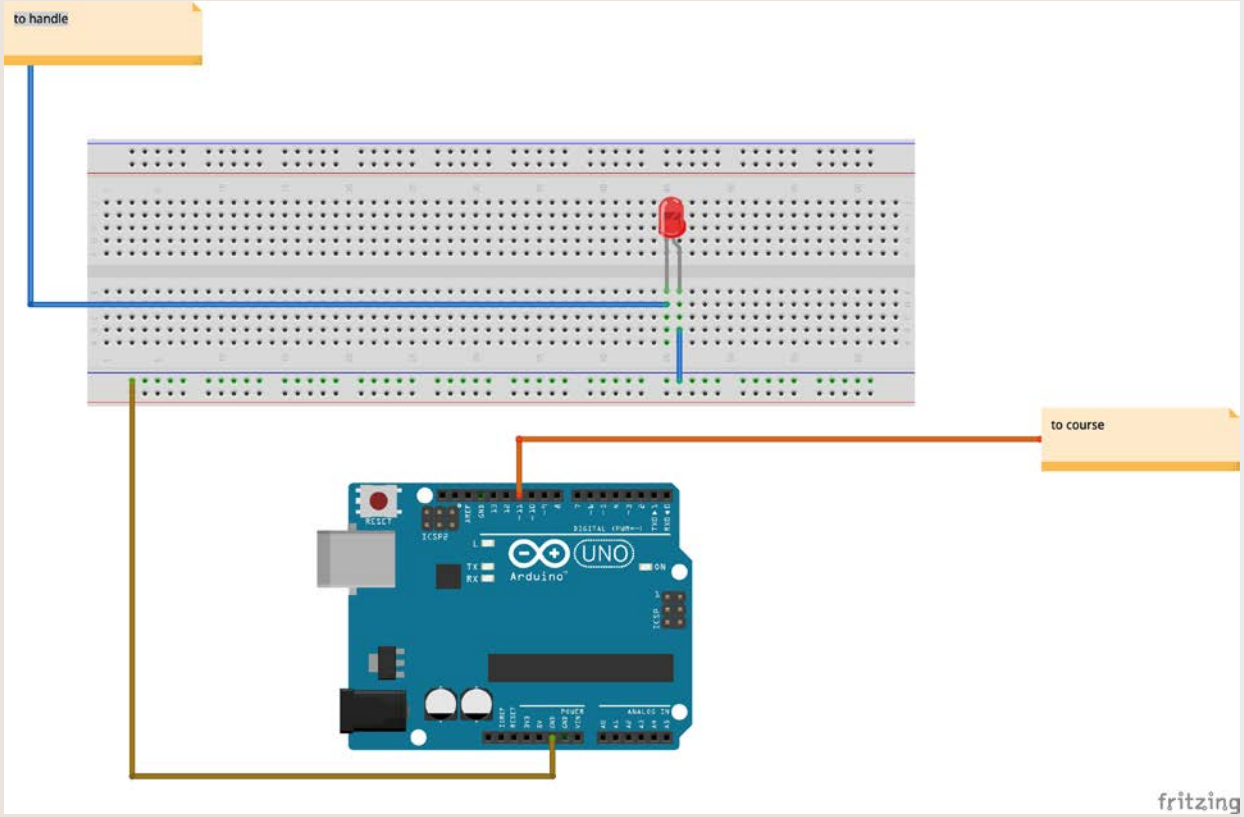
```
// if loop touches the wire
if (digitalRead(circuitPin) == HIGH) {
      Serial.write(2);          // +1 sec video in processing
      digitalWrite(ledPin, HIGH);        // light
      delay(50);
      digitalWrite(vibPin, HIGH);        // start vibration
      delay(50);
      tone(speakerPin, 440);             // buzz
      delay(50);
      } else {            // if loop doesn't touch the wire
      digitalWrite(ledPin, LOW);
      digitalWrite(vibPin, LOW);
      noTone(SpeakerPin);
      } else {        // when start button is not pressed
   Serial.write(0);  // send 0 to processing (welcome screen)
      digitalWrite(ledPin, LOW);
      digitalWritee(vibPin, LOW);
      noTone(SpeakerPin);
  }
}

  if (digitalRead(endPin) == HIGH) { // end button is pressed
   // send 3 to processing to stop timing and play end video
      Serial.write(3);
  } else {                           // continue timing
    // ?
  }

}
```

The first version of the code is just to get a general idea of how the code will be like being testing out.

Note: This does not work!

PROCESS (CODING - ARDUINO) v2

```
int circuitPin = 7;

void setup() {
  pinMode(circuitPin, INPUT);    // setup circuit
  Serial.begin(9600);
}

void loop() {
    digitalWrite(11, HIGH);
}
```



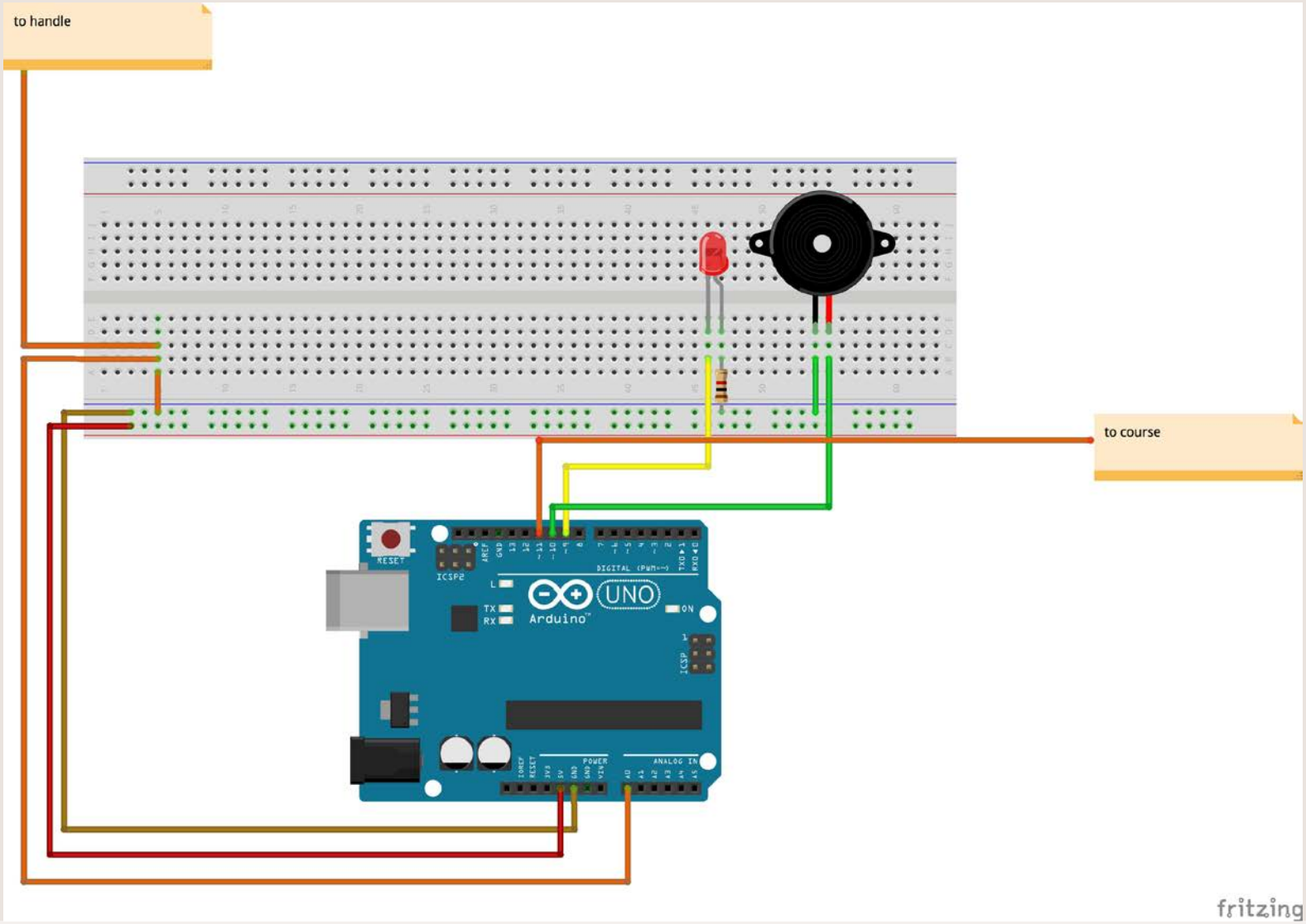
This version of code was done during the play test in class.

When the loop wand touches the wire the LED emits a faint light.

PROCESS (CODING - ARDUINO) v3

```
int readingPin = A0;                        // to test if the circuit is closed
int circuitPin = 11;
int speakerPin = 10;
int ledPin = 9;
int reading = analogRead(readingPin);


void setup() {
    Serial.begin(9600);
    pinMode(circuitPin, OUTPUT);    // setup circuit
    pinMode(speakerPin, INPUT);     // setup buzzer
    pinMode(ledPin, INPUT);         // setup led


}

void loop() {
   if(reading < 10) {              // if loop wand touches the wire
    digitalWrite(ledPin, HIGH);    // led lights up
    tone(speakerPin, 400);         // buzzer buzzes
} else {
    digitalWrite(ledPin, LOW);
    noTone(speakerPin);
}
```

PROCESS (CODING - ARDUINO) v3



Before moving on to processing, we made sure that the circuit is able to work on arduino first. The criteria “if (reading < 10)” is determined by the Serial.println() function tested in another file.

PROCESS (CODING - ARDUINO) v3 (test)

```
const int analogPin = A0;
int circuitPin = 11;

void setup() {
    Serial.begin(9600);
    digitalWrite(circuitPin, HIGH);
}

void loop() {
    int analogValue = analogRead(analogPin);
    Serial.println(analogValue);
    delay(1);
}
```

This method was taught in class to test for the value of current flowing through the analog pin.

The values are seen in the serial monitor.

When the loop wand touches the wire, the value read is 0.

Therefore, we set the criteria for the if statement to be < 10.

PROCESS (CODING - ARDUINO) v4

```
int readingPin = A0;                        // to test if the circuit is closed
int circuitPin = 11;
int speakerPin = 10;
int ledPin = 9;
int reading = analogRead(readingPin);


void setup() {
    Serial.begin(9600);
    pinMode(circuitPin, OUTPUT);     // setup circuit
    pinMode(speakerPin, OUTPUT);     // setup buzzer
    pinMode(ledPin, OUTPUT);         // setup led


}

void loop() {
   if(reading > 10) {              // if loop wand touches the wire
    digitalWrite(ledPin, HIGH);   // led lights up
    digitalWrite(speakerPin, HIGH);       // buzzer buzzes
} else {
    digitalWrite(ledPin, LOW);
    digitalWrite(speakerPin, LOW);
}
}
```

The connection of wires are the same as v3, just that we changed the code for the speakerPin.

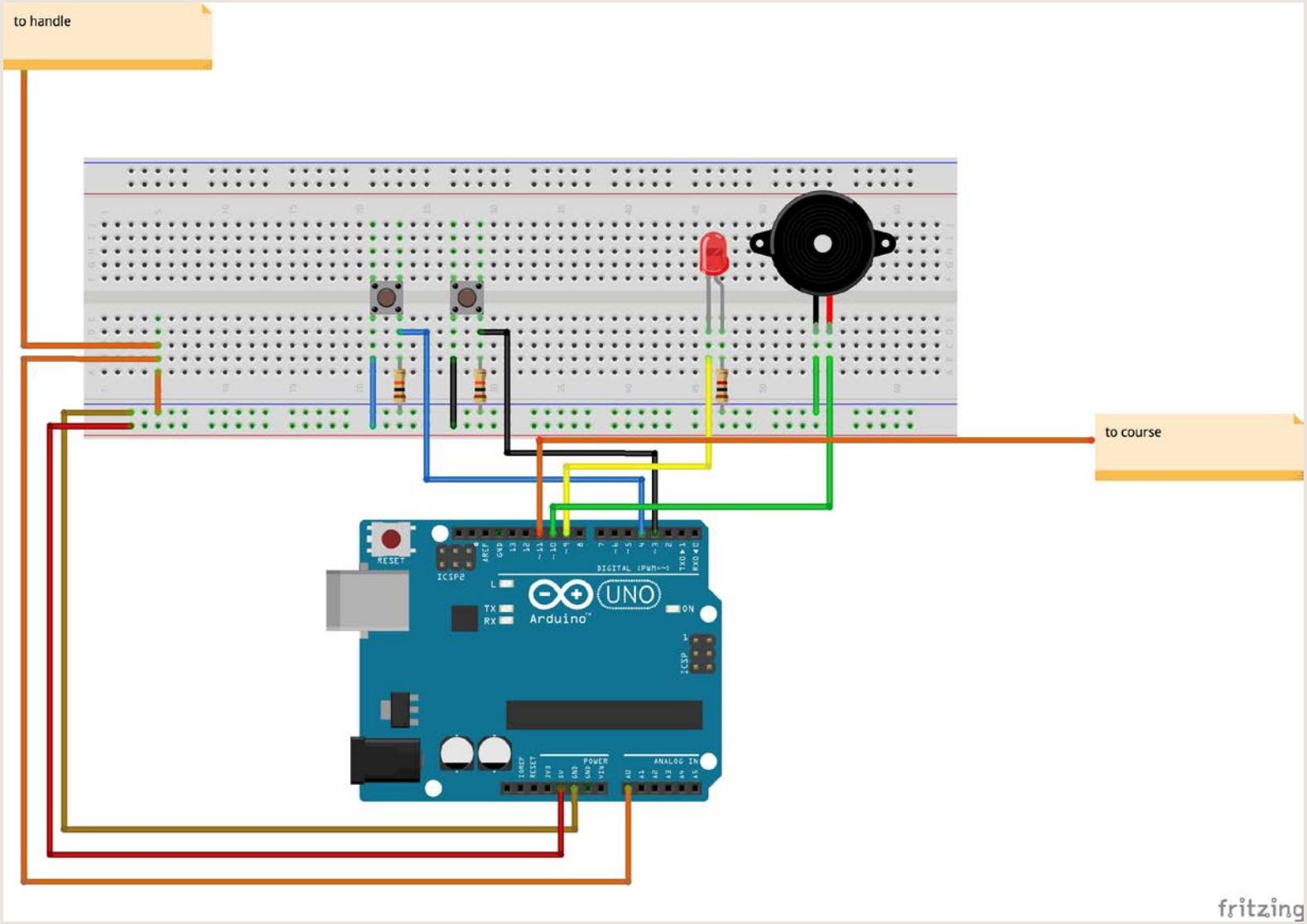
From tone(speakerPin, 440) to digitalWrite(speakerPin, HIGH).

pinMode of speakerPin and ledPin changed to OUTPUT.

PROCESS (CODING - ARDUINO) v5

```
int readingPin = A0;   // to test if the circuit is closed
int circuitPin = 11;
int speakerPin = 10;
int ledPin = 9;
int startPin = 4;
int endPin = 3;
int reading = analogRead(readingPin);

void setup() {
      Serial.begin(9600);
      pinMode(circuitPin, OUTPUT);    // setup circuit
      pinMode(speakerPin, OUTPUT);    // setup buzzer
      pinMode(ledPin, OUTPUT);        // setup led
      pinMode(startPin, INPUT);       // setup start button
      pinMode(endPin, INPUT);         // setup end button
      // 01: welcome screen
      // 02: timer start
      // 03: timer end
  }

void loop() {
   if(reading > 10) {                   // if loop wand touches the wire
      digitalWrite(ledPin, HIGH);       // led lights up
      digitalWrite(speakerPin, HIGH);   // buzzer buzzes
} else {
      digitalWrite(ledPin, LOW);
      digitalWrite(speakerPin, LOW);
}

      if(startPin == 1) {          // when start pin is pressed
        Serial.write(1);           // send 1 to processing to play the video (countup timer)
      } else {
        if(endPin == 1) {          // when end pin is pressed
          Serial.write(2);         // send 2 to processing to stop the video (stop timer, play end video)
          digitalWrite(ledPin, HIGH);
          digitalWrite(speakerPin, HIGH);
      } else {
          Serial.write(0);         // continue playing welcome video on loop
      }
   } }
```

PROCESS (CODING - ARDUINO) v5



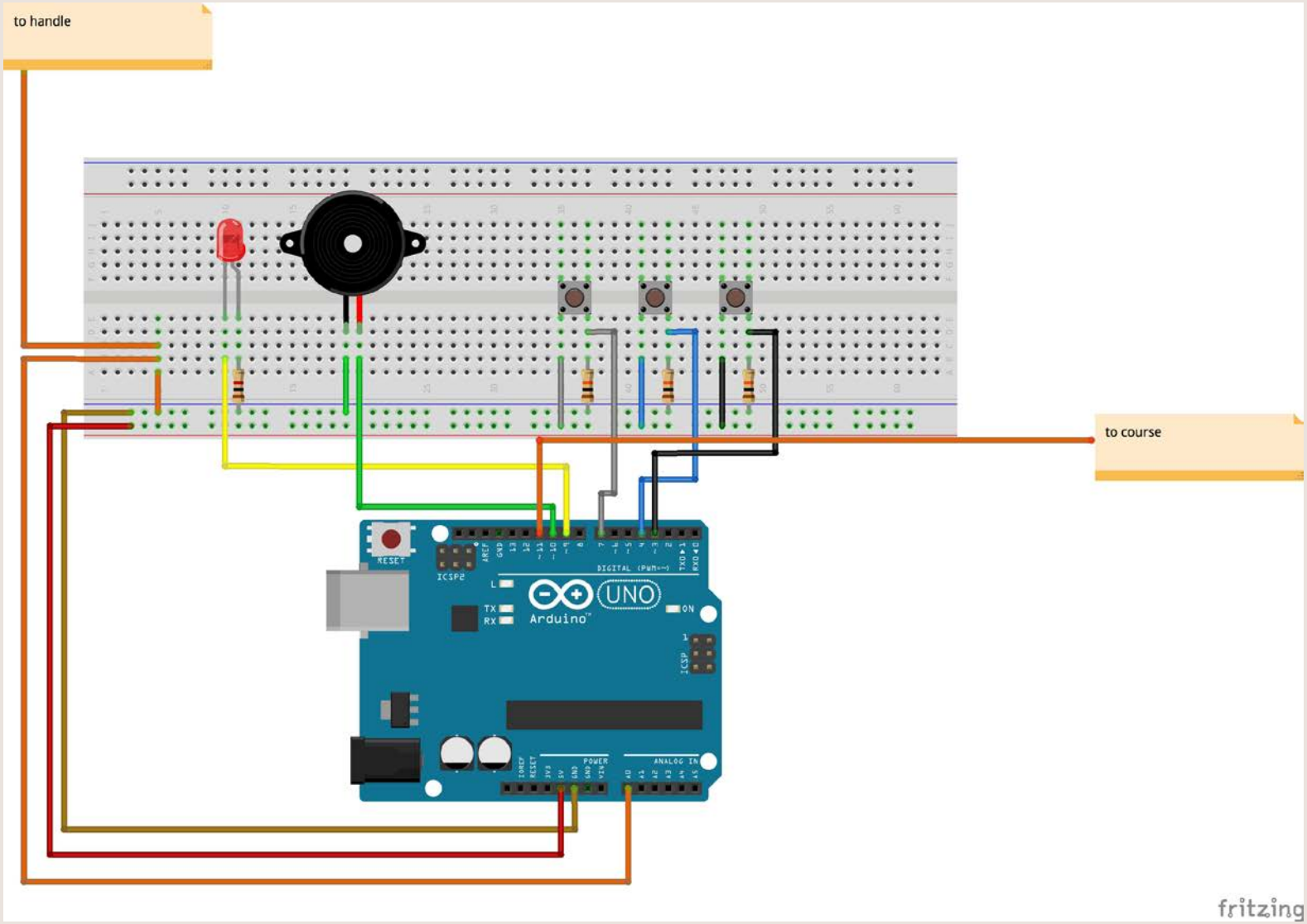
Two buttons are added – one to start the timer and one to stop the timer.
When the button is pressed, it sends either “1” or “2” to processing to play or stop the video.

PROCESS (CODING - ARDUINO) v6 FINAL

```
int readingPin = A0;   // to test if the circuit is closed
int circuitPin = 11;
int speakerPin = 10;
int ledPin = 9;
int startPin = 7;
int endPin = 3;
int prizePin = 4;
int reading = analogRead(readingPin);

void setup() {
      Serial.begin(9600);
      pinMode(circuitPin, OUTPUT);     // setup circuit
      pinMode(speakerPin, OUTPUT);     // setup buzzer
      pinMode(ledPin, OUTPUT);         // setup led
      pinMode(startPin, INPUT);        // setup start button
      pinMode(endPin, INPUT);          // setup end button
      pinMode(prizePin, INPUT);        // setup see results button
      // 01: welcome screen
      // 02: timer start
      // 03: timer end
  }

void loop() {
   if (reading < 10) {                 // if loop wand touches the wire
      digitalWrite(ledPin, HIGH);      // led lights up
      digitalWrite(speakerPin, HIGH);  // buzzer buzzes
   } else {
      digitalWrite(ledPin, LOW);
      digitalWrite(speakerPin, LOW);
}

      if (digitalRead(startPin == 1)) {  // when start pin is pressed
          delay(10);                     // for debouncing
          Serial.write(1);               // send 1 to processing to play the video (countup timer)
}
      if (digitalRead(endPin == 1)) {    // when end pin is pressed
          delay(10);                     // for debouncing
          Serial.write(2);               // send 2 to processing to stop the video (stop timer, play end video)
}
      if (digitalRead(prizePin == 1)) {  // when see results pin is pressed
          delay(10);                     // for debouncing
          Serial.write(3);               // continue playing welcome video on loop
} }
```

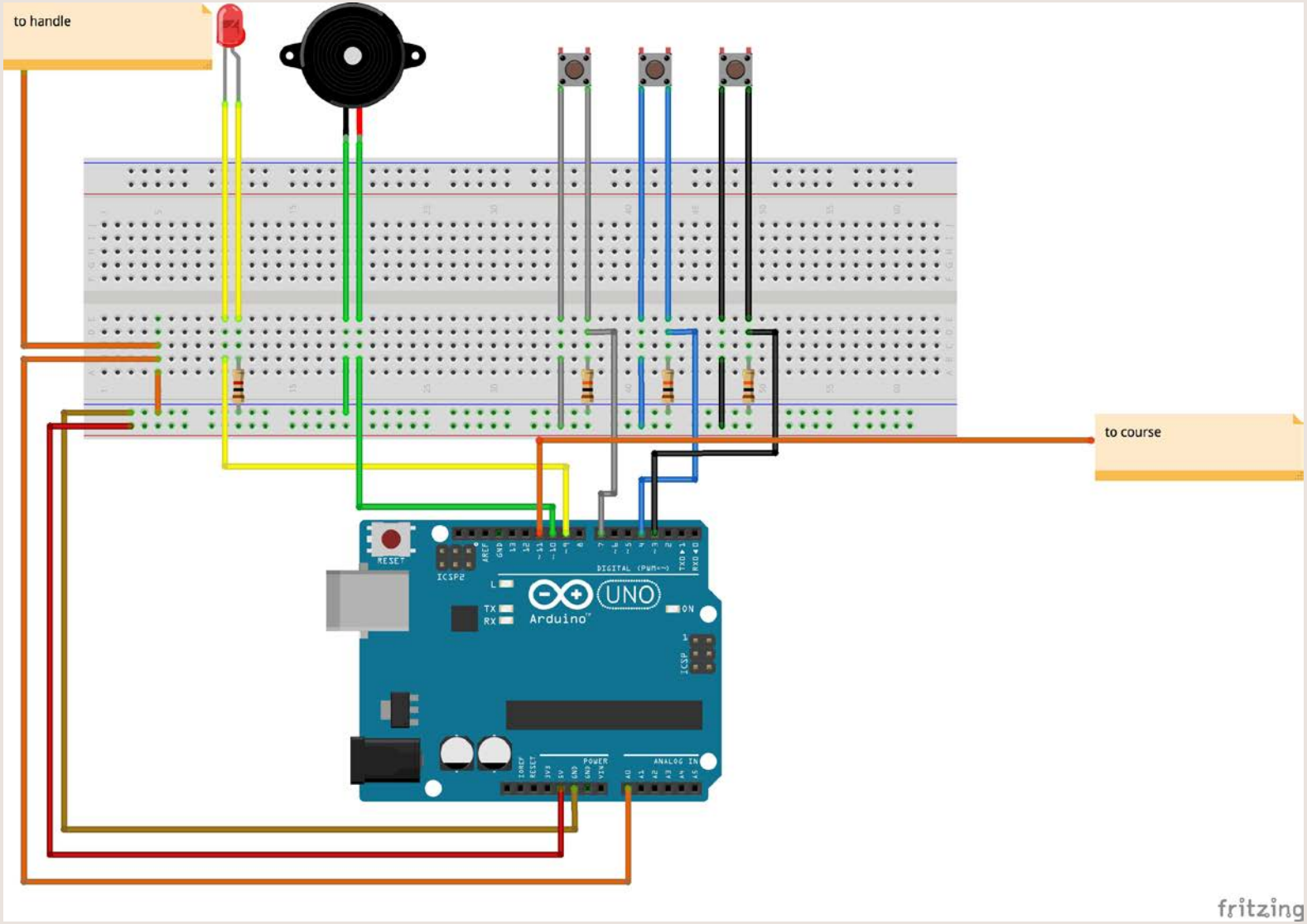
Removed the Serial.write(0) line as there is no need the welcome video is the default video on loop in Processing

PROCESS (CODING - ARDUINO) v6 FINAL



One more button was added – “see results”
When the button is pressed, it sends “3” to processing to play the ending video.

PROCESS (CODING - ARDUINO) v6 FINAL



After making sure that everything works, we added male to female wires to extend the connection of the buttons, LED and piezo buzzer to outside the box.

PROCESS (CODING - PROCESSING) e.g.

EXAMPLE TO PLAY A VIDEO FROM ARDUINO. (FROM LEI)

```
import processing.serial.*;
import processing.video.*;

Serial myPort;    // Create object from Serial class
int val;          // Data received from the serial port

Movie movie;

void setup() {
  size(640, 360);
  background(0);

  String portName = Serial.list()[2];
  print(portName);

  myPort = new Serial(this, portName, 9600);

  // Load and play the video in a loop
  movie = new Movie(this, "transit.mov");
}

void movieEvent(Movie m) {
  m.read();
}

void draw() {

  if ( myPort.available() > 0) {   // If data is available,
    val = myPort.read();           // read it and store it in val
    println(val);
  }

  //if (movie.available() == true) {
  //  movie.read();
  //}

  if (val == 1) {
    movie.loop();
    image(movie, 0, 0, width, height);
  } else {
    movie.pause();
  }
}
```

PROCESS (CODING - PROCESSING) v1 (part 1)

problems noted and changes made are in this colour

```
import processing.serial.*;
import processing.video.*;

String PATH1 = "/Users/ashleytan/Documents/adm/sem 1.2/experimental interaction/final proj/welcome.
mp4";
String PATH2 = "/Users/ashleytan/Documents/adm/sem 1.2/experimental interaction/final proj/stopwatch.
mp4";
String  PATH3  =  "/Users/ashleytan/Documents/adm/sem  1.2/experimental  interaction/final  proj/end.
mp4";

// string to summarise the file path to the videos

Serial myPort;      // Create object from Serial class
int val;            // Data received from the serial port

Movie welcome, timer, ending;     // Naming welcome, timer, and ending for the Movie function

void setup() {
  size(640, 360);        // set size for video
  frameRate(30);         // set frame rate for video
  background(0);

  String portName = Serial.list()[2];
  print(portName);

  myPort = new Serial(this, portName, 9600);

  // Load and play the video in a loop
  welcome = new Movie(this, PATH1);    // link the file to the Movie function
  timer = new Movie(this, PATH2);      // link the file to the Movie function
  ending = new Movie(this, PATH3);     // link the file to the Movie function
}

void movieEvent(Movie m) {
  m.read();
}

void draw() {

  if ( myPort.available() > 0) {  // If data is available,
    val = myPort.read();          // read it and store it in val
    println(val);
  }

  //if (movie.available() == true) {
  //  movie.read();
  //}
```

PROCESS (CODING - PROCESSING) v1 (part 2)

problems noted and changes made are in this colour

```
if (val == 1) {                                  // when start button is pressed
     timer.play();                               // timer video is played
     image(timer, 0, 0, width, height);
    } else {
      welcome.loop;                  // play the welcome/instructions video on a loop if no button is
pressed
      image (welcome, 0, 0, width, height);
  }


if (val == 2) {
      timer.pause();
      ending.play();                          // play the ending video when the timer video pauses
      image(ending, 0, 0, width, height);
// the timer video does not pause because this function is not linked to the first if function
// this if function must be within the "if (val == 1)" function in order for the video to pause
  }

}
```

PROCESS (CODING - PROCESSING) v2 (part 1)

problems noted and changes made are in this colour

```
import processing.serial.*;
import processing.video.*;

String PATH1 = "/Users/ashleytan/Documents/adm/sem 1.2/experimental interaction/final proj/welcome.
mp4";
String PATH2 = "/Users/ashleytan/Documents/adm/sem 1.2/experimental interaction/final proj/stopwatch.
mp4";
String  PATH3  =  "/Users/ashleytan/Documents/adm/sem  1.2/experimental  interaction/final  proj/end.
mp4";

// string to summarise the file path to the videos

Serial myPort;      // Create object from Serial class
int val;      // Data received from the serial port

Movie welcome, timer, ending;     // Naming welcome, timer, and ending for the Movie function

void setup() {
  size(640, 360);       // set size for video
  frameRate(30);        // set frame rate for video
  background(0);

  String portName = Serial.list()[2];
  print(portName);

  myPort = new Serial(this, portName, 9600);

  // Load and play the video in a loop
  welcome = new Movie(this, PATH1);    // link the file to the Movie function
  timer = new Movie(this, PATH2);      // link the file to the Movie function
  ending = new Movie(this, PATH3);          // link the file to the Movie function
}

void movieEvent(Movie m) {
  m.read();
}

void draw() {

  if ( myPort.available() > 0) {       // If data is available,
    val = myPort.read();          // read it and store it in val
    println(val);
  }

  //if (movie.available() == true) {
  //  movie.read();
  //}
```

PROCESS (CODING - PROCESSING) v2 (part 2)

problems noted and changes made are in this colour

```
if (val == 1) {                                  // when start button is pressed
    timer.play();                                // timer video is played
    image(timer, 0, 0, width, height);
} else if (val == 2) {
     timer.pause();
// neater format, but the pause is still not working
     ending.play();                         // play the ending video when the timer video pauses
     image(ending, 0, 0, width, height);
} else {
       welcome.loop;                        // play the welcome/instructions video on a loop if no button
is pressed
       image (welcome, 0, 0, width, height);
  }

}
```

PROCESS (CODING - PROCESSING) v3 (part 1)

problems noted and changes made are in this colour

```
import processing.serial.*;
import processing.video.*;

String PATH1 = "/Users/ashleytan/Documents/adm/sem 1.2/experimental interaction/final proj/welcome.
mp4";
String PATH2 = "/Users/ashleytan/Documents/adm/sem 1.2/experimental interaction/final proj/stopwatch.
mp4";
String  PATH3  =  "/Users/ashleytan/Documents/adm/sem  1.2/experimental  interaction/final  proj/end.
mp4";

// string to summarise the file path to the videos

Serial myPort;      // Create object from Serial class
int val;      // Data received from the serial port

Movie welcome, timer, ending;     // Naming welcome, timer, and ending for the Movie function

void setup() {
  size(640, 360);       // set size for video
  frameRate(30);        // set frame rate for video
  background(0);

  String portName = Serial.list()[2];
  print(portName);

  myPort = new Serial(this, portName, 9600);

  // Load and play the video in a loop
  welcome = new Movie(this, PATH1);    // link the file to the Movie function
  timer = new Movie(this, PATH2);      // link the file to the Movie function
  ending = new Movie(this, PATH3);          // link the file to the Movie function
}

void movieEvent(Movie m) {
  m.read();
}

void draw() {

  if ( myPort.available() > 0) {       // If data is available,
    val = myPort.read();          // read it and store it in val
    println(val);
  }

  //if (movie.available() == true) {
  //  movie.read();
  //}
```

PROCESS (CODING - PROCESSING) v3 (part 2)

problems noted and changes made are in this colour

```
if (val == 1) {                                  // when start button is pressed
    timer.play();                                // timer video is played
    image(timer, 0, 0, width, height);
} else {
    welcome.loop;                                // play the welcome/instructions video on a loop if no
button is pressed
    image (welcome, 0, 0, width, height);
  }

float md = timer.duration();                     // float is to store numbers with decimal point
float mt = timer.time();                         // float is to store numbers with decimal point

if (mt < md && val == 2) {                       // if the timer video is playing and the stop button is
pressed
    timer.pause();
// pause the video for 5 seconds if not the video will stop immediately, and the participant will
not be able to see their timings
    delay(5000);
// was not able to use the delay function because it will delay the next video
    ending.play();                          // play the ending video when the timer video pauses
    image(ending, 0, 0, width, height);
    }

}
```

PROCESS (CODING - PROCESSING) FINAL (part 1)

problems noted and changes made are in this colour

```
import processing.serial.*;
import processing.video.*;

String PATH1 = "/Users/ashleytan/Documents/adm/sem 1.2/experimental interaction/final proj/welcome.mp4";
String PATH2 = "/Users/ashleytan/Documents/adm/sem 1.2/experimental interaction/final proj/stopwatch.mp4";
String PATH3 = "/Users/ashleytan/Documents/adm/sem 1.2/experimental interaction/final proj/end.mp4";

// string to summarise the file path to the videos

Serial myPort;       // Create object from Serial class
int val;        // Data received from the serial port

Movie welcome, timer, ending;   // Naming welcome, timer, and ending for the Movie function

int tt;              // integer for total time taken by participant, for millis() function later
int wait = 5000;     // for millis() function later

void setup() {
  size(640, 360);          // set size for video
  frameRate(30);           // set frame rate for video
  background(0);

  String portName = Serial.list()[2];
  print(portName);

  myPort = new Serial(this, portName, 9600);

  // Load and play the video in a loop
  welcome = new Movie(this, PATH1);  // link the file to the Movie function
  timer = new Movie(this, PATH2);          // link the file to the Movie function
  ending = new Movie(this, PATH3);         // link the file to the Movie function
}

void movieEvent(Movie m) {
  m.read();
}

void draw() {

  if ( myPort.available() > 0) {     // If data is available,
    val = myPort.read();             // read it and store it in val
    println(val);
  }

  //if (movie.available() == true) {
  //  movie.read();
  //}
```

PROCESS (CODING - PROCESSING) FINAL (part 2)

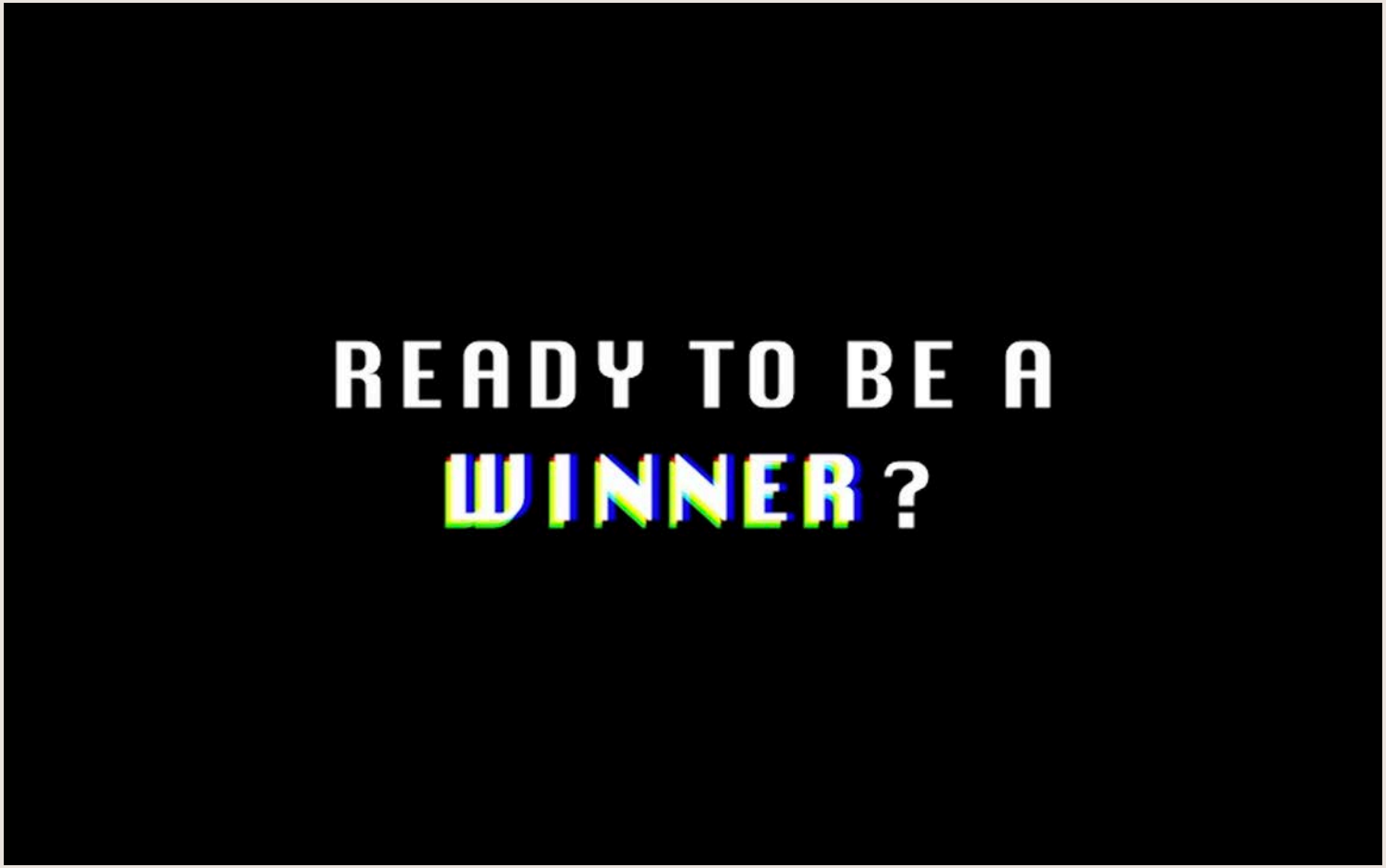
problems noted and changes made are in this colour

```
if (val == 1) {                              // when start button is pressed
    timer.play();                                 // timer video is played
    image(timer, 0, 0, width, height);
    tt = millis();
// current time that the programme has run for when the start button is pressed
} else {
    welcome.loop;                                 // play the welcome/instructions video on a loop if no button
is pressed
    image (welcome, 0, 0, width, height);
  }

float md = timer.duration();                      // float is to store numbers with decimal point
float mt = timer.time();                          // float is to store numbers with decimal point

if (mt < md && val == 2) {                   // if the timer video is playing and the stop button is pressed
    timer.speed(0.01);                            // to slow down the video as if it is paused
    timer.volume(0);                         // no volume because when the video slows down the sound is still
emitted
    image(timer, 0, 0, width, height);
    if (millis() - tt >= wait) {
// to delay 5 seconds (delay function cannot be used because it will delay the following process)
    timer.pause();
    tt = millis();
    }
}

// had to add another button in the end because playing two videos consecutively while delaying the first
video (so that the participant can see the timing) will cause the second video to lag
if (mt < md && val == 3) {                   // when the timer played + when show results button is pressed
    ending.play();                           // play ending video
    image(ending, 0, 0, width, height);
  }

}
```

PROCESS (VIDEO - WELCOME)

The videos were all made in Adobe After Effects CC.

The glitch in the first video was from a tutorial on Youtube (ref below). The glitch is added on to “winner” to as a teaser/hint to what will happen in the end.

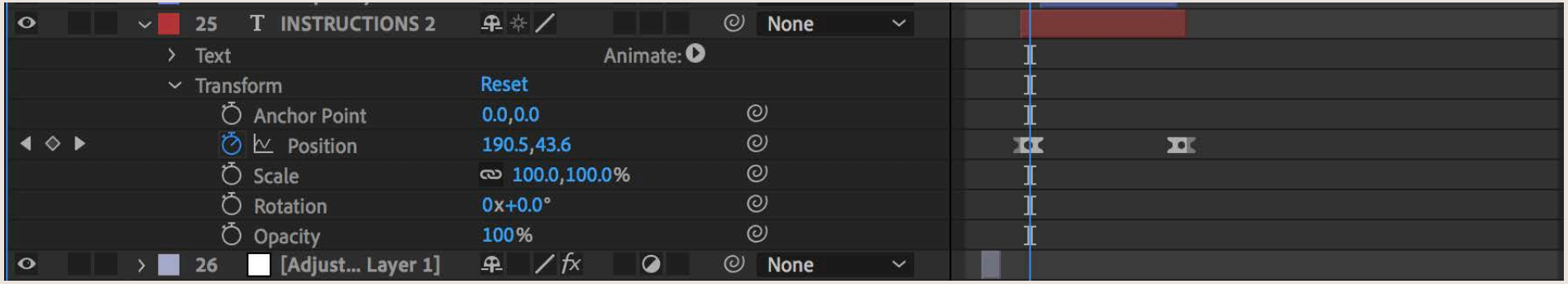


PROCESS (VIDEO - TRANSITIONS)

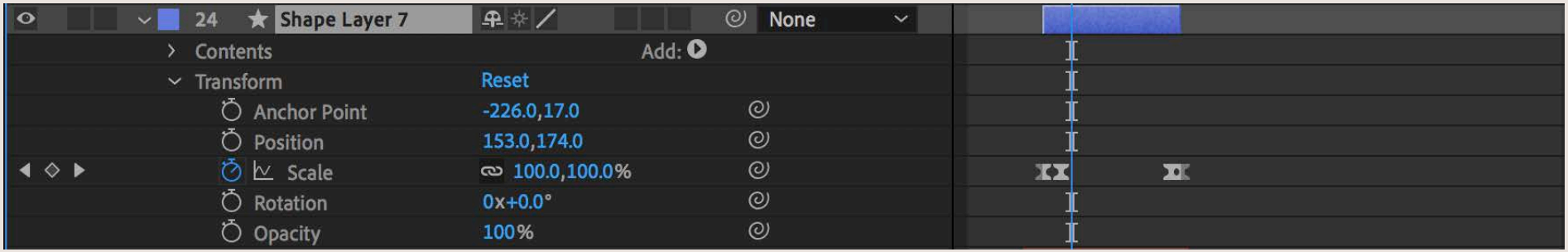
There are two types of transitions used in all the videos – moving in or out of the frame and scaling up or down the sizes to enter or exit the frames.

Before we applied the transitions, the final positions of the objects are set in place. Then, we used key frames to adjust the positions and size of the objects at a particular timing.

For example, if the scaling up transition is used to bring the object into frame, a key frame will be added to the first frame of the object with the scale set to 0%. Then, depending on how fast or slow we want the object to come into frame, we go forward a few frames (e.g. one sec) and set it to 100%. Between the first key frame and the second key frame, the object will scale up from 0% to 100% in one second. Then, in order for the animation to be smooth, we added in keyframe assistant > easy ease to the key frames.



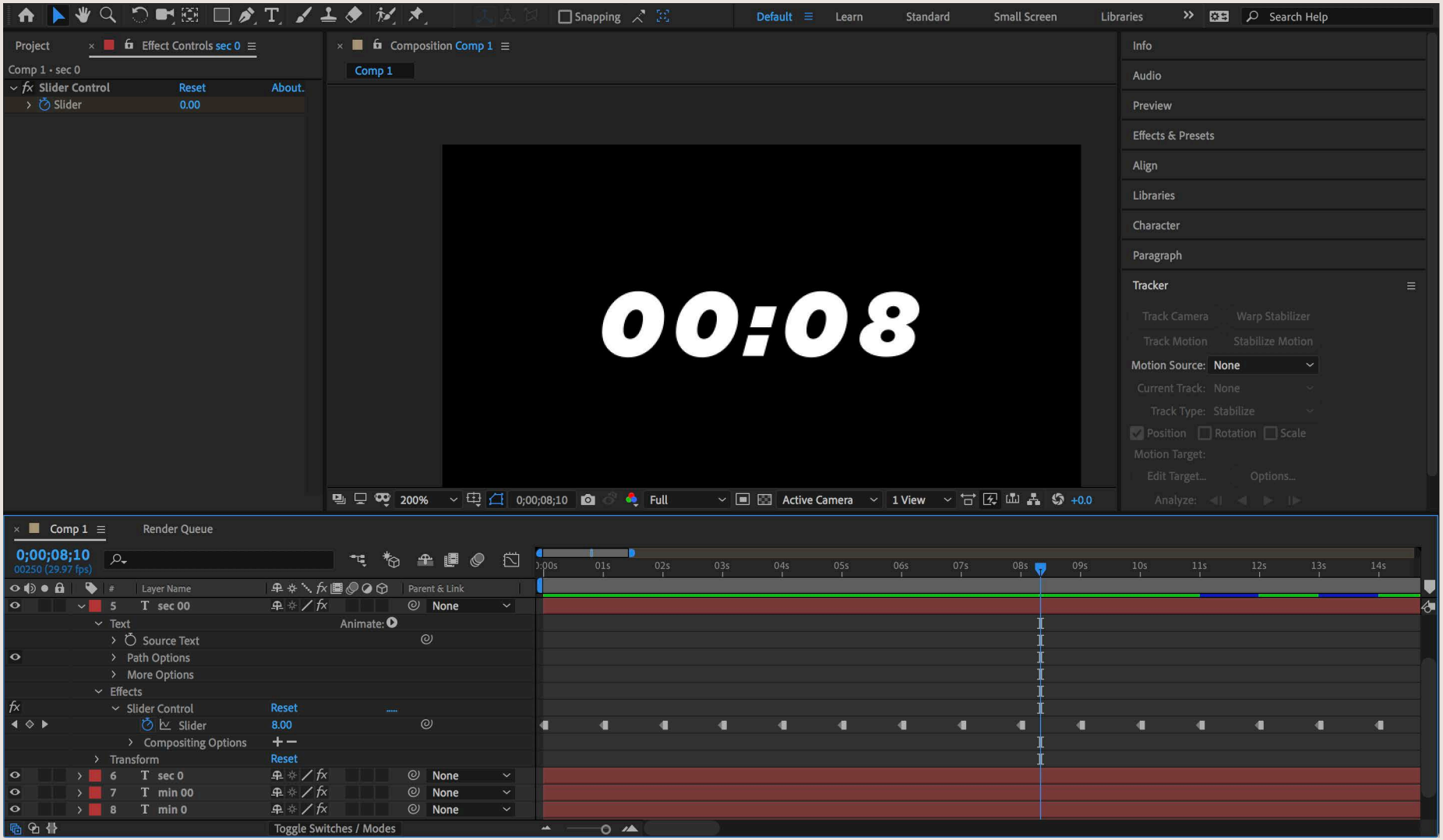
e.g. moving the object in and out of the frame.



e.g. scaling the object up and down.

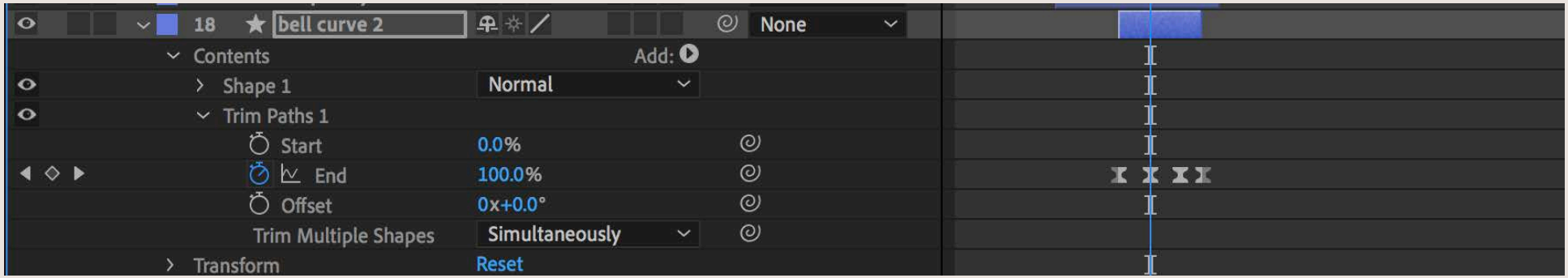
PROCESS (VIDEO - STOPWATCH)

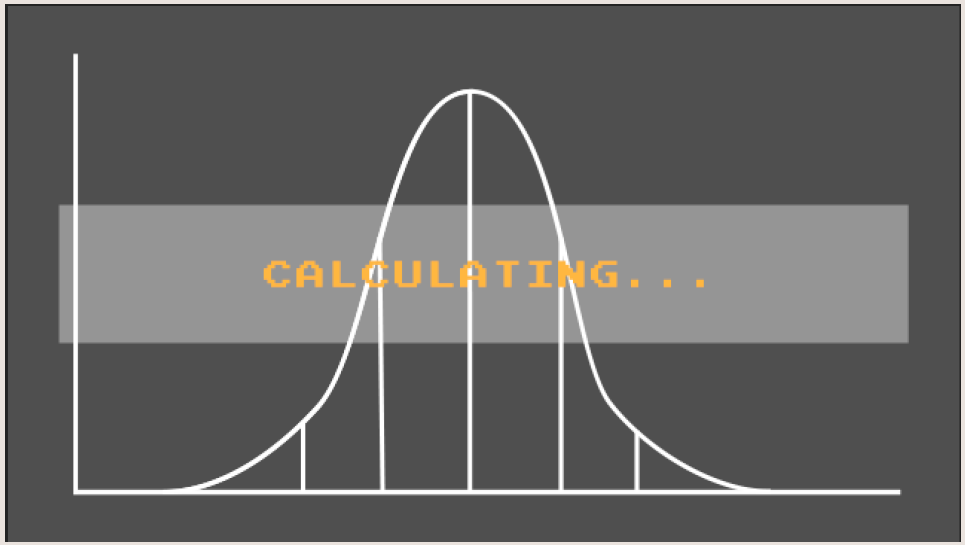
Similar to the transitions used earlier on, we used key frames to change the number as each second passes. To do that, we first have to add a slider control to the object's effects control panel. Then, create a number for every digit for "00:00". For every second or minute that passes, add one to the value of the slider. Then, add a keyframe to animate it.

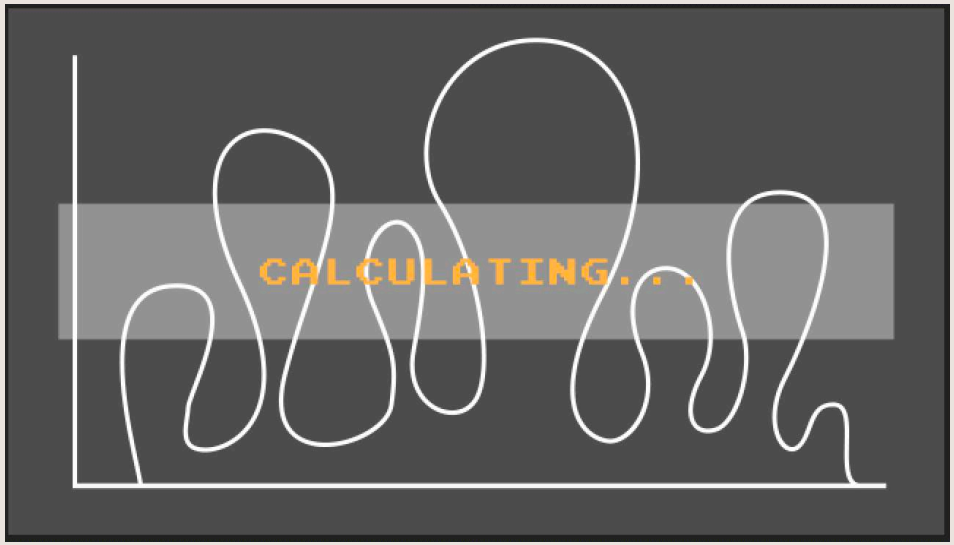


PROCESS (VIDEO - GRAPH)

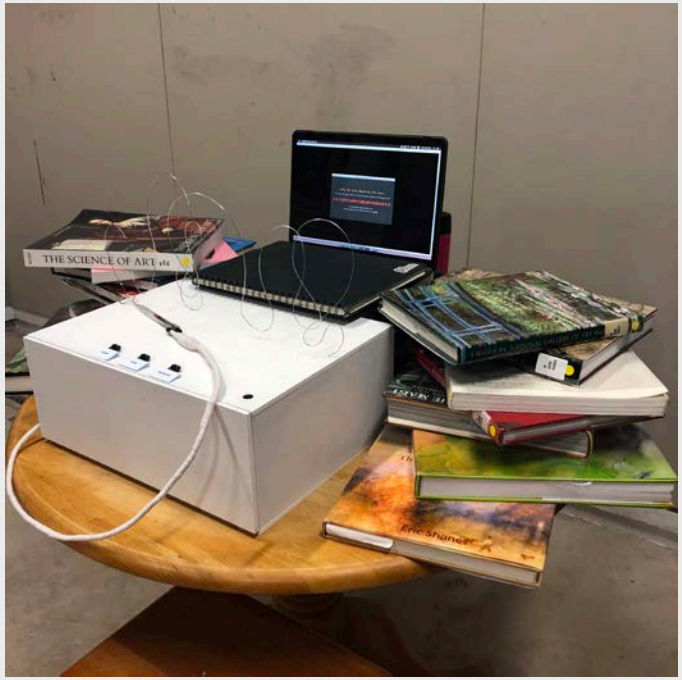
For the graph, we used trim path to animate the path of the lines. Using key krames, we set the value of the Start and End value accordingly to animate the lines.







FINAL IN-CLASS SET-UP



INSIGHTS AND REFLECTION

I remembered during the first lesson Lei asked us what we expect to learn from Experimental Interaction. I remembered that I said I wasn't clear what this module is about, and I'm not sure what to expect. Thinking about what I said, I feel thankful and fulfilled that I became more knowledgable about coding, designing for meaningful interactions, and learning how to design for a seamless flow of interaction.

Through the micro projects, we were introduced to simple coding on Arduino and learning how to design for interaction. In our final project, we had think through the flow of both the participant's interaction and coding that is in Arduino and Processing. The technical aspects of coding and setting up the circuit is like a puzzle to me. To be able to find certain functions in the codes to fit the interaction which we want to convey to our participants is tough, but to be able to solve the puzzle in the end is very fulfilling. For example, we had so many problems on Processing regarding the videos, such as not being able to pause the video for a while, and not being to able to have a smooth transition from one video to the other. Though trial and error and researching (esp on forums, the coding community is extremely helpful!), we were able to find the missing piece to fit our codes. Not only were we able to convey the message through a smooth interaction, we were also able to improve our problem solving skills through critical thinking.

Other than the technical aspects, what I have learnt from this project and course is that every element of the user's interaction is important. Be it the physical aesthetics of the installation, or the flow of interaction, it is important to the user's experience to our installation. To be able to create a meaningful experience for the participants and audience, we can only do it by testing it with real participants other than the creators so that we can know what kind of unexpected scenarios that can come up and what gaps are in the process. We will able to know what elements to remove, and what we can improve on to make the flow of interaction more seamless for the participant/audience.

Through this course, I had a steep learning curve for the technical aspect, but I was also able to learn how to think about certain issues in life and look at it in different perspectives through the insights and comments given by my classmates. I used to think that technlogy only had the function of improving our lives, but now I think that we can make use of the technology we have now to create discussions on certain issues, or simply to just explore and test our theories on human behaviour.