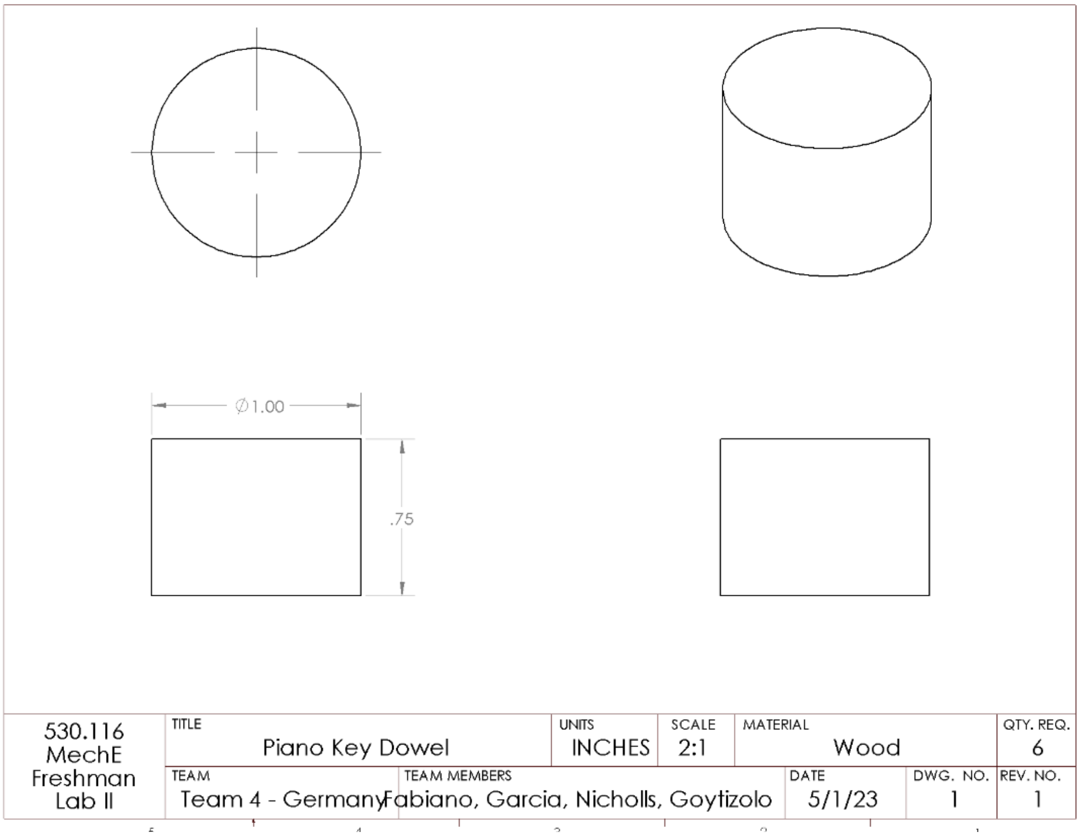
Final Design Report

By Team 4 - Germany (Gaby, Veronica, Lance, Ronald)

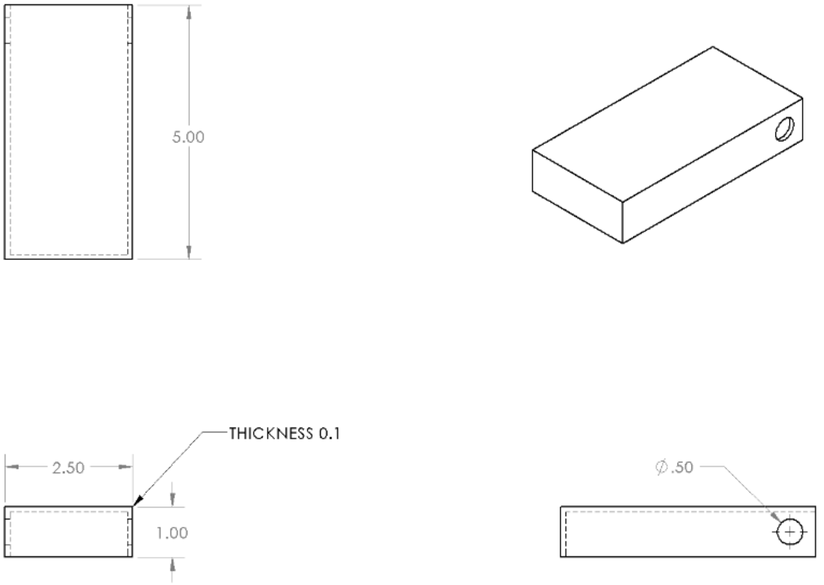
Our project is a piano that resembles the carnival game of "Whack-a-mole," allowing the user to play snippets of work from two of the most famous German pianists: Bach and Beethoven. To attract new players, our ultrasonic sensor looks for people walking by and waves a German flag from the side. Once the user walks in front of the game, the flag stops waving and the user can select from six snippets, each corresponding to one of the six keys on the piano. These snippets are Ode to Joy (Beethoven), Moonlight Sonata (Beethoven), Fur Elise (Beethoven), Symphony No. 5 (Beethoven), Toccata (Bach), and Brandenburg (Bach). Once one is chosen, the LEDs randomly light up, giving the user three seconds to press it. Suppose the user does not press it in time or presses the wrong key, the game resets to the selection phase. But, if the user makes it to the end of the song, the game rewards the user with a flash of lights and a nice winning jingle.

CAD Models

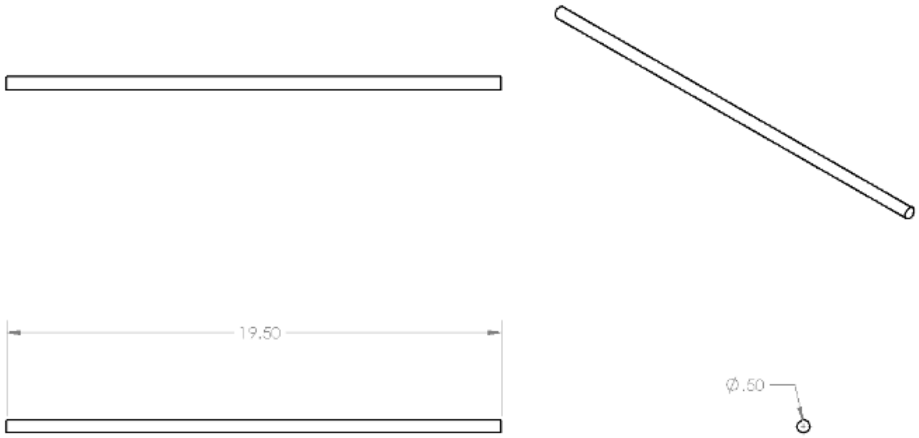
1. Piano keys with the dowel



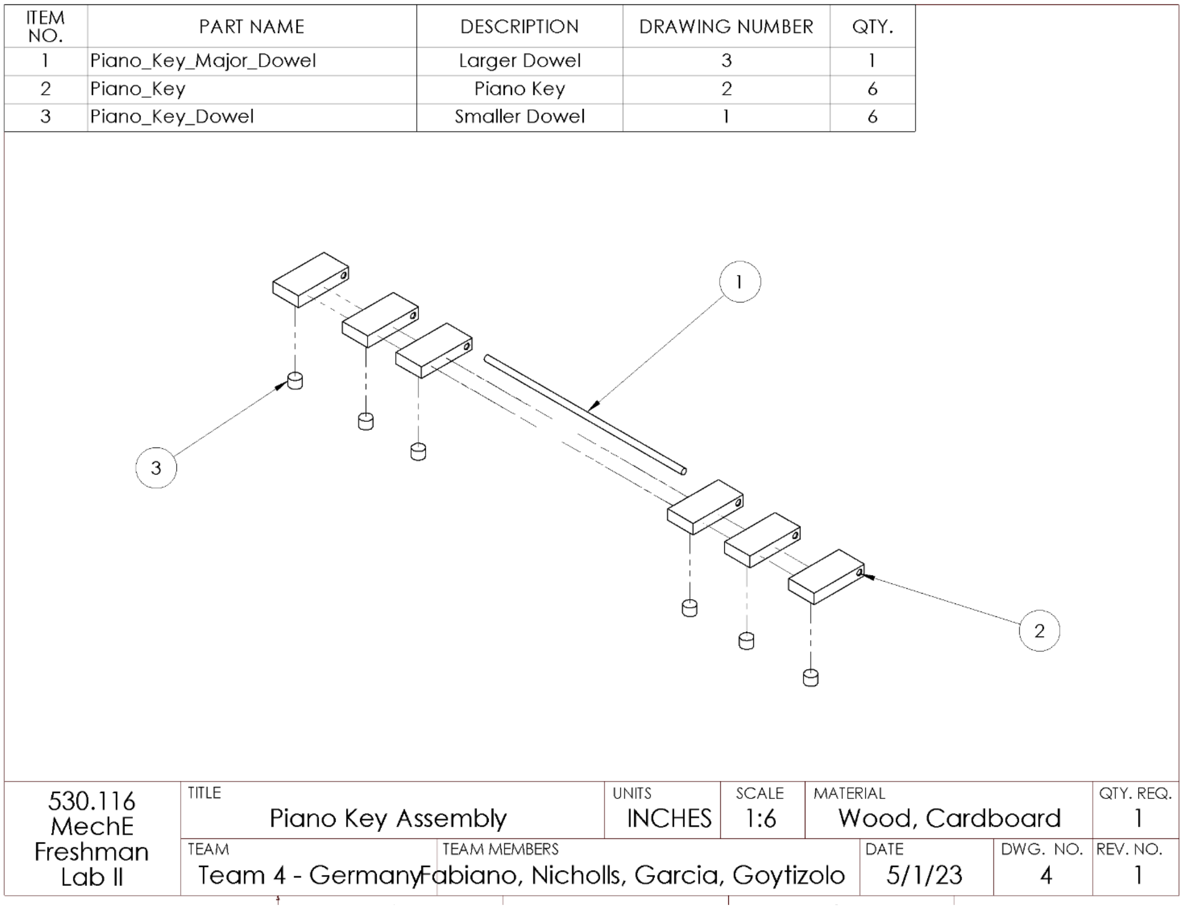
530.116 MechE Freshman Lab II	TITLE Piano Key Dowel	UNITS INCHES	SCALE 2:1	MATERIAL Wood	QTY. REQ. 6
	TEAM Team 4 - Germany	TEAM MEMBERS Fabiano, Garcia, Nicholls, Goytizolo	DATE 5/1/23	DWG. NO. 1	REV. NO. 1



530.116 MechE Freshman Lab II	TITLE Piano Key	UNITS INCHES	SCALE 1:2	MATERIAL Cardboard	QTY. REQ. 6
	TEAM Team 4 - Germany	TEAM MEMBERS Fabiano, Garcia, Nicholls, Goytizolo	DATE 5/1/23	DWG. NO. 2	REV. NO. 1



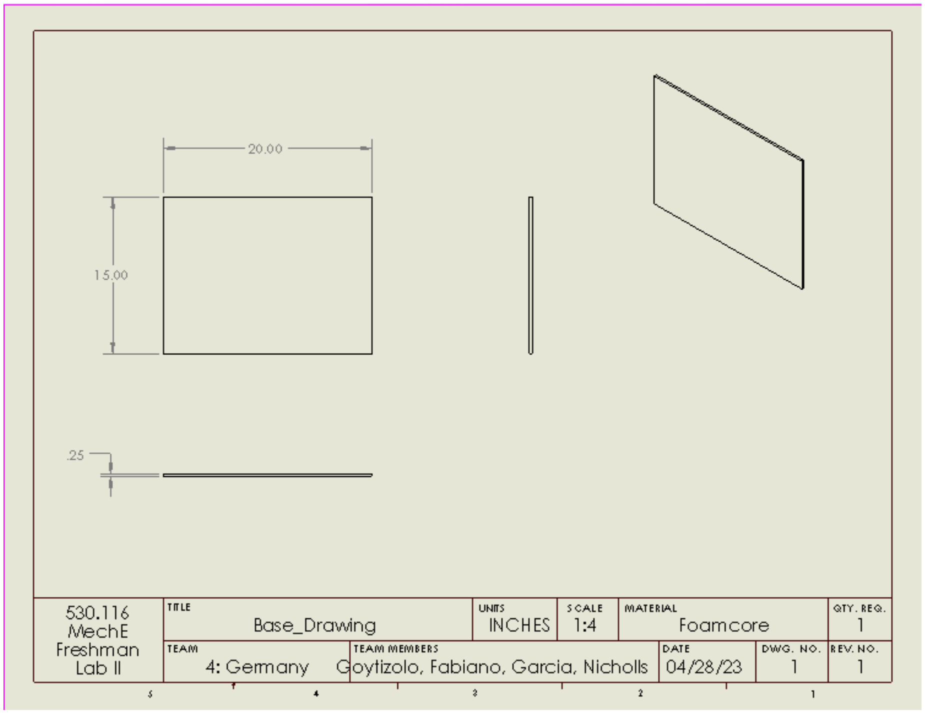
530.116 MechE Freshman Lab II	TITLE Piano Key Major Dowel	UNITS INCHES	SCALE 1:4	MATERIAL Wood	QTY. REQ. 1
	TEAM Team 4 - Germany	TEAM MEMBERS Fabiano, Garcia, Nicholls, Goytizolo	DATE 5/1/23	DWG. NO. 3	REV. NO. 1



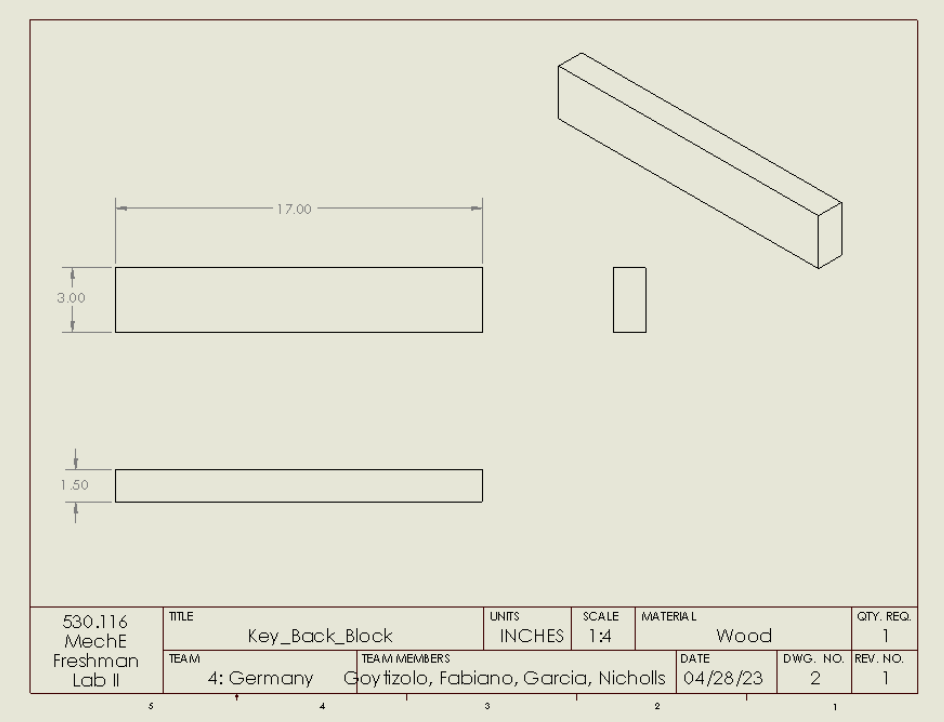
ITEM NO.	PART NAME	DESCRIPTION	DRAWING NUMBER	QTY.
1	Piano_Key_Major_Dowel	Larger Dowel	3	1
2	Piano_Key	Piano Key	2	6
3	Piano_Key_Dowel	Smaller Dowel	1	6

530.116 MechE Freshman Lab II	TITLE Piano Key Assembly	UNITS INCHES	SCALE 1:6	MATERIAL Wood, Cardboard	QTY. REQ. 1
	TEAM Team 4 - Germany	TEAM MEMBERS Fabiano, Nicholls, Garcia, Goytizolo	DATE 5/1/23	DWG. NO. 4	REV. NO. 1

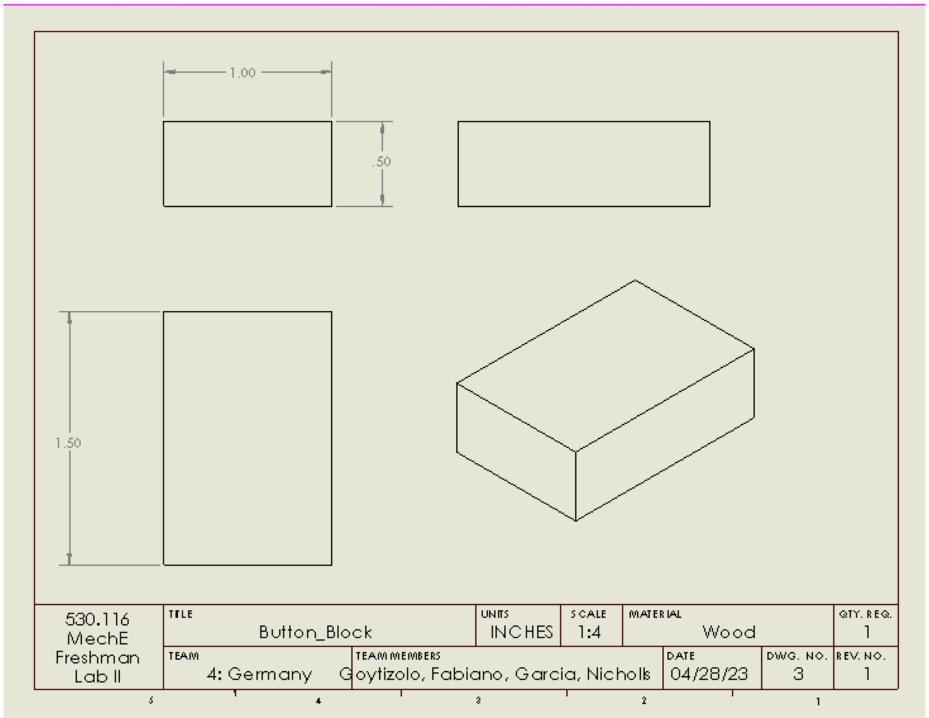
2. Base of Piano



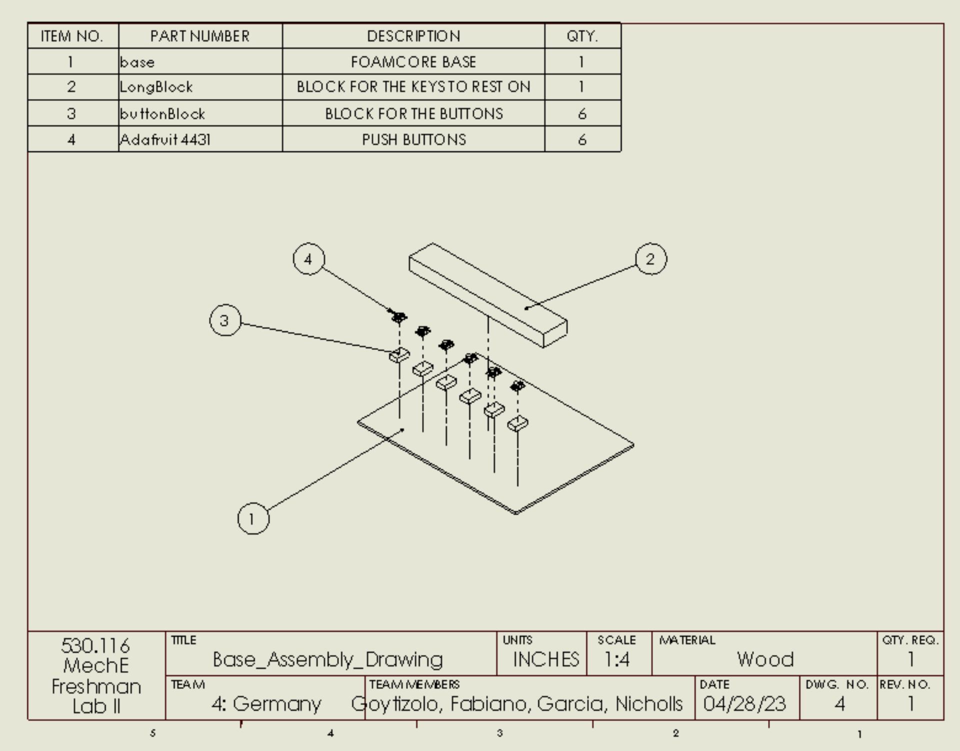
530.116 MechE Freshman Lab II	TITLE Base_Drawing	UNITS INCHES	SCALE 1:4	MATERIAL Foamcore	QTY. REQ. 1
	TEAM 4: Germany	TEAM MEMBERS Goytizolo, Fabiano, Garcia, Nicholls	DATE 04/28/23	DWG. NO. 1	REV. NO. 1



530.116 MechE Freshman Lab II	TITLE Key_Back_Block	UNITS INCHES	SCALE 1:4	MATERIAL Wood	QTY. REQ. 1
	TEAM 4: Germany	TEAM MEMBERS Goytizolo, Fabiano, Garcia, Nicholls	DATE 04/28/23	DWG. NO. 2	REV. NO. 1



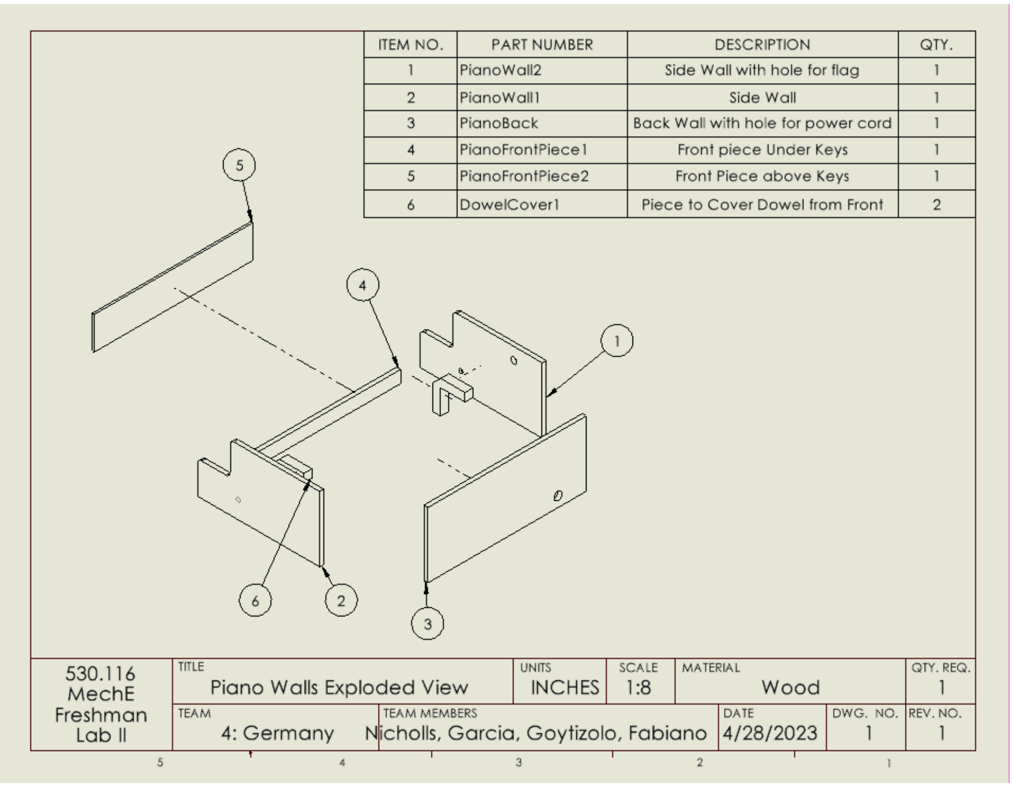
530.116 MechE Freshman Lab II	TITLE Button_Block	UNITS INCHES	SCALE 1:4	MATERIAL Wood	QTY. REQ. 1
	TEAM 4: Germany	TEAM MEMBERS Goytizolo, Fabiano, Garcia, Nicholls	DATE 04/28/23	DWG. NO. 3	REV. NO. 1



ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	base	FOAMCORE BASE	1
2	LongBlock	BLOCK FOR THE KEYS TO REST ON	1
3	buttonBlock	BLOCK FOR THE BUTTONS	6
4	Adafruit 4431	PUSH BUTTONS	6

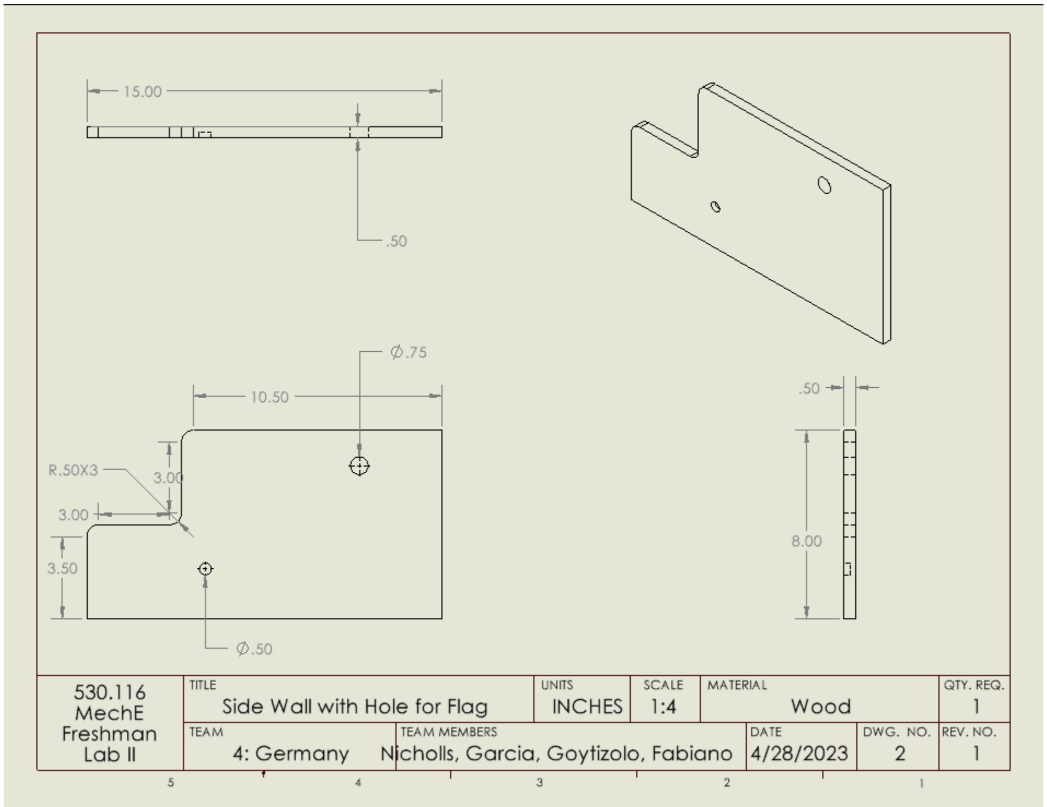
530.116 MechE Freshman Lab II	TITLE Base_Assembly_Drawing	UNITS INCHES	SCALE 1:4	MATERIAL Wood	QTY. REQ. 1
	TEAM 4: Germany	TEAM MEMBERS Goytizolo, Fabiano, Garcia, Nicholls	DATE 04/28/23	DWG. NO. 4	REV. NO. 1

3. The Structure

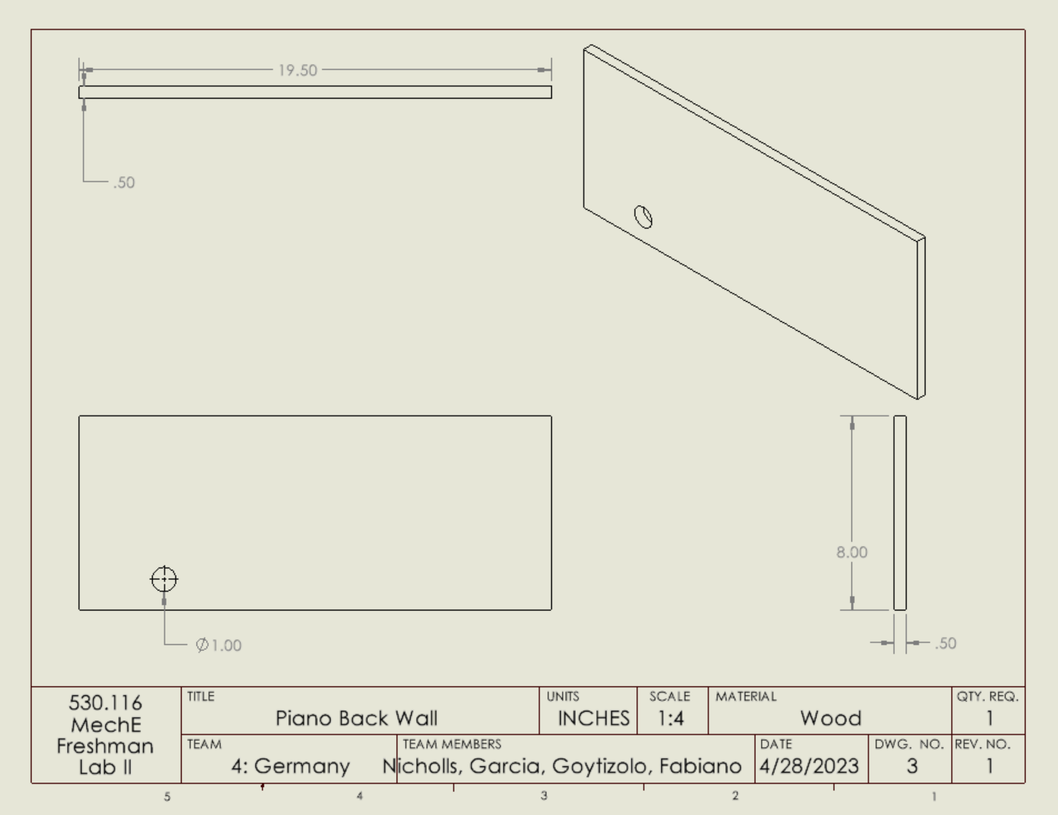


ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	PianoWall2	Side Wall with hole for flag	1
2	PianoWall1	Side Wall	1
3	PianoBack	Back Wall with hole for power cord	1
4	PianoFrontPiece1	Front piece Under Keys	1
5	PianoFrontPiece2	Front Piece above Keys	1
6	DowelCover1	Piece to Cover Dowel from Front	2

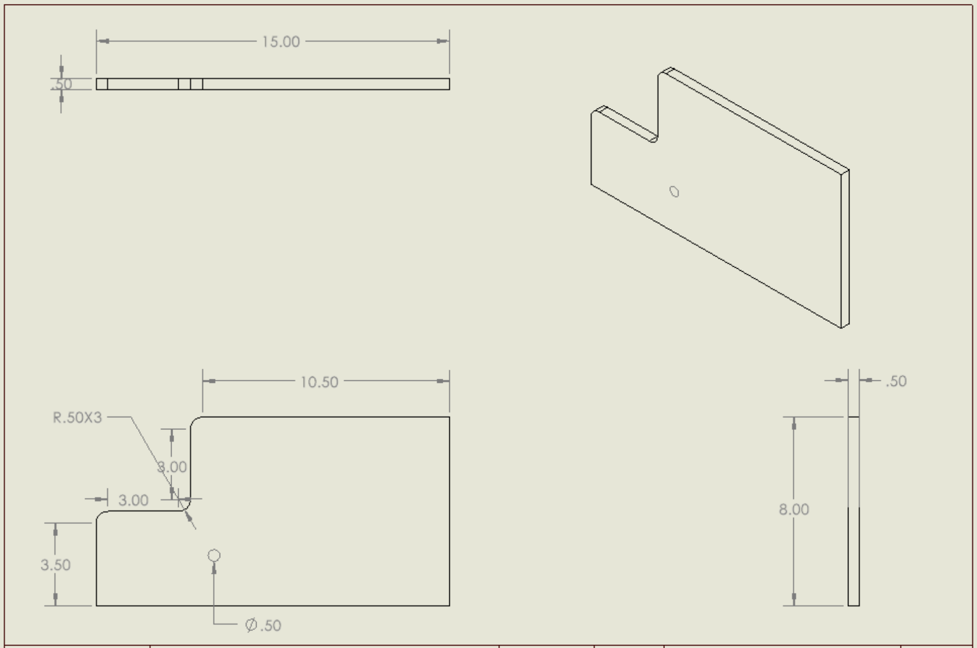
530.116 MechE Freshman Lab II	TITLE Piano Walls Exploded View	UNITS INCHES	SCALE 1:8	MATERIAL Wood	QTY. REQ. 1
	TEAM 4: Germany	TEAM MEMBERS Nicholls, Garcia, Goytizolo, Fabiano	DATE 4/28/2023	DWG. NO. 1	REV. NO. 1



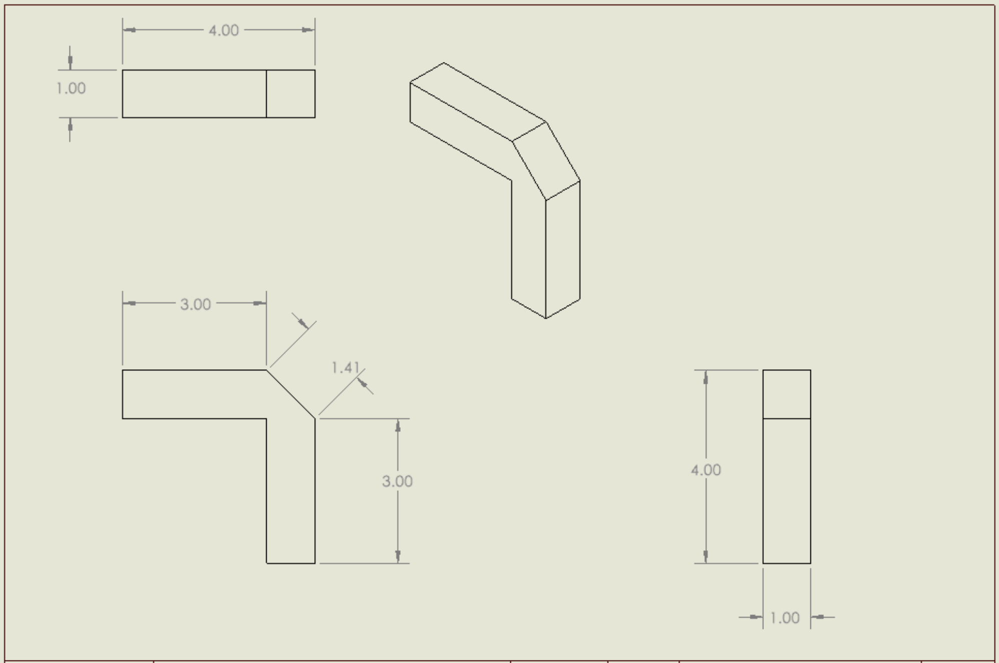
530.116 MechE Freshman Lab II	TITLE Side Wall with Hole for Flag	UNITS INCHES	SCALE 1:4	MATERIAL Wood	QTY. REQ. 1
	TEAM 4: Germany	TEAM MEMBERS Nicholls, Garcia, Goytizolo, Fabiano	DATE 4/28/2023	DWG. NO. 2	REV. NO. 1



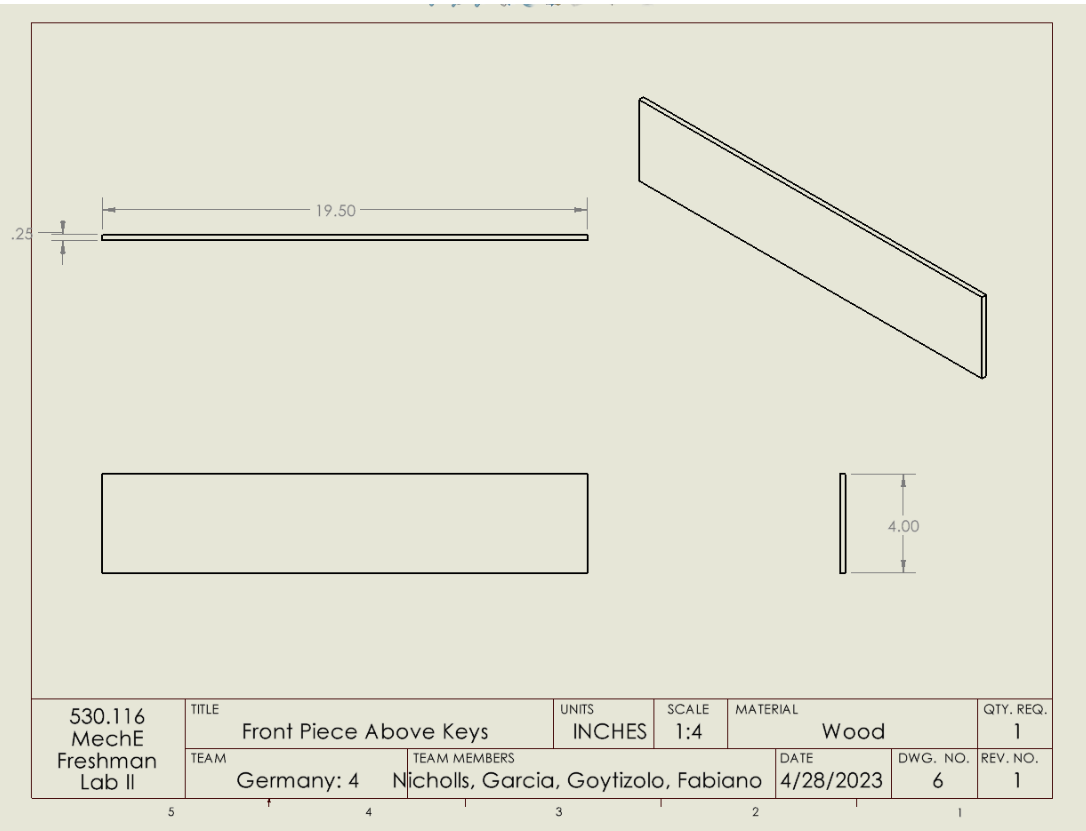
530.116 MechE Freshman Lab II	TITLE Piano Back Wall	UNITS INCHES	SCALE 1:4	MATERIAL Wood	QTY. REQ. 1
	TEAM 4: Germany	TEAM MEMBERS Nicholls, Garcia, Goytizolo, Fabiano	DATE 4/28/2023	DWG. NO. 3	REV. NO. 1



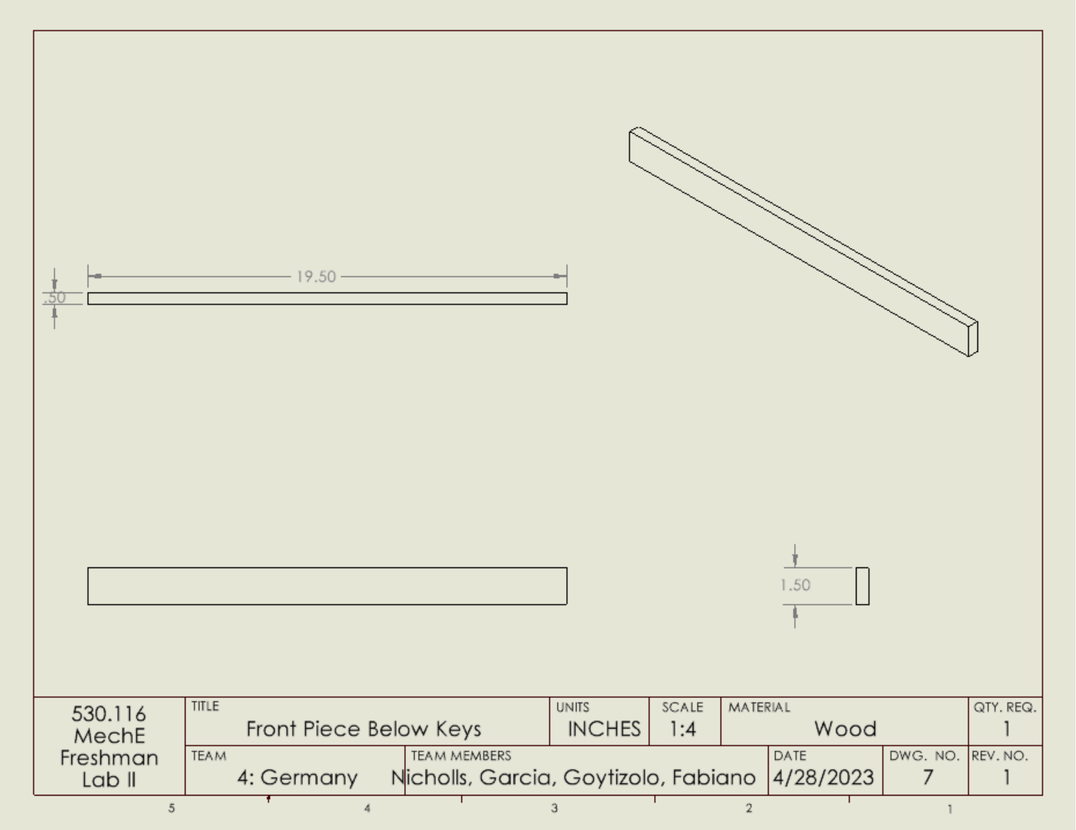
530.116 MechE Freshman Lab II	TITLE Side Wall	UNITS INCHES	SCALE 1:4	MATERIAL Wood	QTY. REQ. 1
	TEAM 4: Germany	TEAM MEMBERS Nicholls, Garcia, Goytizolo, Fabiano	DATE 4/28/2023	DWG. NO. 4	REV. NO. 1



530.116 MechE Freshman Lab II	TITLE Dowel Cover	UNITS INCHES	SCALE 1:2	MATERIAL Wood	QTY. REQ. 1
	TEAM 4: Germany	TEAM MEMBERS Nicholls, Garcia, Goytizolo, Fabiano	DATE 4/28/2023	DWG. NO. 5	REV. NO. 1

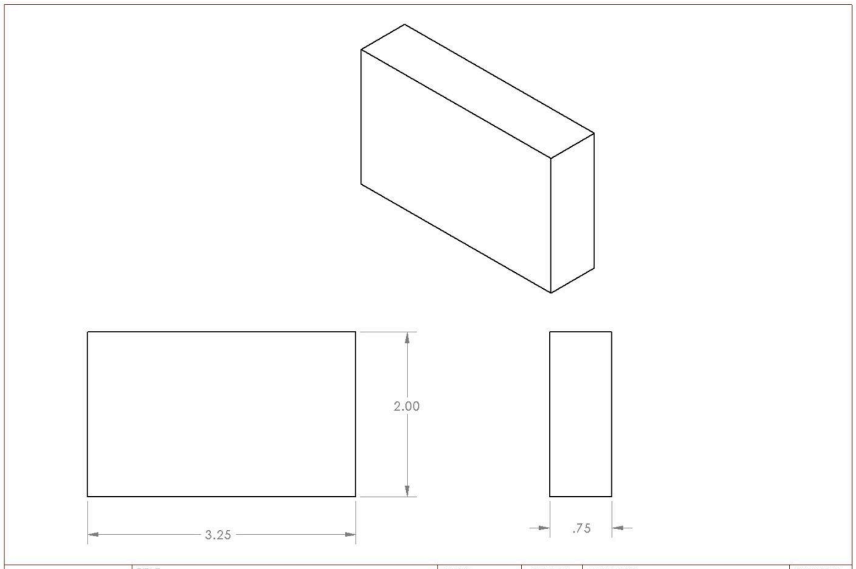


530.116 MechE Freshman Lab II	TITLE Front Piece Above Keys	UNITS INCHES	SCALE 1:4	MATERIAL Wood	QTY. REQ. 1
	TEAM Germany: 4	TEAM MEMBERS Nicholls, Garcia, Goytizolo, Fabiano	DATE 4/28/2023	DWG. NO. 6	REV. NO. 1

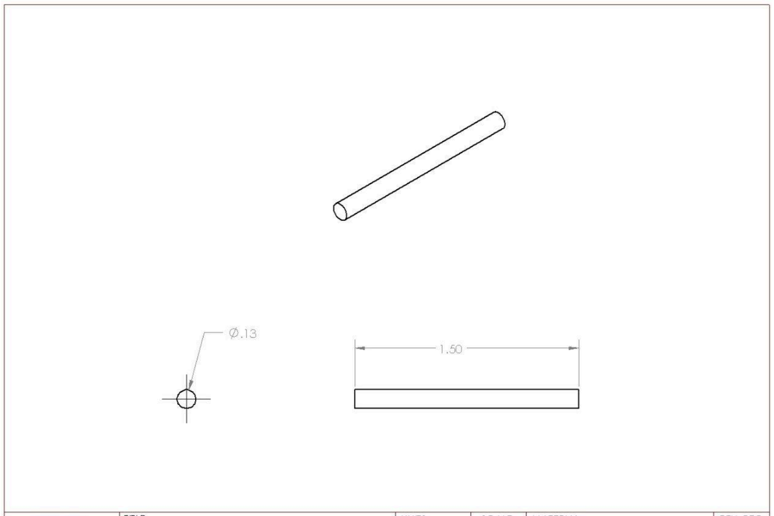


530.116 MechE Freshman Lab II	TITLE Front Piece Below Keys	UNITS INCHES	SCALE 1:4	MATERIAL Wood	QTY. REQ. 1
	TEAM 4: Germany	TEAM MEMBERS Nicholls, Garcia, Goytizolo, Fabiano	DATE 4/28/2023	DWG. NO. 7	REV. NO. 1

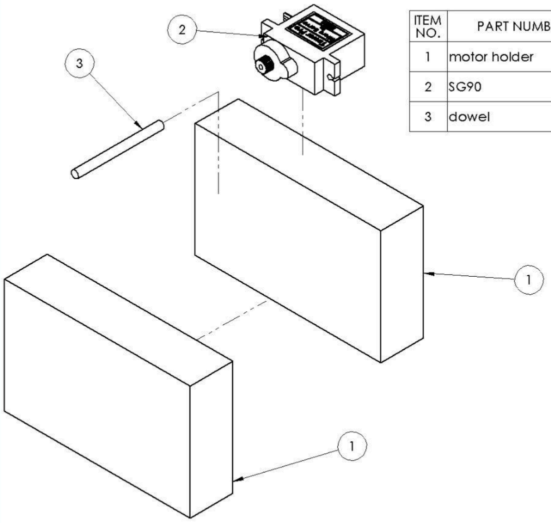
4. The Flag-waving Mechanism



530.116 MechE Freshman Lab II	TITLE flag holder block	UNITS INCHES	SCALE 1:1	MATERIAL wood	QTY. REQ. 2
	TEAM Germany	TEAM MEMBERS Gaby, Veronica, Lance, Ronald	DATE 4/30	DWG. NO. 1	REV. NO. 1



530.116 MechE Freshman Lab II	TITLE dowel	UNITS INCHES	SCALE 1:1	MATERIAL wood	QTY. REQ. 1
	TEAM Germany	TEAM MEMBERS Gaby, Veronica, Ronald, Lance	DATE	DWG. NO. 3	REV. NO. 1



ITEM NO.	PART NUMBER	DESCRIPTION	DRAWING NUMBER	QTY.
1	motor holder	wood block	1	2
2	SG90	servo motor	2	1
3	dowel	for flag	3	1

530.116 MechE Freshman Lab II	TITLE	UNITS INCHES	SCALE 1:1	MATERIAL	QTY. REQ.
	TEAM	TEAM MEMBERS	DATE	DWG. NO.	REV. NO.

5 4 3 2 1

Final Project Calculations

Team 4 - Germany

April 2023

Wood Amount

$$\begin{aligned}\text{Total Area of Wood } &= (19.5 * 1.5 + 2 * 8 * 15 + 20 * 15 + 19.5 * 15 + 19.5 * 4)\text{in}^2 \\ &= 939.75\text{in}^2\end{aligned}$$

Rubber Band Calculations

Given:

1. $k = k_{rb}$
2. l as the distance the rubber band is attached from the center of the dowel
3. h as the radial height of the key (starting from the center of the dowel and going up)
4. w as the long side of the piano key

This implies that the total displacement that the rubber band encompasses is given by:

$$\|2h, l\| = \sqrt{4h^2 + l^2} = \ell$$

And we know:

$$\tau = Fd$$

$$\implies \tau_{rb} = (k\ell)(h)$$

But because we want the piano key to be stable at an angle $\theta = 0$, the torque due to gravity must cancel the torque due to the rubber band. In other words,

$$\tau_{rb} = \tau_g$$
$$\implies k\ell h = mgw$$

We know $w = 5, h = 0.5$, and we take $k = 1$

$$\implies \ell = 2mg(5)$$
$$\implies \sqrt{1 + l^2} = 10mg$$
$$\implies l = \sqrt{100m^2g^2 - 1}$$

Because cardboard is quite light, we can approximate the weight of the cardboard to be $\approx 0.2lbs^2$ (factor of safety of 10)

$$\implies l = \sqrt{100 * 0.2^2 - 1} = 1.7\text{in}$$

This implies that the hypotenuesal distance of the rubber band should be 1.7 inches.

Cost Analysis

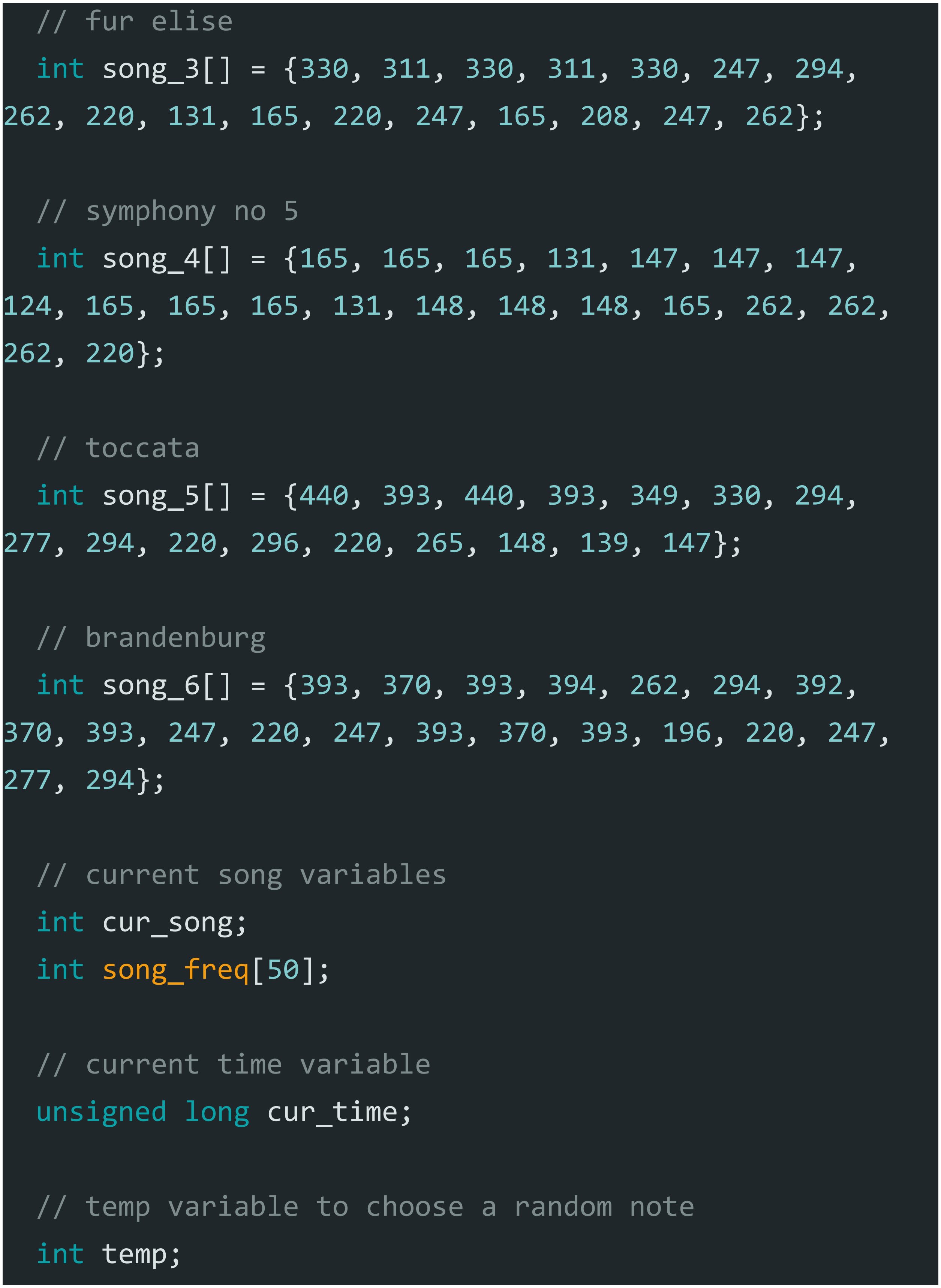
Item	Quantity Used	Total Cost
12”x24”x½” in Wood	2	$39.98
Button (with wires)	6	$9.00
ELEGOO 120 pcs Wires 40pin Male to Female 40pin Male to Male 40pin Female to Female	60	$3.49
Gikfun 2" 4Ohm 3W Full Range Audio Speaker Stereo Woofer Loudspeaker for Arduino (Pack of 2pcs) EK1725	1	$4.84
Cardboard	2	$0.00
Paper	6	$0.00
Ultrasonic sensor	1	*Can be ignored as per the assignment (part of Arduino kit).*
Servo Motor	1	*Can be ignored as per the assignment (part of Arduino kit).*
Arduino	2	*Can be ignored as per the assignment (part of Arduino kit).*
LED	6	*Can be ignored as per the assignment (part of Arduino kit).*
Duct Tape	1	*Can be ignored as per Marra’s words (everything in the lab is available for use)*
Popsicle sticks	5	*Can be ignored as per Marra’s words (everything in the lab is available for use)*
Rubber bands	3	*Can be ignored as per Marra’s words (everything in the lab is available for use)*
Nails	5	*Can be ignored as per Marra’s words (everything in the lab is available for use)*

Dowel	3	*Can be ignored as per Marra’s words (everything in the lab is available for use)*
Black paint	1	*Can be ignored as per Marra’s words (everything in the lab is available for use)*
Foamcore	2	*Can be ignored as per Marra’s words (everything in the lab is available for use)*

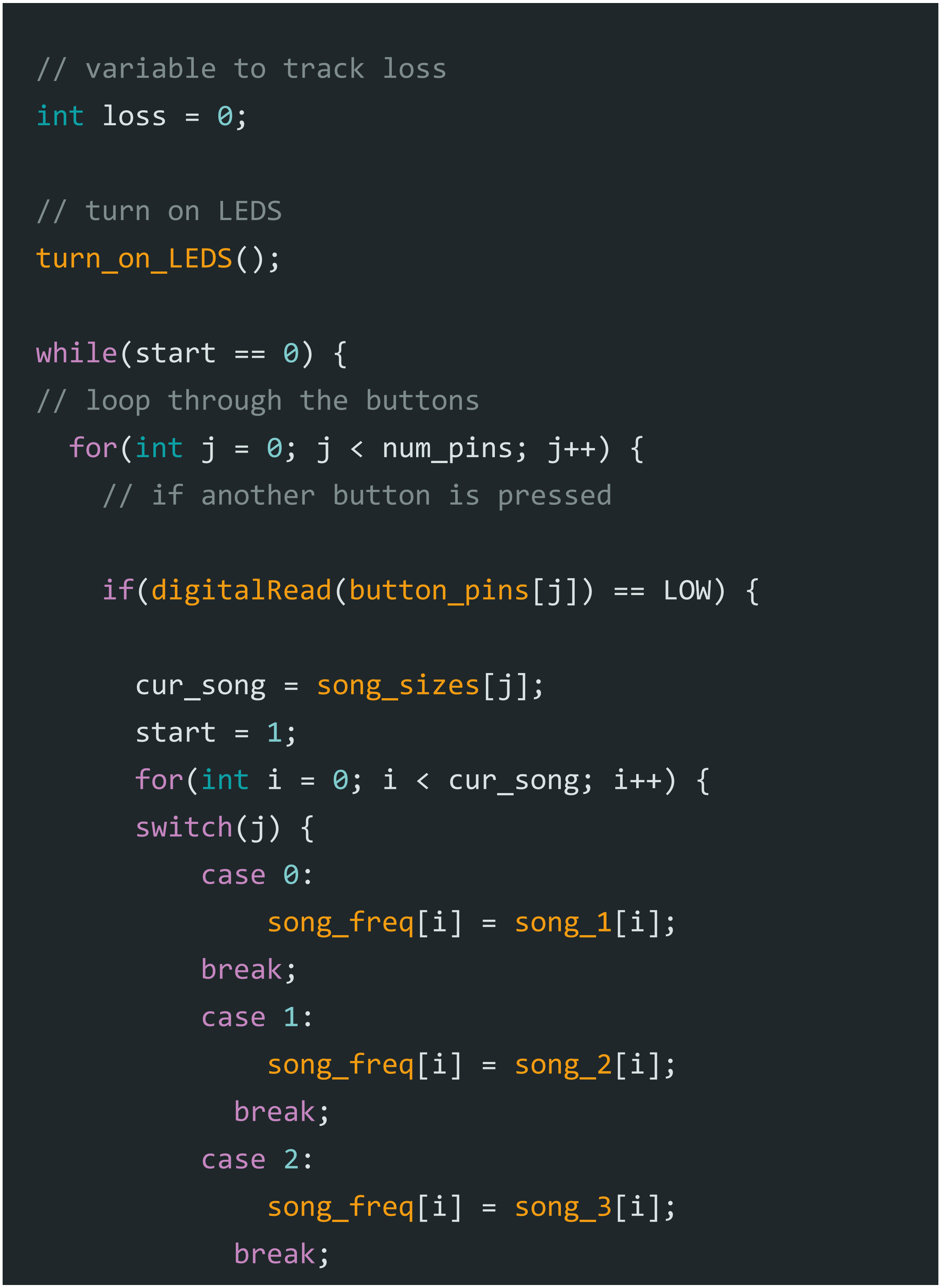
Code Section:

```
/* Code for the game itself */

//led pins
int led1 = 8;
int led2 = 9;
int led3 = 10;
int led4 = 11;
int led5 = 12;
int led6 = 13;

int led_pins[] = {led1, led2, led3, led4, led5, led6};

//button pins
int button1 = 2;
int button2 = 3;
int button3 = 4;
int button4 = 5;
int button5 = 6;
int button6 = 7;

int button_pins[] = {button1, button2, button3, button4,
button5, button6};
```

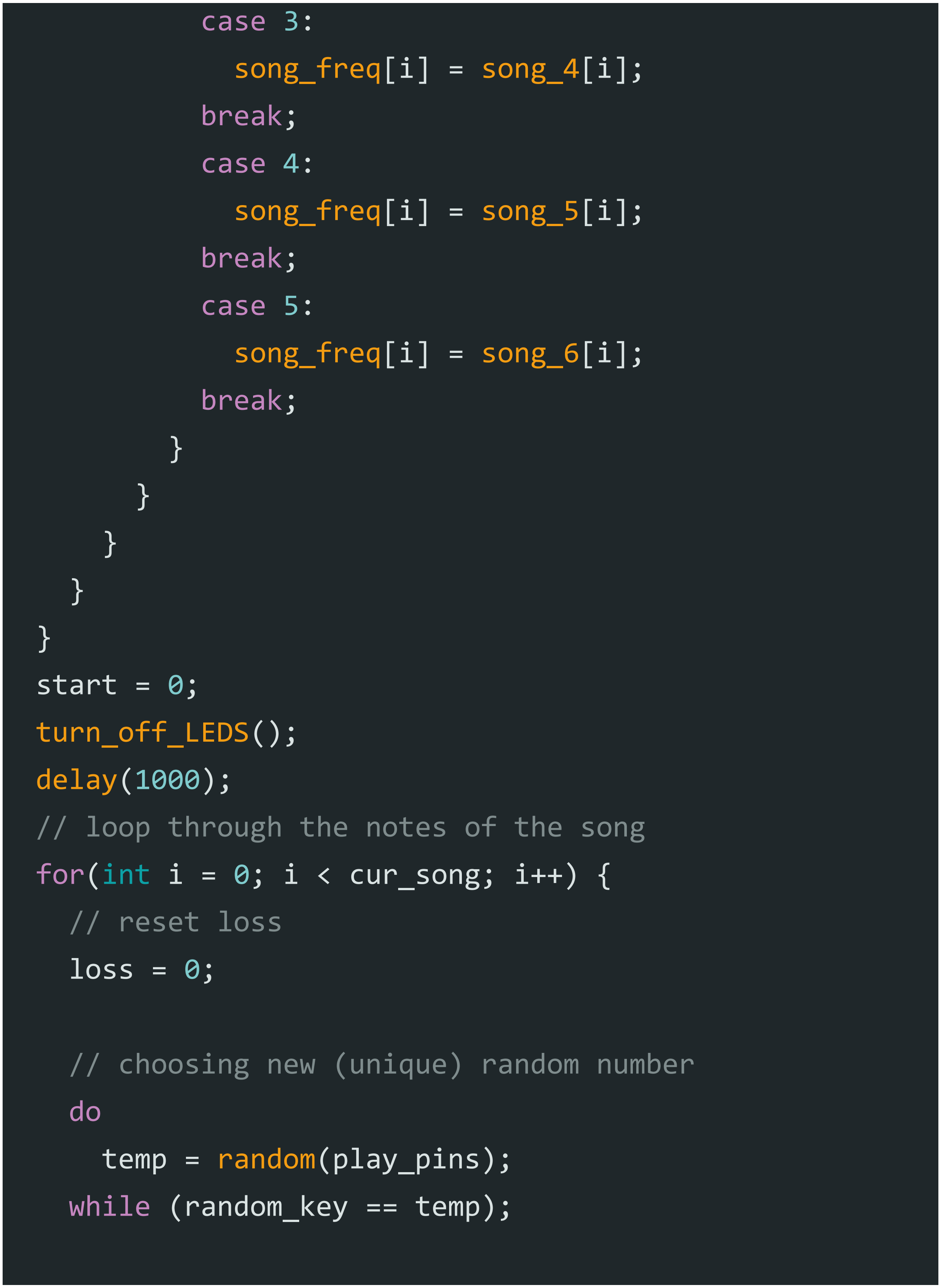
```
// speaker pin
int speaker = A0;

int num_pins = 6; // number of pins being used
int play_pins = 6; // number of pins in play (used for
testing)
int random_key = 0; // integer for random key

unsigned long start_time; // ulong for start time
int start = 0; // variable to track if player has
started

// time duration
unsigned long TIME_DURATION = 3000;

void setup() {

  // setting up buttons and leds
  for(int i = 0; i < num_pins; i++) {
    pinMode(led_pins[i], OUTPUT);
    pinMode(button_pins[i], INPUT);
  }

  // begin
  Serial.begin(9600);
  // save initial start time (should be around 0)
  start_time = millis();
}

void loop() {

  // length of each song
  int song_1_size = 15;
  int song_2_size = 31;
  int song_3_size = 17;
  int song_4_size = 20;
  int song_5_size = 16;
  int song_6_size = 20;

  int song_sizes[] = {song_1_size, song_2_size,
song_3_size, song_4_size, song_5_size, song_6_size};

  // ode to joy
  int song_1[] = {247, 247, 262, 294, 294, 262, 247,
220, 196, 196, 220, 247, 247, 220, 220};

  // moonlight sonata
  int song_2[] = {220, 294, 349, 220, 294, 349, 220,
294, 349, 220, 294, 349, 233, 294, 349, 233, 294, 349,
233, 311, 392, 233, 311, 392, 220, 294, 349, 220, 277,
330, 294};
```



```
  // fur elise
  int song_3[] = {330, 311, 330, 311, 330, 247, 294,
262, 220, 131, 165, 220, 247, 165, 208, 247, 262};

  // symphony no 5
  int song_4[] = {165, 165, 165, 131, 147, 147, 147,
124, 165, 165, 165, 131, 148, 148, 148, 165, 262, 262,
262, 220};

  // toccata
  int song_5[] = {440, 393, 440, 393, 349, 330, 294,
277, 294, 220, 296, 220, 265, 148, 139, 147};

  // brandenburg
  int song_6[] = {393, 370, 393, 394, 262, 294, 392,
370, 393, 247, 220, 247, 393, 370, 393, 196, 220, 247,
277, 294};

  // current song variables
  int cur_song;
  int song_freq[50];

  // current time variable
  unsigned long cur_time;

  // temp variable to choose a random note
  int temp;
```

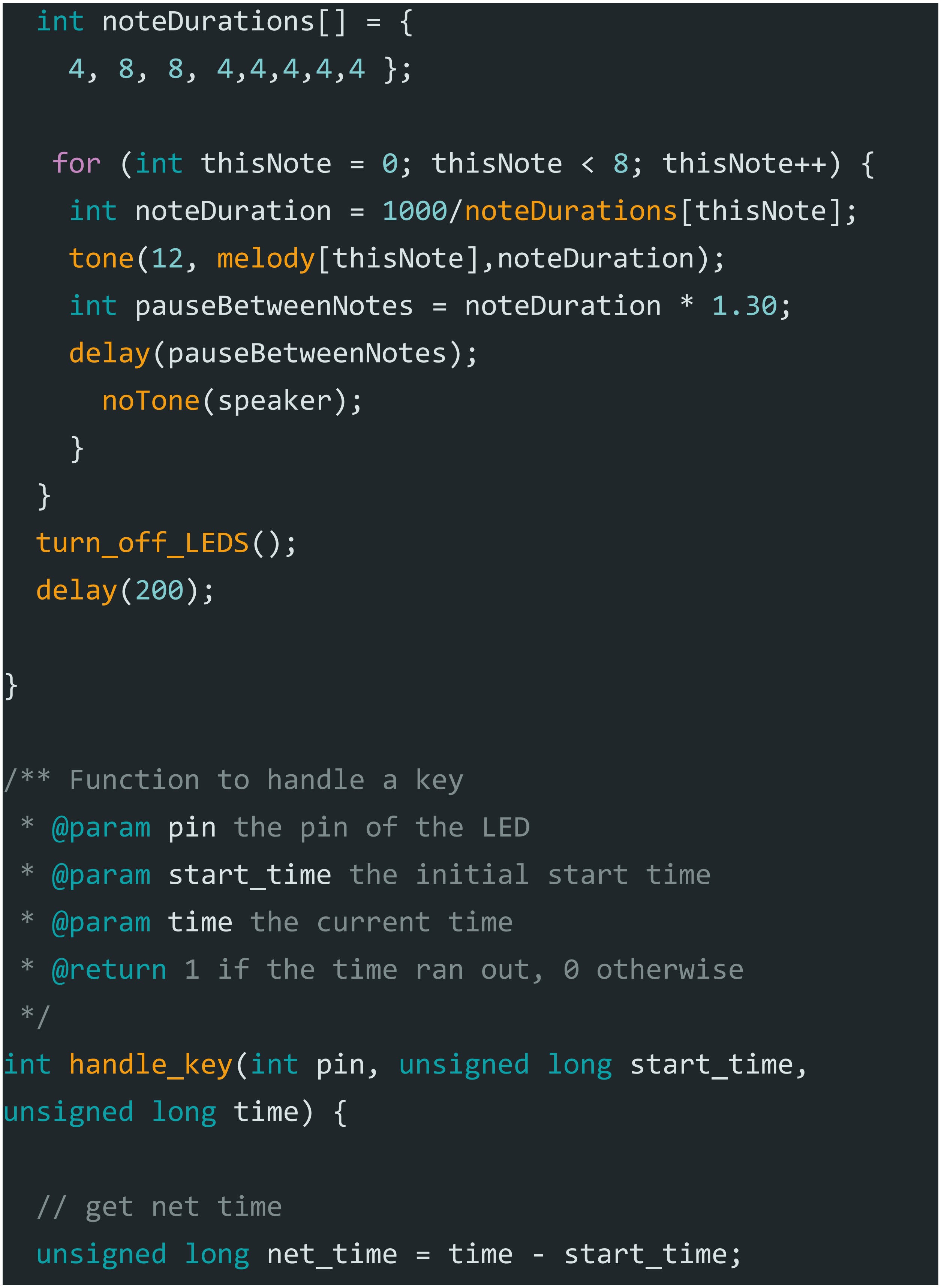


```
// variable to track loss
int loss = 0;

// turn on LEDS
turn_on_LEDS();

while(start == 0) {
// loop through the buttons
  for(int j = 0; j < num_pins; j++) {
    // if another button is pressed

    if(digitalRead(button_pins[j]) == LOW) {

      cur_song = song_sizes[j];
      start = 1;
      for(int i = 0; i < cur_song; i++) {
      switch(j) {
          case 0:
              song_freq[i] = song_1[i];
          break;
          case 1:
              song_freq[i] = song_2[i];
            break;
          case 2:
              song_freq[i] = song_3[i];
            break;
```

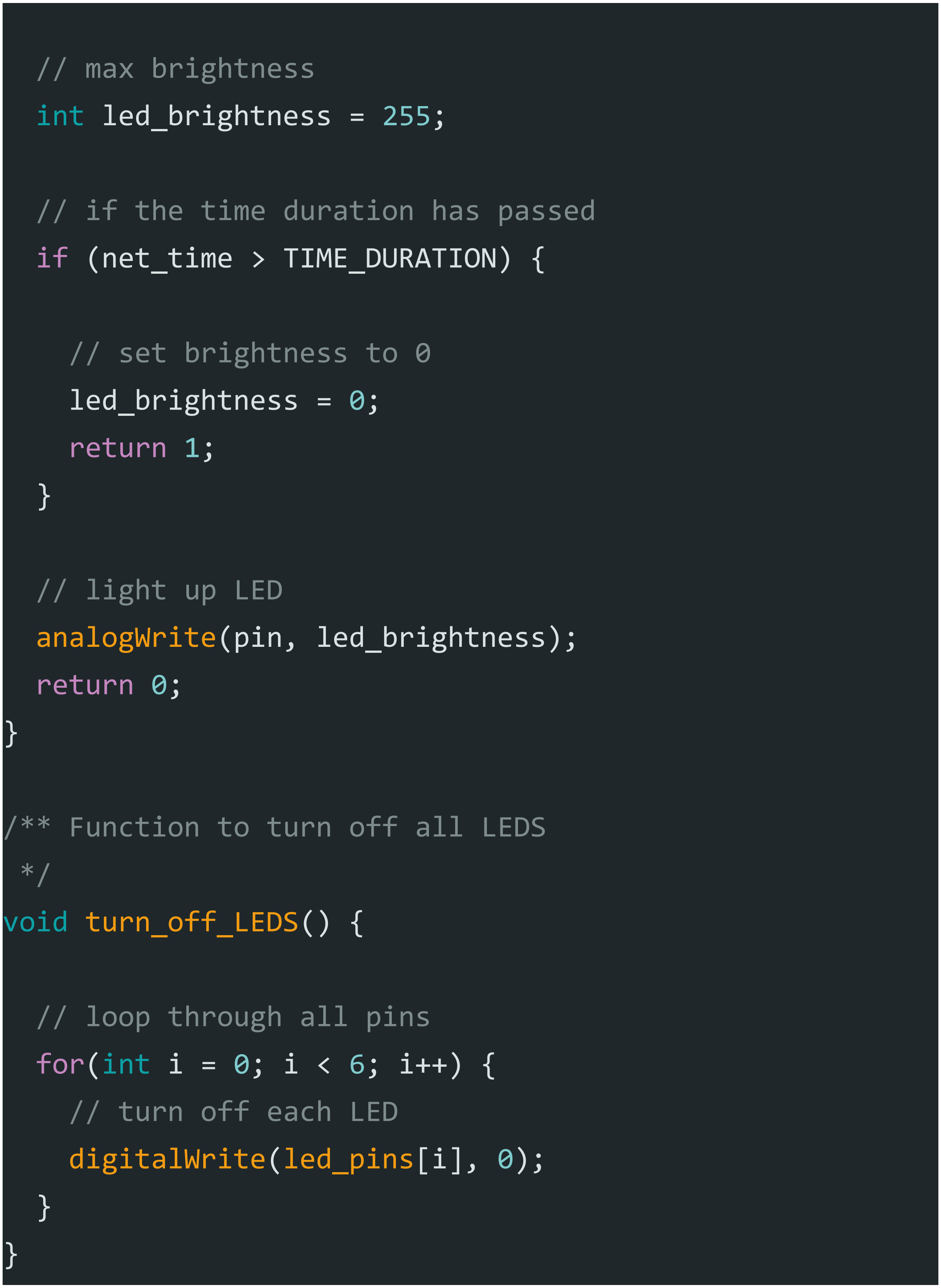


```
          case 3:
            song_freq[i] = song_4[i];
          break;
          case 4:
            song_freq[i] = song_5[i];
          break;
          case 5:
            song_freq[i] = song_6[i];
          break;
        }
      }
    }
  }
}
start = 0;
turn_off_LEDS();
delay(1000);
// loop through the notes of the song
for(int i = 0; i < cur_song; i++) {
  // reset loss
  loss = 0;

  // choosing new (unique) random number
  do
    temp = random(play_pins);
  while (random_key == temp);
```

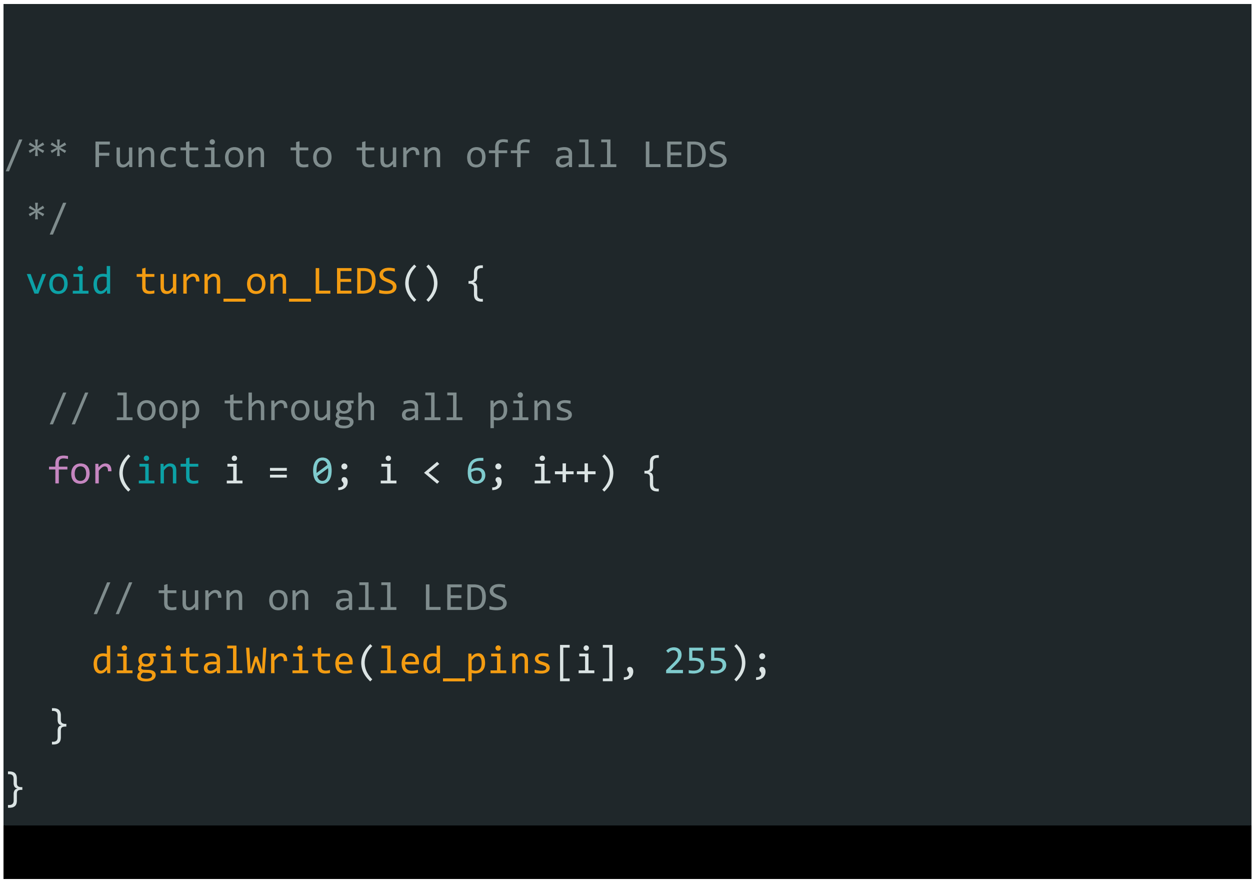
```
    random_key = temp;

    // initialize the start and current times
    start_time = millis();
    cur_time = millis();

    // loop through until the button is pressed OR until
the time runs out
    while (handle_key(led_pins[random_key], start_time,
cur_time) == 0 && digitalRead(button_pins[random_key])
!= LOW) {

      // update the time
      cur_time = millis();
      // loop through the buttons
      for(int j = 0; j < play_pins; j++) {

        // skip the current button
        if(j == random_key) continue;

        // if another button is pressed
        if(digitalRead(button_pins[j]) == LOW) {

          // raise loss flag
          loss = 1;
          // escape for loop
          break;
        }
      }

      // escape while loop
      if (loss == 1) {
        break;
      }
    }

    // if either time ran out or the wrong key was
pressed
    if (handle_key(led_pins[random_key], start_time,
cur_time) != 0 || loss == 1) {

      // activate "loss"
      turn_on_LEDS();

      // add delay to let the lights display
      delay(1000);
      loss = 1;

      // turn off speaker
      noTone(speaker);

      // reset song
      i = 0;
      break;
    } else {

      // play the current note
      tone(speaker, song_freq[i], 500);
    }

    // turn off LEDS (resetting them)
    turn_off_LEDS();

    // replaces delay (only continue when user has
stopped pressing the button)
    while(loss == 0 &&
digitalRead(button_pins[random_key]) == LOW) {
    }
    delay(50);
  }

  // if the user won, play the win melody
  if(loss == 0) {
    delay(300);
    turn_on_LEDS();
    delay(300);
    int melody[] = {
    262, 196,196, 220, 196,0, 247, 262};

  // note durations: 4 = quarter note, 8 = eighth note,
etc.:
```



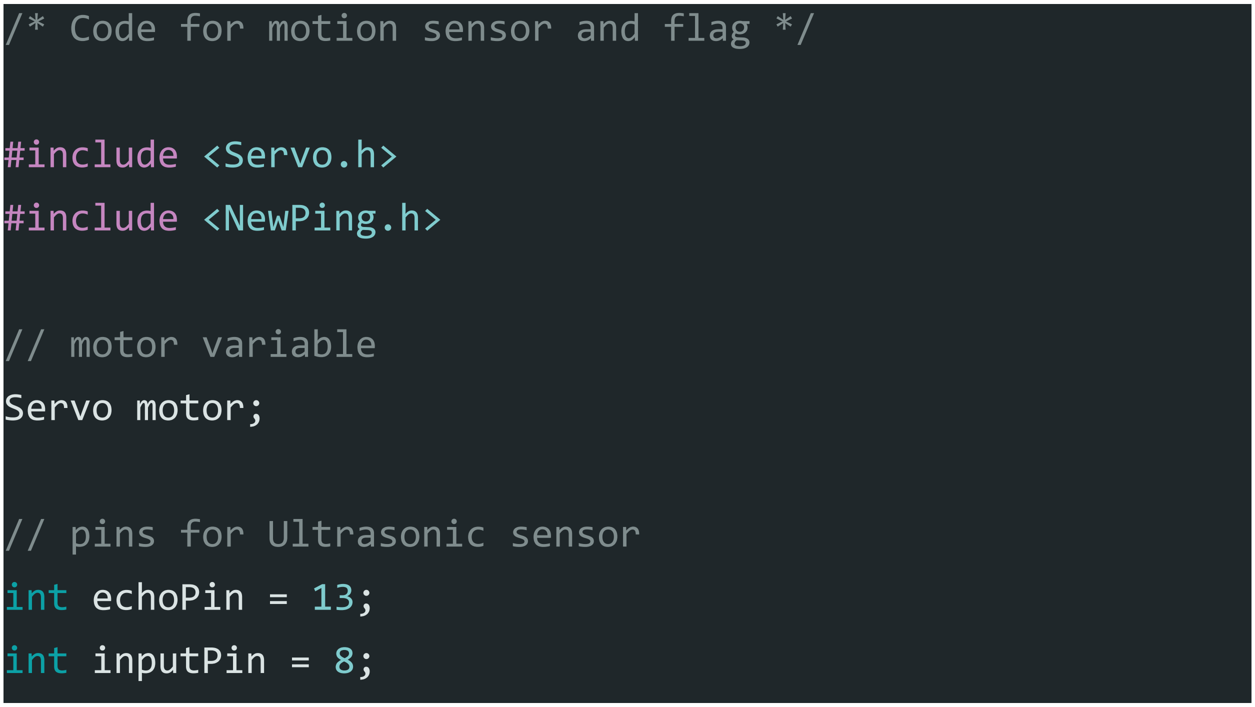
```
  int noteDurations[] = {
    4, 8, 8, 4,4,4,4,4 };

   for (int thisNote = 0; thisNote < 8; thisNote++) {
    int noteDuration = 1000/noteDurations[thisNote];
    tone(12, melody[thisNote],noteDuration);
    int pauseBetweenNotes = noteDuration * 1.30;
    delay(pauseBetweenNotes);
      noTone(speaker);
    }
  }
  turn_off_LEDS();
  delay(200);

}

/** Function to handle a key
 * @param pin the pin of the LED
 * @param start_time the initial start time
 * @param time the current time
 * @return 1 if the time ran out, 0 otherwise
 */
int handle_key(int pin, unsigned long start_time,
unsigned long time) {

  // get net time
  unsigned long net_time = time - start_time;
```



```
  // max brightness
  int led_brightness = 255;

  // if the time duration has passed
  if (net_time > TIME_DURATION) {

    // set brightness to 0
    led_brightness = 0;
    return 1;
  }

  // light up LED
  analogWrite(pin, led_brightness);
  return 0;
}

/** Function to turn off all LEDS
 */
void turn_off_LEDS() {

  // loop through all pins
  for(int i = 0; i < 6; i++) {
    // turn off each LED
    digitalWrite(led_pins[i], 0);
  }
}
```

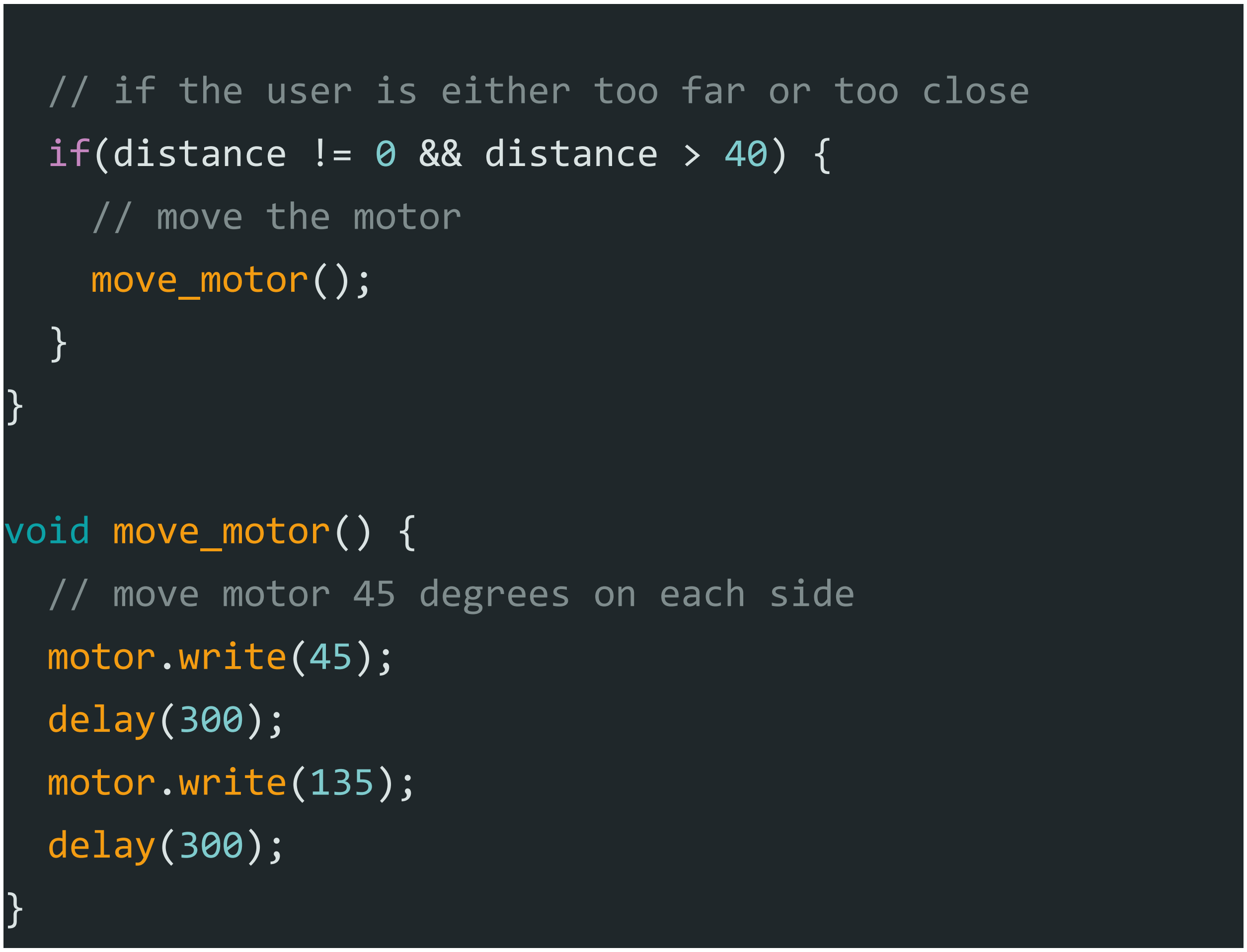


```
/** Function to turn off all LEDS
 */
 void turn_on_LEDS() {

  // loop through all pins
  for(int i = 0; i < 6; i++) {

    // turn on all LEDS
    digitalWrite(led_pins[i], 255);
  }
}
```



```
/* Code for motion sensor and flag */

#include <Servo.h>
#include <NewPing.h>

// motor variable
Servo motor;

// pins for Ultrasonic sensor
int echoPin = 13;
int inputPin = 8;
```

```
// pin for motor
int motor_pin = 4;

// initializing the sonar with a max distance of 100 cm
NewPing sonar(inputPin, echoPin, 100);

// var for distance
int distance;

void setup() {
  // declaring variables
  pinMode(inputPin, INPUT);
  motor.attach(motor_pin);
  motor.write(90);
  Serial.begin(9600);
}

void loop() {

  delay(50);

  // read distance
  distance = sonar.ping_cm();

  // print distance (testing)
  Serial.println(distance);
```



```
  // if the user is either too far or too close
  if(distance != 0 && distance > 40) {
    // move the motor
    move_motor();
  }
}

void move_motor() {
  // move motor 45 degrees on each side
  motor.write(45);
  delay(300);
  motor.write(135);
  delay(300);
}
```

Reflections

What went right and what went wrong?

- The piano key mechanism worked well, however, there were a few keys in which the rubber bands came loose so they had fallen. These keys were still functional.
- The code worked well except for a bug with the first and last piano keys (predicted to be physical).
- We weren't able to accomplish all that we had planned but still had something that worked.

What would you change if you had to do it again?

- We wish we focused more time on the physical aspect of the project/cosmetics, as we ran out of time for it at the end.
- We wish that we had found a better way to attach the rubber bands to the key to make them last longer.

What things did you learn that will help you in your next design project?

- We learned that materials selection is very important to get down initially, as we had an issue with that.

What suggestions do you have for improving the design project next year?

- We think that the design project for next year should not have a random selection, or at least make the range of choices reasonable (some groups got a disadvantage compared to others).
- We think that wood should be provided since it costs a lot and takes up most of the budget.

Other Considerations

When we were developing the piano key mechanism, we created it out of recyclable and cheap materials as a way to reduce our budget. We also ensured that the code allowed users of all reaction times to participate, as it did not punish the user for holding on to the button for too long. For safety, we made sure that the game was stable and that no nails or wires were sticking out that any users could touch.

To maintain cultural sensitivity, we made efforts to pick themes that were not offensive to any group of people. Even though the song selections and composers are from Germany, classical music, in particular, is common and developed across many different European countries.