

LegSense: A Lower Body Sensory Device

GEOFFREY COHEN*

Winchester Thurston School

ABSTRACT

LegSense is a product designed to aid novice and professional weightlifters as well as doctors and physical therapists.

1. Introduction

The goal of the author's project is to create an efficient lower body measurement device that will track the movement of the user's legs in order to perfect technique, and prevent injury in athletics. In all athletic events there is a proper technique to the motion of the player. For example, though there may be an infinite number of ways to get a basketball into the hoop from the foul line, there is one ideal shot that will go in every time. Every basketball player strives for their shot to be as perfect as possible. How can they achieve this level of perfection? It comes down to the player's movement. If the player squats exactly at this angle and releases the ball with exactly this much force at the precise angle every time, he or she will never miss. The goal of the author's project is to achieve that level of perfection across multiple domains using their product.

The author decided to focus their attention on lower body extremities due to their passion for track and field. In track and field, every event demands perfect technique. Whether one is running, jumping, or throwing, lower body movement and power is necessary to be successful. For sprinters, it is their whole world. Its not as simple as simple as just picking up one's legs and moving forward. There is a technique. For each individual athlete there is an ideal stride length, knee raise, foot position, etc. Furthermore, for jumpers there are even exact angles an athlete should achieve in order to produce the maximum distance. For example, triple jumpers must raise their knees to an ideal 90° from the body bringing their thigh parallel to the runway and hold this position for as long as possible during the "step" phase of a jump. You can see this by looking at figures 1 , 2, and 3 that although all three of

these jumpers all are from different countries, have different coaching staffs, and different training they are all striving to achieve that 90° . However, not all athletes have access to coaching that can help them achieve such perfection. This is one of the tasks of this product. To aid the user in the perfection of their technique.



FIG. 1. Dzmitry Platnitski's "step" phase in the triple jump event at the London Olympics in 2012



FIG. 2. Benjamin Compaoré's "step" phase in the triple jump event at the London Olympics in 2012

In order for this product to be useful in perfecting the technique of various athletes, it must be functional for a diverse group of users: tall, short, bigger, thinner etc. As a result, the product is a pair of running tights capable of being compatible with a wide range of athletes. This is necessary because another function of this product is to be used by athletes with different body types than track athletes, specifically weight lifters. This product is primarily

*Current affiliation: Winchester Thurston School



FIG. 3. Christian Taylor's "step" phase in the triple jump event at the London Olympics in 2012

targeted to be used to help weightlifters achieve ideal technique, lower risk during lifting, and prevent injury. Additionally the product can be used to measure range of motion of those in physical therapy who have already been injured.

Similar products include that of a product designed by BMW to help train swimmers to achieve the ideal movements while underwater. For the Rio 2016 Olympics some U.S.A. athletes trained with this product designed by BMW. The program was successful as it helped American athletes achieve 16 gold medals at the games. LegSense is different from BMW's because it only measures lower body where as BMW's performs full body measurement. Additionally, the author's project use different technical sensors that are far less sophisticated.

2. Implementation

First the author started the project by testing the basic functions of the sensor. For this project, the LSM9DS1 Breakout Board (figure 4) was the sensor used. The board itself is a composition of three individual sensors: a gyroscope, an accelerometer, and a magnetometer. A gyroscope measures angular velocity in degrees per second. Put simply, a gyroscope measures how fast the board is rotating along a certain axis. The second sensor, the accelerometer, measures acceleration in meters per second per second (m/s^2). The acceleration of the board computes if the velocity of the board is speeding up or slowing down. Finally, the magnetometer is a sensor that measures surrounding magnetic fields in gauss. These sensors are depicted below in figure 5.

After compiling all of the data from each individual sensor, the orientation of the board in space can be quite accurate. For the simplicity of this project, the author chose to focus primarily on the accelerometer to determine orientation. This decision was made because the data accelerometer provided was functional for measuring the necessary angles in this project.

```

1 void looper() {
  imu.readAccel();
3
  Serial.println(imu.calcAccel(imu.ax), 2);

```

```

5   Serial.println(imu.calcAccel(imu.ay), 2);
7   Serial.println(imu.calcAccel(imu.az), 2);
9 }
```

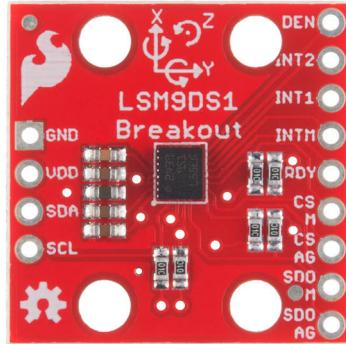


FIG. 4. LSM9DS1 Breakout Board

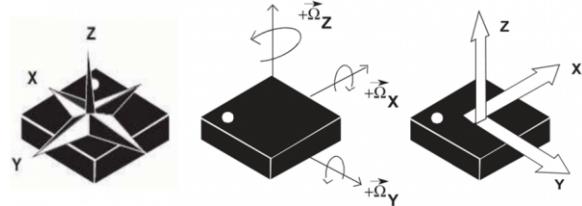


FIG. 5. Pitch Roll and Yaw for Various Sensors

Once the author was able to receive and store the incoming data from the board, the author progressed to implementing multiple boards. The project would require eight total boards to accurately compute the data for each of the four joints (right hip, left hip, right knee, left knee.) Each limb would require two LSM9DS1 boards each.

The LSM9DS1 board delivers the sensor data to the arduino microcomputer via Inter-integrated Circuits (I2C.) I2C communication was more effective for this project than its competitors, serial ports and SPI, due to its hardware simplicity only requiring two hard connections as well as its superior speed. Regardless of these benefits there is one drawback to using I2C communication: an arduino microcomputer is only capable of communicating with a single I2C device. To overcome this disadvantage the author invested in a TCA select device I2C expander (TCA9548A) as shown in figure 6 developed and manufactured by Adafruit Industries.

With the TCA select device, the arduino microcomputer is capable of communicating over I2C with up to eight sensors. The precise amount required for the project.

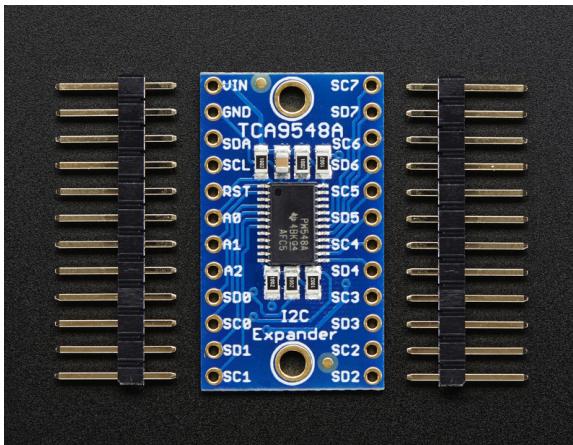


FIG. 6. TCA Select Device

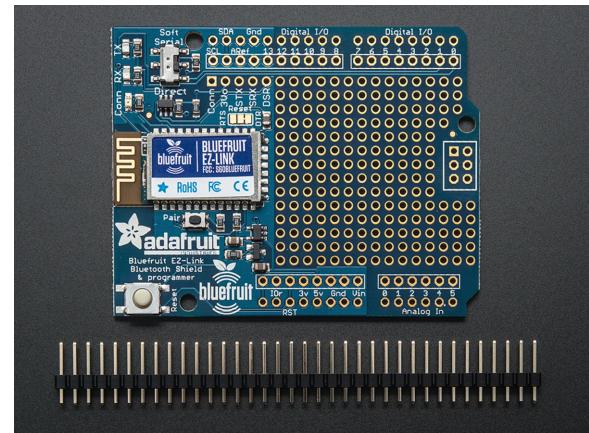


FIG. 7. EZ-Link Bluefruit Shield

The TCA select device works by selecting the called upon sensor, reading the data, then moving on to the next.

```

2 void tcaselect(uint8_t i) {
3   if (i > 7) return;
4
5   Wire.beginTransmission(TCAADDR);
6   Wire.write(1 << i);
7   Wire.endTransmission();
}
```

It was important to the author that device be functional via bluetooth. If the product were ever to be used practically in the field, wireless compatibility would be necessary to reduce clutter and enhance simplicity. In order for the product to communicate via bluetooth, the author invested in a device called Bluefruit EZ-Link, as shown in figure 7, manufactured by Adafruit Industries. The Bluefruit EZ-Link shield is an arduino shield that is capable of sending data from the arduino to the serial monitor on a computer via bluetooth.

```

1 port = new Serial(this, "/dev/cu.AdafruitEZ-
2 Link88b3-SPP", 9600);
3 port.bufferUntil('\n');
```

Once the bluetooth was functioning, the fundamentals of the project were complete. The device was able to read the incoming data from all eight sensors then transmit them to a computer via bluetooth, where they could be read and interpreted.

Once the fundamentals of the project were running successfully, the author proceeded to design a holder for each sensor. The holder had to be light weight, compact, sturdy, and most importantly, functional. The holder would support two sensors simultaneously then be sewed

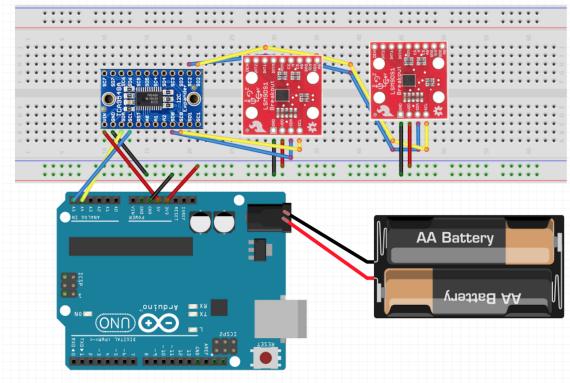


FIG. 8. Diagram

to a pair of running tights. To design this piece, the author used Autodesk Inventor 3-D modeling. To test each improving model, the author would print the model using a 3-D printer and check for sensor compatibility. After many iterations

From there, the author proceeded to wire the device. Each sensor required a total of four wires. Rather than have 32 wires (eight sensors multiplied by four wires each) sprawled from the user's hips to legs, the author decided to braid the wires into four groups, one for each position of the sensor.

The author then brought all of the hardware together by sewing the soldered arduino, EZ-Link shield, and TCA select device, to the hips of the tights, then sewing each sensor to the various parts of the legs, one on each quad, as well as one on each lower leg below the knee. Now that the sensors and arduino were connected to the tights, the proper length wire was soldered between the two then sewn down for security.

ARDUINO CODE

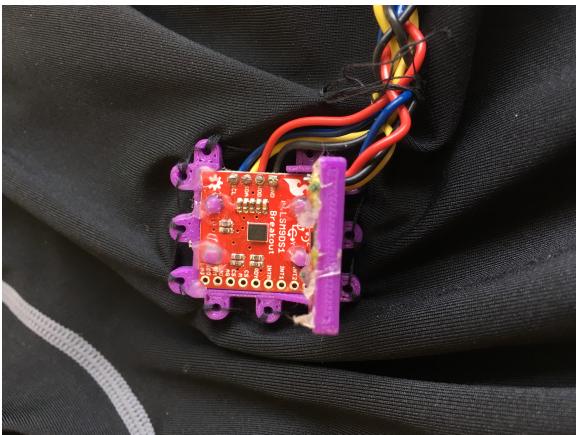


FIG. 9. Sensor Holder In Place

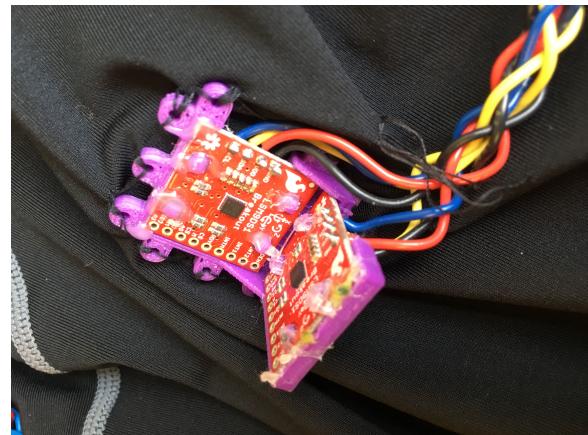


FIG. 12. Final Sensor Holder Configuration



FIG. 10. Braided Wire

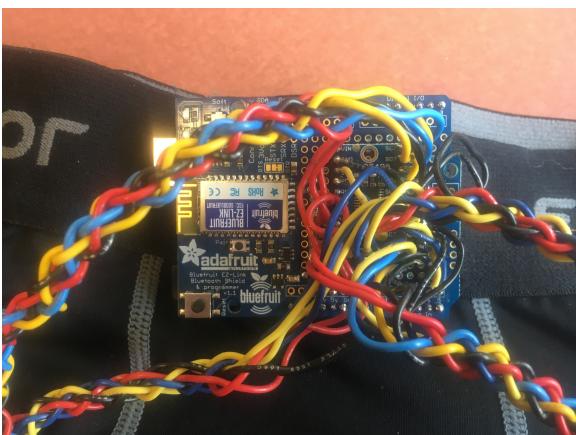


FIG. 11. Final Arduino with Components

```
2 #include<Wire.h>
# include "Wire.h"
# define TCAADDR 0x70
```

```
4 extern "C" {
5 #include <utility/twi.h> // from Wire library ,
6   so we can do bus scanning
7
8 #include <LSM9DS1_Registers.h>
9 #include <LSM9DS1_Types.h>
10 #include <SparkFunLSM9DS1.h>
11 #include <SPI.h>
12 LSM9DS1 imu;
13 #define LSM9DS1_AG 0x6B // Would be 0x6A if
14   SDO_AG is LOW
15 #define PRINT_CALCULATED
16
17 const int MPU_addr = 104; // I2C address of the
18   MPU-6050
19 int16_t AcX, AcY, AcZ, Tmp, GyX, GyY, GyZ;
20
21 void tcaselect(uint8_t i) {
22   if (i > 7) return;
23
24   Wire.beginTransmission(TCAADDR);
25   Wire.write(1 << i);
26   Wire.endTransmission();
27
28 void setup() {
29   Serial.begin(9600);
30   Wire.begin();
31
32   tcaselect(0);
33   setter();
34
35   tcaselect(1);
36   setter();
37
38   tcaselect(2);
39   setter();
40
41   tcaselect(3);
42   setter();
43
44   tcaselect(4);
45   setter();
```

```

46 tcaselect(5);
47 setter();
48
49 tcaselect(6);
50 setter();
51
52 tcaselect(7);
53 setter();
54
55 Serial.println("done");
56
57 }
58 void loop() {
59
60 tcaselect(0);
61 looperZero();
62
63 tcaselect(1); // THIS IS WHERE THE LAG IS
64 looperOne(); //ON THESE FUNCTIONS HERE
65
66 tcaselect(2);
67 looperTwo();
68
69 tcaselect(3);
70 looperThree();
71
72 tcaselect(4);
73 looperFour();
74
75 tcaselect(5);
76 looperFive();
77
78 tcaselect(6);
79 looperSix();
80
81 tcaselect(7);
82 looperSeven();
83 //delay(60);
84
85 }
86
87 void setter() {
88   Serial.begin(9600);
89   imu.settings.device.commInterface =
90     IMU_MODE_I2C;
91   imu.settings.device.agAddress = LSM9DS1_A_G;
92   imu.begin();
93
94
95 void looperZero() {
96   imu.readAccel();
97
98   Serial.print("A");
99   Serial.println(imu.calcAccel(imu.ax), 2);
100
101   Serial.print("B");
102   Serial.println(imu.calcAccel(imu/ay), 2);
103
104   Serial.print("C");
105   Serial.println(imu.calcAccel(imu.az), 2);
106 }
107
108
109
110 void looperOne() {
111   imu.readAccel();
112
113   Serial.print("D");
114   Serial.println(imu.calcAccel(imu.ax), 2);
115
116   Serial.print("E");
117   Serial.println(imu.calcAccel(imu.ay), 2);
118
119   Serial.print("F");
120   Serial.println(imu.calcAccel(imu.az), 2);
121 }
122
123 void looperTwo() {
124   imu.readAccel();
125
126   Serial.print("G");
127   Serial.println(imu.calcAccel(imu.ax), 2);
128
129   Serial.print("H");
130   Serial.println(imu.calcAccel(imu.ay), 2);
131
132   Serial.print("I");
133   Serial.println(imu.calcAccel(imu.az), 2);
134 }
135
136 void looperThree() {
137   imu.readAccel();
138
139   Serial.print("J");
140   Serial.println(imu.calcAccel(imu.ax), 2);
141
142   Serial.print("K");
143   Serial.println(imu.calcAccel(imu.ay), 2);
144
145   Serial.print("L");
146   Serial.println(imu.calcAccel(imu.az), 2);
147 }
148
149 void looperFour() {
150   imu.readAccel();
151
152   Serial.print("M");
153   Serial.println(imu.calcAccel(imu.ax), 2);
154
155   Serial.print("N");
156   Serial.println(imu.calcAccel(imu.ay), 2);
157
158   Serial.print("O");
159   Serial.println(imu.calcAccel(imu.az), 2);
160
161 }
162
163 void looperFive() {
164   imu.readAccel();
165
166   Serial.print("P");
167   Serial.println(imu.calcAccel(imu.ax), 2);
168
169   Serial.print("Q");
170   Serial.println(imu.calcAccel(imu.ay), 2);
171
172   Serial.print("R");
173   Serial.println(imu.calcAccel(imu.az), 2);
174
175 }
```

```

176 }
178 void looperSix() {
179   imu.readAccel();
180
181   Serial.print("S");
182   Serial.println(imu.calcAccel(imu.ax), 2);
183
184   Serial.print("T");
185   Serial.println(imu.calcAccel(imu.ay), 2);
186
187   Serial.print("U");
188   Serial.println(imu.calcAccel(imu.az), 2);
189 }
190
191 void looperSeven() {
192   imu.readAccel();
193
194   Serial.print("V");
195   Serial.println(imu.calcAccel(imu.ax), 2);
196
197   Serial.print("W");
198   Serial.println(imu.calcAccel(imu.ay), 2);
199
200   Serial.print("X");
201   Serial.println(imu.calcAccel(imu.az), 2);
202 }
```

PROCESSING CODE

```

1 import processing.serial.*;
2 Serial port;
3
4 boolean button1;
5 boolean button2;
6 boolean button3;
7 boolean button4;
8 boolean button5;
9 boolean button6;
10 boolean button1r;
11 boolean button2r;
12 boolean button3r;
13 boolean button4r;
14 boolean button5r;
15 boolean button6r;
16 color rightLeg;
17 color leftLeg;
18 color clickHighlight;
19
20 float rightHipX1 = 0;
21 float rightHipY1 = 0;
22 float rightHipZ1 = 0;
23
24 float rightHipX2 = 0;
25 float rightHipY2 = 0;
26 float rightHipZ2 = 0;
27
28
29 float leftHipX1 = 0;
30 float leftHipY1 = 0;
31 float leftHipZ1 = 0;
32
33
34
35 float leftHipX2 = 0;
36 float leftHipY2 = 0;
37 float leftHipZ2 = 0;
38
39
40 float rightLegX1 = 0;
41 float rightLegY1 = 0;
42 float rightLegZ1 = 0;
43
44 float rightLegX2 = 0;
45 float rightLegY2 = 0;
46 float rightLegZ2 = 0;
47
48
49 float leftLegX1 = 0;
50 float leftLegY1 = 0;
51 float leftLegZ1 = 0;
52
53 float leftLegX2 = 0;
54 float leftLegY2 = 0;
55 float leftLegZ2 = 0;
56
57
58 float comboA = 0;
59 float comboB = 0;
60 float comboC = 0;
61 float comboD = 0;
62 float comboE = 0;
63 float comboF = 0;
64 float comboG = 0;
65 float comboH = 0;
66 float comboI = 0;
67 float comboJ = 0;
68 float comboK = 0;
69 float comboL = 0;
70 float comboM = 0;
71 float comboN = 0;
72 float comboO = 0;
73 float comboP = 0;
74 float comboQ = 0;
75 float comboR = 0;
76 float comboS = 0;
77 float comboT = 0;
78 float comboU = 0;
79 float comboV = 0;
80 float comboW = 0;
81 float comboX = 0;
82
83 float comboLf = 0;
84 float comboIf = 0;
85 float comboCf = 0;
86 float comboFff = 0;
87
88
89
90
91 float comboAa = 0;
92 float comboBb = 0;
93 float comboCc = 0;
94 float comboDd = 0;
95 float comboEe = 0;
96 float comboFf = 0;
97 float comboGg = 0;
98 float comboHh = 0;
99 float comboIi = 0;
```

```

101 float comboJj = 0;
102 float comboKk = 0;
103 float comboLl = 0;
104 float comboMm = 0;
105 float comboNn = 0;
106 float comboOo = 0;
107 float comboPp = 0;
108 float comboQq = 0;
109 float comboRr = 0;
110 float comboSs = 0;
111 float comboTt = 0;
112 float comboUu = 0;
113 float comboVv = 0;
114 float comboWw = 0;
115 float comboXx = 0;
116
117 float tol = 10;
118
119 int limb = 100;
120
121 int colorR = 255;
122 int colorG = 0;
123 int colorB = 0;
124
125
126 void setup ()
127 {
128   size(1200, 800, P3D);
129   background(0, 0, 0);
130   //frameRate(10);
131   button1 = false;
132   button2 = false;
133   button3 = false;
134   button4 = false;
135   button5 = false;
136   button6 = false;
137   button1r = false;
138   button2r = false;
139   button3r = false;
140   button4r = false;
141   button5r = false;
142   button6r = false;
143   rightLeg = color(255, 0, 0); //RED
144   leftLeg = color(0, 0, 255); //BLUE
145   clickHighlight = color(20, 192, 245);
146
147   port = new Serial(this, "/dev/cu.AdafruitEZ-
148     Link88b3-SPP", 9600); // /dev/cu.usbmodem1411
149   port.bufferUntil('\n');
150 }
151
152 void draw()
153 {
154   // if (frameCount % 2 == 0)
155   // {
156   // float rightHipAngle = map(rightHip, -16424,
157   //   16424, -180, 180);
158   background(0, 0, 0);
159
160   fill(255);
161   stroke(0);
162   strokeWeight(0);
163   rect(0, 30, 50, 70, 0, 7, 7, 0); // tab1
164   rect(0, 105, 50, 70, 0, 7, 7, 0); // tab2
165   rect(0, 180, 50, 70, 0, 7, 7, 0); // tab3
166   rect(0, 305, 50, 70, 0, 7, 7, 0); // tab4
167   rect(0, 380, 50, 70, 0, 7, 7, 0); // tab5
168   rect(0, 455, 50, 70, 0, 7, 7, 0); // tab6
169
170
171   rect(0, 740, 120, 60, 7, 7, 0, 0); // angle
172   toggle
173
174   fill(0);
175   textSize(12);
176   text("Right Click For", 17, 760);
177   textSize(30);
178   text("Angle", 17, 787);
179
180   // trackBack();
181
182   tab1character();
183   tab2character();
184   tab3character();
185   tab4character();
186   tab5character();
187   tab6character();
188   checkButtons();
189   changeTab();
190
191   // println(frameRate);
192
193   // println(rightHipAngle);
194 }
195
196 void changeTab()
197 {
198   if (button1)
199   {
200     tab1change();
201     tab1character();
202   } else if (button2) {
203     tab2change();
204     tab2character();
205   } else if (button3) {
206     tab3change();
207     tab3character();
208   } else if (button1r) {
209     tab1change();
210     tab1character();
211     angles();
212   } else if (button2r) {
213     tab2change();
214     tab2character();
215     anglesRight();
216   } else if (button3r) {
217     tab3change();
218     tab3character();
219     anglesLeft();
220   }
221
222   if (button4) {
223     tab4change();
224     tab4character();
225   } else if (button5) {
226     tab5change();
227   }

```

```

227     tab5character();
228 } else if (button6) {
229     tab6change();
230     tab6character();
231 } else if (button4r) {
232     tab4change();
233     tab4character();
234     anglesSide();
235 } else if (button5r) {
236     anglesRightSide();
237     tab5change();
238     tab5character();
239 } else if (button6r) {
240     anglesLeftSide();
241     tab6change();
242     tab6character();
243 }
244
245 // if (button1 == true && button5r == true) {
246 //     blackAngles();
247 //     tab1change();
248 //     tab5change();
249 //     angles();
250 // }
251 }
252
253 void checkButtons()
254 {
255     if (mousePressed && (mouseX >0 && mouseX<50
256         && mouseY>30 && mouseY<100) && (mouseButton
257         == LEFT))
258     {
259         button1 = true;
260         button2 = false;
261         button3 = false;
262         button1r = false;
263         button2r = false;
264         button3r = false;
265     }
266     if (mousePressed && (mouseX >0 && mouseX<50
267         && mouseY>105 && mouseY<175) && (mouseButton == LEFT))
268     {
269         button1 = false;
270         button2 = true;
271         button3 = false;
272         button1r = false;
273         button2r = false;
274         button3r = false;
275     }
276     if (mousePressed && (mouseX >0 && mouseX<50
277         && mouseY>185 && mouseY<250) && (mouseButton == LEFT))
278     {
279         button1 = false;
280         button2 = false;
281         button3 = true;
282         button1r = false;
283         button2r = false;
284         button3r = false;
285     }
286     if (mousePressed && (mouseX >0 && mouseX<50
287         && mouseY>305 && mouseY<375) && (mouseButton == LEFT))
288     {
289         button4 = true;
290
291         button5 = false;
292         button6 = false;
293         button4r = false;
294         button5r = false;
295         button6r = false;
296     }
297
298     if (mousePressed && (mouseX >0 && mouseX<50
299         && mouseY>450 && mouseY<520) && (mouseButton == LEFT))
300     {
301         button4 = false;
302         button5 = true;
303         button6 = false;
304         button4r = false;
305         button5r = false;
306         button6r = false;
307     }
308
309     if (mousePressed && (mouseX >0 && mouseX<50
310         && mouseY>30 && mouseY<100) && (mouseButton == RIGHT))
311     {
312         button1 = false;
313         button2 = false;
314         button3 = false;
315         button1r = true;
316         button2r = false;
317         button3r = false;
318     }
319
320     if (mousePressed && (mouseX >0 && mouseX<50
321         && mouseY>105 && mouseY<175) && (mouseButton == RIGHT))
322     {
323         button1 = false;
324         button2 = false;
325         button3 = false;
326         button1r = false;
327         button2r = true;
328         button3r = false;
329     }
330
331     if (mousePressed && (mouseX >0 && mouseX<50
332         && mouseY>180 && mouseY<250) && (mouseButton == RIGHT))
333     {
334         button1 = false;
335         button2 = false;
336         button3 = false;
337         button1r = false;
338         button2r = false;
339         button3r = true;
340     }
341
342     if (mousePressed && (mouseX >0 && mouseX<50
343         && mouseY>305 && mouseY<375) && (mouseButton == RIGHT))
344     {
345         button4 = false;
346
347         button5 = false;
348         button6 = false;
349         button4r = false;
350         button5r = false;
351         button6r = false;
352     }

```

```

339 button5 = false;
341 button6 = false;
343 button4r = true;
345 button5r = false;
347 button6r = false;
349 }
351 if (mousePressed && (mouseX >0 && mouseX<50
353 && mouseY>380 && mouseY<450) &&
355 mouseButton == RIGHT))
357 {
359 button4 = false;
361 button5 = false;
363 button6 = false;
365 button4r = false;
367 button5r = true;
369 button6r = true;
371 delay(0);
373 void tab1character()
375 {
377 stroke(rightLeg); // right Leg tab1
379 strokeWeight(3.5);
381 line(25, 75, 35, 84);

383 stroke(rightLeg); // right Leg tab1
385 strokeWeight(3.5);
387 line(35, 84, 35, 93);

389 stroke(rightLeg); // right arm tab1
391 strokeWeight(3.5);
393 line(25, 60, 35, 75);

395 stroke(rightLeg); // right foot tab1
397 strokeWeight(3.5);
399 line(35, 93, 40, 93);

399 stroke(leftLeg); // left Leg tab1
401 strokeWeight(3.5);
403 line(25, 75, 26, 84);

405 stroke(leftLeg); // left Leg tab1
407 strokeWeight(3.5);
409 line(26, 84, 18, 87);

411 stroke(leftLeg); // left arm tab1
413 strokeWeight(3.5);
415 line(25, 60, 15, 75);

417 stroke(leftLeg); // left foot tab1
419 strokeWeight(3.5);
421 line(15, 93, 15, 87);

423 pushMatrix();
425 fill(0);
427 stroke(0);
429 ellipse(25, 45, 10, 10); // tab1 head
431 strokeWeight(3.5);
433 line(25, 45, 25, 75); // tab1 body
435 strokeWeight(0);
437 fill(255);
439 stroke(255);
441 strokeWeight(2);
443 line(28, 48, 31, 48);

445 void tab1change()
447 {
449 float comboA = (degrees(atan(sqrt(sq(
451 rightHipY1) + sq(rightHipX1)))/rightHipZ1))
453 ); //yaw RIGHT Leg TOP
455 float comboB = (degrees(atan(sqrt(sq(
457 rightHipY2) + sq(rightHipX2)))/rightHipZ2))
459 ); //yaw

461 if ((comboA > 0) && (comboB < 0)) {
463 comboC = (comboA);
465 }
467 if ((comboA < 0) && (comboB < 0)) {
469 comboC = (((-comboB)) + 90);
471 }
473 if ((comboA < 0) && (comboB > 0)) {
475 comboC = (-90 + -comboB);
477 }
479 if ((comboA > 0) && (comboB > 0)) {
481 comboC = (-comboA);
483 }

485 float comboD = (degrees(atan(sqrt(sq(
487 leftHipY1) + sq(leftHipX1)))/leftHipZ1)));
489 //yaw left Leg TOP
491 float comboE = (degrees(atan(sqrt(sq(
493 leftHipY2) + sq(leftHipX2)))/leftHipZ2)));
495 //yaw

497 if ((comboD > 0) && (comboE < 0)) {
501 comboF = (comboD);
503 }
505 if ((comboD < 0) && (comboE < 0)) {
507 comboF = (((-comboE)) + 90);
509 }
511 if ((comboD < 0) && (comboE > 0)) {
513 comboF = (-90 + -comboE);
515 }
517 if ((comboD > 0) && (comboE > 0)) {
519 comboF = (-comboD);
521 }

```

```

float comboG = (degrees(atan(sqrt(sq(
    leftLegY1) + sq(leftLegX1))/leftLegZ1)));
//yaw      left Leg BOTTOM
515
517
float comboH = (degrees(atan(sqrt(sq(
    leftLegY2) + sq(leftLegX2))/leftLegZ2)));
//yaw
519
521
if ((comboG > 0) && (comboH < 0)) {
    comboI = (comboG);
523
}
if ((comboG < 0) && (comboH < 0)) {
    comboI = (((-comboH)) + 90);
525
}
if ((comboG < 0) && (comboH > 0)) {
    comboI = (-90 + -comboH);
527
}
if ((comboG > 0) && (comboH > 0)) {
    comboI = (-comboG);
529
}

531
533
535
537
539
541
543
545
547
549
551
553
555
557
559
561
563
565
567
569
571
573
575
577
579

pushMatrix();
strokeWeight(10);
translate(350, 400);
rotate(radians(comboFff));
translate(0, 125); //-----
stroke(0, 0, 255); //Left Leg BOTTOM
rotate(radians(comboIf - comboFff));
line(0, 125, 0, 0);
// camera();
popMatrix();

pushMatrix();
strokeWeight(10);
translate(350, 400);
stroke(255, 0, 0); //Right Leg Top
rotate(radians(comboCf));
line(0, 125, 0, 0);
//camera();
popMatrix();

pushMatrix();
strokeWeight(10);
translate(350, 400);
rotate(radians(comboCf));
translate(0, 125); //-----
stroke(255, 0, 0); //RIGHT Leg BOTTOM
rotate(radians(comboLf - comboCf));
line(0, 125, 0, 0);
//camera();
popMatrix();

strokeWeight(2);
stroke(0);
fill(255);
ellipse(350, 100, 100, 100);
ellipse(375, 85, 15, 20);
line(375, 120, 400, 120);
stroke(255);
strokeWeight(10);
line(350, 100, 350, 400);
strokeWeight(0);
stroke(255);
fill(clickHighlight);
rect(0, 30, 50, 70, 0, 7, 7, 0);

println(comboC, comboL, comboI, comboF);
// println(leftHipY1, leftHipX1, leftHipZ1,
leftHipX2, leftHipX2, leftHipZ2);
}

void tab2character()
{
    stroke(rightLeg); //right Leg tab1
    strokeWeight(3.5);
    line(25, 150, 35, 159);

    stroke(rightLeg); //right Leg tab1
    strokeWeight(3.5);
    line(35, 159, 35, 168);
}

```

```

581 stroke(rightLeg); // right arm tab1
strokeWeight(3.5);
line(25, 135, 35, 150);

585 stroke(rightLeg); // right foot tab1
strokeWeight(3.5);
line(35, 168, 40, 168);

589 fill(0);
stroke(0);
ellipse(25, 120, 10, 10); // tab2 head
strokeWeight(3.5);
line(25, 120, 25, 150); // tab2 body
strokeWeight(0);
fill(255);
ellipse(27, 119, 3, 4);
camera();
fill(255);
stroke(255);
strokeWeight(2);
line(28, 123, 31, 123);
}

603 void tab2change()
{
  stroke(0);
  fill(255);
  strokeWeight(2);
  ellipse(350, 100, 100, 100);
  ellipse(375, 85, 15, 20);
  line(375, 120, 400, 120);
  stroke(255);
  strokeWeight(10);
  line(350, 100, 350, 400);
  strokeWeight(0);
  stroke(255);
  fill(clickHighlight);
  rect(0, 105, 50, 70, 0, 7, 7, 0);

  619 float comboA = (degrees(atan(sqrt(sq(
    rightHipY1) + sq(rightHipX1)))/rightHipZ1)))
    ); //yaw  RIGHT Leg TOP
  float comboB = (degrees(atan(sqrt(sq(
    rightHipY2) + sq(rightHipX2)))/rightHipZ2)))
    ); //yaw

  623 if ((comboA > 0) && (comboB < 0)) {
    comboC = (comboA);
  }
  627 if ((comboA < 0) && (comboB < 0)) {
    comboC = (((-comboB)) + 90);
  }
  631 if ((comboA < 0) && (comboB > 0)) {
    comboC = (-90 + -comboB);
  }
  633 if ((comboA > 0) && (comboB > 0)) {
    comboC = (-comboA);
  }

  637 float comboJ = (degrees(atan(sqrt(sq(
    rightLegY1) + sq(rightLegX1)))/rightLegZ1)))
    ); //yaw  RIGHT Leg BOTTOM
}

641 float comboK = (degrees(atan(sqrt(sq(
    rightLegY2) + sq(rightLegX2)))/rightLegZ2)))
    ); //yaw

643 if ((comboJ > 0) && (comboK < 0)) {
  comboL = (comboJ);
}
  645 if ((comboJ < 0) && (comboK < 0)) {
  comboL = (((-comboK)) + 90);
}
  647 if ((comboJ < 0) && (comboK > 0)) {
  comboL = (-90 + -comboK);
}
  649 if ((comboJ > 0) && (comboK > 0)) {
  comboL = (-comboJ);
}

653 strokeWeight(10);
translate(350, 400);
stroke(255, 0, 0); // Right Leg Top
rotate(radians(comboC));
line(0, 125, 0, 0);
camera();

657 strokeWeight(10);
translate(350, 400);
rotate(radians(comboC));
translate(0, 125); //-----
stroke(255, 0, 0); //RIGHT Leg BOTTOM
rotate(radians(comboL - comboC));
line(0, 125, 0, 0);
camera();

661 void tab3character()
{
  stroke(leftLeg); // left Leg tab1
  strokeWeight(3.5);
  line(25, 225, 26, 234);

  stroke(leftLeg); // left Leg tab1
  strokeWeight(3.5);
  line(26, 234, 18, 237);

  stroke(leftLeg); // left arm tab1
  strokeWeight(3.5);
  line(25, 210, 15, 225);

  stroke(leftLeg); // left foot tab1
  strokeWeight(3.5);
  line(15, 243, 15, 237);

  fill(0);
  stroke(0);
  ellipse(25, 195, 10, 10); // tab3 head
  strokeWeight(3.5);
  line(25, 195, 25, 225); // tab3 body
  strokeWeight(0);
  fill(255);
  ellipse(27, 194, 3, 4);
  camera();
  fill(255);
  stroke(255);
  strokeWeight(2);
}

```

```

703   line(28, 198, 31, 198);
705 }
706
707 void tab3change()
708 {
709   stroke(0);
710   fill(255);
711   strokeWeight(2);
712   ellipse(350, 100, 100, 100);
713   ellipse(375, 85, 15, 20);
714   line(375, 120, 400, 120);
715   stroke(255);
716   strokeWeight(10);
717   line(350, 100, 350, 400);
718   strokeWeight(0);
719   stroke(255);
720   fill(clickHighlight);
721   rect(0, 180, 50, 70, 0, 7, 7, 0);
722
723   float comboD = (degrees(atan(sqrt(sq(
724     leftHipY1) + sq(leftHipX1)))/leftHipZ1)));
725   //yaw left Leg TOP
726   float comboE = (degrees(atan(sqrt(sq(
727     leftHipY2) + sq(leftHipX2)))/leftHipZ2)));
728   //yaw
729
730   if ((comboD > 0) && (comboE < 0)) {
731     comboF = (comboD);
732   }
733   if ((comboD < 0) && (comboE < 0)) {
734     comboF = (((-comboE)) + 90);
735   }
736   if ((comboD < 0) && (comboE > 0)) {
737     comboF = (-90 + -comboE);
738   }
739   if ((comboD > 0) && (comboE > 0)) {
740     comboF = (-comboD);
741
742
743   float comboG = (degrees(atan(sqrt(sq(
744     leftLegY1) + sq(leftLegX1)))/leftLegZ1)));
745   //yaw left Leg BOTTOM
746   float comboH = (degrees(atan(sqrt(sq(
747     leftLegY2) + sq(leftLegX2)))/leftLegZ2)));
748   //yaw
749
750   if ((comboG > 0) && (comboH < 0)) {
751     comboI = (comboG);
752   }
753   if ((comboG < 0) && (comboH < 0)) {
754     comboI = (((-comboH)) + 90);
755   }
756   if ((comboG < 0) && (comboH > 0)) {
757     comboI = (-90 + -comboH);
758   }
759   if ((comboG > 0) && (comboH > 0)) {
760     comboI = (-comboG);
761
762   translate(350, 400);
763   strokeWeight(10);
764   stroke(0, 0, 255); //Left Leg Top  rotate(
765     radians(comboF));
766   rotate(radians(comboF));
767
768   line(0, 125, 0, 0);
769   camera();
770
771   strokeWeight(10);
772   translate(350, 400);
773   rotate(radians(comboF));
774   translate(0, 125); //-----
775   stroke(0, 0, 255); // Left Leg BOTTOM
776   rotate(radians(comboI - comboF));
777   line(0, 125, 0, 0);
778   camera();
779
780   void tab4character()
781   {
782     stroke(rightLeg); // right Leg tab1
783     strokeWeight(3.5);
784     line(25, 350, 35, 368);
785
786     stroke(rightLeg); // right arm tab1
787     strokeWeight(3.5);
788     line(25, 335, 35, 350);
789
790     stroke(rightLeg); // right foot tab1
791     strokeWeight(3.5);
792     line(35, 368, 40, 368);
793
794     stroke(leftLeg); // left Leg tab1
795     strokeWeight(3.5);
796     line(25, 350, 15, 368);
797
798     stroke(leftLeg); // left arm tab1
799     strokeWeight(3.5);
800     line(25, 335, 15, 350);
801
802     stroke(leftLeg); // left foot tab1
803     strokeWeight(3.5);
804     line(15, 368, 10, 368);
805
806
807     fill(0);
808     stroke(0);
809     ellipse(25, 320, 10, 10); //tab1 head
810     strokeWeight(3.5);
811     line(25, 320, 25, 350); //tab1 body
812     strokeWeight(0);
813     fill(255);
814     ellipse(27, 319, 3, 4);
815     ellipse(23, 319, 3, 4);
816
817     camera();
818     fill(255);
819     stroke(255);
820     strokeWeight(2);
821     line(24, 323, 26, 323);
822
823   void tab4change()
824   {
825     stroke(0);
826     strokeWeight(0);
827     fill(255);
828     fill(clickHighlight);
829     rect(0, 305, 50, 70, 0, 7, 7, 7, 0);
830
831     stroke(rightLeg); // right Leg tab1
832     strokeWeight(3.5);

```

```

827   line(25, 350, 35, 368);
828
829   stroke(rightLeg); // right arm tab1
830   strokeWeight(3.5);
831   line(25, 335, 35, 350);
832
833   stroke(rightLeg); // right foot tab1
834   strokeWeight(3.5);
835   line(35, 368, 40, 368);
836
837   stroke(leftLeg); // left Leg tab1
838   strokeWeight(3.5);
839   line(25, 350, 15, 368);
840
841   stroke(leftLeg); // left arm tab1
842   strokeWeight(3.5);
843   line(25, 335, 15, 350);
844
845   stroke(leftLeg); // left foot tab1
846   strokeWeight(3.5);
847   line(15, 368, 10, 368);
848
849   fill(0);
850   stroke(0);
851   ellipse(25, 320, 10, 10); // tab1 head
852   strokeWeight(3.5);
853   line(25, 320, 25, 350); // tab1 body
854   strokeWeight(0);
855   fill(255);
856   ellipse(27, 319, 3, 4);
857   ellipse(23, 319, 3, 4);
858
859   stroke(0);
860   fill(255);
861   strokeWeight(2);
862   ellipse(850, 100, 100, 100);
863   ellipse(875, 85, 15, 20);
864   line(875, 120, 905, 120);
865   stroke(255);
866   strokeWeight(10);
867   line(850, 130, 850, 400);
868
869   // pushMatrix();
870   // fill(0);
871   // stroke(0);
872   // ellipse(25, 45, 10, 10); // tab1 head
873   // strokeWeight(3.5);
874   // line(25, 45, 25, 75); // tab1 body
875   // strokeWeight(0);
876   // fill(255);
877   // ellipse(27, 44, 3, 4);
878   // popMatrix();
879   // camera();
880   // fill(255);
881   // stroke(255);
882   // strokeWeight(2);
883   // line(28, 48, 31, 48);
884
885   float comboA = (degrees(atan(sqrt(sq(
886     rightHipY1) + sq(rightHipX1)))/rightHipZ1))
887   ); //yaw  RIGHT Leg TOP
888   float comboB = (degrees(atan(sqrt(sq(
889     rightHipY2) + sq(rightHipX2)))/rightHipZ2))
890   ); //yaw
891
892   if ((comboA > 0) && (comboB < 0)) {
893     comboC = (comboA);
894   }
895   if ((comboA < 0) && (comboB < 0)) {
896     comboC = (((-comboB)) + 90);
897   }
898   if ((comboA < 0) && (comboB > 0)) {
899     comboC = (-90 + -comboB);
900   }
901   if ((comboA > 0) && (comboB > 0)) {
902     comboC = (-comboA);
903   }
904
905   float comboD = (degrees(atan(sqrt(sq(
906     leftHipY1) + sq(leftHipX1)))/leftHipZ1));
907   //yaw  left Leg TOP
908   float comboE = (degrees(atan(sqrt(sq(
909     leftHipY2) + sq(leftHipX2)))/leftHipZ2));
910   //yaw
911
912   if ((comboD > 0) && (comboE < 0)) {
913     comboF = (comboD);
914   }
915   if ((comboD < 0) && (comboE < 0)) {
916     comboF = (((-comboE)) + 90);
917   }
918   if ((comboD < 0) && (comboE > 0)) {
919     comboF = (-90 + -comboE);
920   }
921   if ((comboD > 0) && (comboE > 0)) {
922     comboF = (-comboD);
923   }
924
925   float comboG = (degrees(atan(sqrt(sq(
926     leftLegY1) + sq(leftLegX1)))/leftLegZ1));
927   //yaw  left Leg BOTTOM
928   float comboH = (degrees(atan(sqrt(sq(
929     leftLegY2) + sq(leftLegX2)))/leftLegZ2));
930   //yaw
931
932   if ((comboG > 0) && (comboH < 0)) {
933     comboI = (comboG);
934   }
935   if ((comboG < 0) && (comboH < 0)) {
936     comboI = (((-comboH)) + 90);
937   }
938   if ((comboG < 0) && (comboH > 0)) {
939     comboI = (-90 + -comboH);
940   }
941   if ((comboG > 0) && (comboH > 0)) {
942     comboI = (-comboG);
943   }
944
945   float comboJ = (degrees(atan(sqrt(sq(
946     rightLegY1) + sq(rightLegX1)))/rightLegZ1))
947   ); //yaw  RIGHT Leg BOTTOM
948   float comboK = (degrees(atan(sqrt(sq(
949     rightLegY2) + sq(rightLegX2)))/rightLegZ2))
950   ); //yaw

```

```

943 if ((comboJ > 0) && (comboK < 0)) {
944   comboL = (comboJ);
945 }
946 if ((comboJ < 0) && (comboK < 0)) {
947   comboL = (((-comboK)) + 90);
948 }
949 if ((comboJ < 0) && (comboK > 0)) {
950   comboL = (-90 + -comboK);
951 }
952 if ((comboJ > 0) && (comboK > 0)) {
953   comboL = (-comboJ);
954 }

955
956
957 if (abs(comboLf - comboL) > tol) {
958   comboLf = comboL;
959 }
960 if (abs(comboIf - comboI) > tol) {
961   comboIf = comboI;
962 }
963 if (abs(comboCf - comboC) > tol) {
964   comboCf = comboC;
965 }
966 if (abs(comboFff - comboF) > tol) {
967   comboFff = comboF;
968 }

969 //COLORS
970
971 if ((comboF > -100 && comboF < -70) && (
972   comboC > -100 && comboC < -70) && (comboL >
973   -20 && comboL < 60) && (comboI > -20 &&
974   comboI < 60)) {
975   colorR = 0;
976   colorG = 255;
977   colorB = 0;
978 } else if ((comboF > -80 && comboF < -30) &&
979   (comboC > -80 && comboC < -30) && (comboL >
980   -20 && comboL < 60) && (comboI > -20 &&
981   comboI < 60)) {
982   colorR = 255;
983   colorG = 255;
984   colorB = 0;
985 } else{
986   colorR = 255;
987   colorG = 0;
988   colorB = 0;
989 }

990 // println(comboC, comboL, comboI, comboF);
991 pushMatrix();
992 translate(850, 400);
993 strokeWeight(10);
994 stroke(colorR, colorG, colorB); // Left Leg
995   Top rotate(radians(comboF));
996   rotate(radians(comboFff));
997   line(0, 125, 0, 0);
998   popMatrix();
999   // camera();

1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187

```

```

1063   line(25, 410, 35, 425);           1119
1065   stroke(rightLeg); //right foot tab1 1121
1067   strokeWeight(3.5);                1123
1069   line(35, 443, 40, 443);          1125
1071   fill(0);                         1127
1073   stroke(0);                      1129
1075   ellipse(25, 395, 10, 10); //tab2 head 1131
1077   strokeWeight(3.5);                1133
1079   line(25, 395, 25, 425); //tab2 body 1135
1081   strokeWeight(0);                 1137
1083   fill(255);                      1139
1085   ellipse(27, 394, 3, 4);          1141
1087   ellipse(23, 394, 3, 4);          1143
1089
1091   camera();                        1145
1093   fill(255);                      1147
1095   stroke(255);                   1149
1097   strokeWeight(2);                 1151
1099   line(24, 398, 26, 398);          1153
1101
1103 void tab5change() {               1155
1105   stroke(0);                      1157
1107   strokeWeight(0);                 1159
1109   fill(255);                      1161
1111   fill(clickHighlight);           1163
1113   rect(0, 380, 50, 70, 0, 7, 7, 0); 1165
1115
1117   stroke(0);                      1167
1119   fill(255);                      1169
1121   strokeWeight(2);                 1171
1123   ellipse(850, 100, 100, 100);    1173
1125   ellipse(870, 85, 15, 20);       1175
1127   ellipse(830, 85, 15, 20);       1177
1129   line(835, 120, 865, 120);      1179
1131   stroke(255);                   1181
1133   strokeWeight(10);                1183
1135   line(850, 130, 850, 400);      1185
1137
1139 void tab6character() {           1187
1141   stroke(leftLeg); //left Leg tab3 1189
1143   strokeWeight(3.5);                1191
1145   line(25, 500, 15, 518);          1193
1147
1149
1151
1153
1155
1157
1159
1161
1163
1165
1167
1169
1171
1173
1175
1177
1179
1181
1183
1185
1187
1189
1191
1193
1195
1197
1199
1201
1203
1205
1207
1209
1211
1213
1215
1217
1219
1221
1223
1225
1227
1229
1231
1233
1235
1237
1239
1241
1243
1245
1247
1249
1251
1253
1255
1257
1259
1261
1263
1265
1267
1269
1271
1273
1275
1277
1279
1281
1283
1285
1287
1289
1291
1293
1295
1297
1299
1301
1303
1305
1307
1309
1311
1313
1315
1317
1319
1321
1323
1325
1327
1329
1331
1333
1335
1337
1339
1341
1343
1345
1347
1349
1351
1353
1355
1357
1359
1361
1363
1365
1367
1369
1371
1373
1375
1377
1379
1381
1383
1385
1387
1389
1391
1393
1395
1397
1399
1401
1403
1405
1407
1409
1411
1413
1415
1417
1419
1421
1423
1425
1427
1429
1431
1433
1435
1437
1439
1441
1443
1445
1447
1449
1451
1453
1455
1457
1459
1461
1463
1465
1467
1469
1471
1473
1475
1477
1479
1481
1483
1485
1487
1489
1491
1493
1495
1497
1499
1501
1503
1505
1507
1509
1511
1513
1515
1517
1519
1521
1523
1525
1527
1529
1531
1533
1535
1537
1539
1541
1543
1545
1547
1549
1551
1553
1555
1557
1559
1561
1563
1565
1567
1569
1571
1573
1575
1577
1579
1581
1583
1585
1587
1589
1591
1593
1595
1597
1599
1601
1603
1605
1607
1609
1611
1613
1615
1617
1619
1621
1623
1625
1627
1629
1631
1633
1635
1637
1639
1641
1643
1645
1647
1649
1651
1653
1655
1657
1659
1661
1663
1665
1667
1669
1671
1673
1675
1677
1679
1681
1683
1685
1687
1689
1691
1693
1695
1697
1699
1701
1703
1705
1707
1709
1711
1713
1715
1717
1719
1721
1723
1725
1727
1729
1731
1733
1735
1737
1739
1741
1743
1745
1747
1749
1751
1753
1755
1757
1759
1761
1763
1765
1767
1769
1771
1773
1775
1777
1779
1781
1783
1785
1787
1789
1791
1793
1795
1797
1799
1801
1803
1805
1807
1809
1811
1813
1815
1817
1819
1821
1823
1825
1827
1829
1831
1833
1835
1837
1839
1841
1843
1845
1847
1849
1851
1853
1855
1857
1859
1861
1863
1865
1867
1869
1871
1873
1875
1877
1879
1881
1883
1885
1887
1889
1891
1893
1895
1897
1899
1901
1903
1905
1907
1909
1911
1913
1915
1917
1919
1921
1923
1925
1927
1929
1931
1933
1935
1937
1939
1941
1943
1945
1947
1949
1951
1953
1955
1957
1959
1961
1963
1965
1967
1969
1971
1973
1975
1977
1979
1981
1983
1985
1987
1989
1991
1993
1995
1997
1999
2001
2003
2005
2007
2009
2011
2013
2015
2017
2019
2021
2023
2025
2027
2029
2031
2033
2035
2037
2039
2041
2043
2045
2047
2049
2051
2053
2055
2057
2059
2061
2063
2065
2067
2069
2071
2073
2075
2077
2079
2081
2083
2085
2087
2089
2091
2093
2095
2097
2099
2101
2103
2105
2107
2109
2111
2113
2115
2117
2119
2121
2123
2125
2127
2129
2131
2133
2135
2137
2139
2141
2143
2145
2147
2149
2151
2153
2155
2157
2159
2161
2163
2165
2167
2169
2171
2173
2175
2177
2179
2181
2183
2185
2187
2189
2191
2193
2195
2197
2199
2201
2203
2205
2207
2209
2211
2213
2215
2217
2219
2221
2223
2225
2227
2229
2231
2233
2235
2237
2239
2241
2243
2245
2247
2249
2251
2253
2255
2257
2259
2261
2263
2265
2267
2269
2271
2273
2275
2277
2279
2281
2283
2285
2287
2289
2291
2293
2295
2297
2299
2301
2303
2305
2307
2309
2311
2313
2315
2317
2319
2321
2323
2325
2327
2329
2331
2333
2335
2337
2339
2341
2343
2345
2347
2349
2351
2353
2355
2357
2359
2361
2363
2365
2367
2369
2371
2373
2375
2377
2379
2381
2383
2385
2387
2389
2391
2393
2395
2397
2399
2401
2403
2405
2407
2409
2411
2413
2415
2417
2419
2421
2423
2425
2427
2429
2431
2433
2435
2437
2439
2441
2443
2445
2447
2449
2451
2453
2455
2457
2459
2461
2463
2465
2467
2469
2471
2473
2475
2477
2479
2481
2483
2485
2487
2489
2491
2493
2495
2497
2499
2501
2503
2505
2507
2509
2511
2513
2515
2517
2519
2521
2523
2525
2527
2529
2531
2533
2535
2537
2539
2541
2543
2545
2547
2549
2551
2553
2555
2557
2559
2561
2563
2565
2567
2569
2571
2573
2575
2577
2579
2581
2583
2585
2587
2589
2591
2593
2595
2597
2599
2601
2603
2605
2607
2609
2611
2613
2615
2617
2619
2621
2623
2625
2627
2629
2631
2633
2635
2637
2639
2641
2643
2645
2647
2649
2651
2653
2655
2657
2659
2661
2663
2665
2667
2669
2671
2673
2675
2677
2679
2681
2683
2685
2687
2689
2691
2693
2695
2697
2699
2701
2703
2705
2707
2709
2711
2713
2715
2717
2719
2721
2723
2725
2727
2729
2731
2733
2735
2737
2739
2741
2743
2745
2747
2749
2751
2753
2755
2757
2759
2761
2763
2765
2767
2769
2771
2773
2775
2777
2779
2781
2783
2785
2787
2789
2791
2793
2795
2797
2799
2801
2803
2805
2807
2809
2811
2813
2815
2817
2819
2821
2823
2825
2827
2829
2831
2833
2835
2837
2839
2841
2843
2845
2847
2849
2851
2853
2855
2857
2859
2861
2863
2865
2867
2869
2871
2873
2875
2877
2879
2881
2883
2885
2887
2889
2891
2893
2895
2897
2899
2901
2903
2905
2907
2909
2911
2913
2915
2917
2919
2921
2923
2925
2927
2929
2931
2933
2935
2937
2939
2941
2943
2945
2947
2949
2951
2953
2955
2957
2959
2961
2963
2965
2967
2969
2971
2973
2975
2977
2979
2981
2983
2985
2987
2989
2991
2993
2995
2997
2999
3001
3003
3005
3007
3009
3011
3013
3015
3017
3019
3021
3023
3025
3027
3029
3031
3033
3035
3037
3039
3041
3043
3045
3047
3049
3051
3053
3055
3057
3059
3061
3063
3065
3067
3069
3071
3073
3075
3077
3079
3081
3083
3085
3087
3089
3091
3093
3095
3097
3099
3101
3103
3105
3107
3109
3111
3113
3115
3117
3119
3121
3123
3125
3127
3129
3131
3133
3135
3137
3139
3141
3143
3145
3147
3149
3151
3153
3155
3157
3159
3161
3163
3165
3167
3169
3171
3173
3175
3177
3179
3181
3183
3185
3187
3189
3191
3193
3195
3197
3199
3201
3203
3205
3207
3209
3211
3213
3215
3217
3219
3221
3223
3225
3227
3229
3231
3233
3235
3237
3239
3241
3243
3245
3247
3249
3251
3253
3255
3257
3259
3261
3263
3265
3267
3269
3271
3273
3275
3277
3279
3281
3283
3285
3287
3289
3291
3293
3295
3297
3299
3301
3303
3305
3307
3309
3311
3313
3315
3317
3319
3321
3323
3325
3327
3329
3331
3333
3335
3337
3339
3341
3343
3345
3347
3349
3351
3353
3355
3357
3359
3361
3363
3365
3367
3369
3371
3373
3375
3377
3379
3381
3383
3385
3387
3389
3391
3393
3395
3397
3399
3401
3403
3405
3407
3409
3411
3413
3415
3417
3419
3421
3423
3425
3427
3429
3431
3433
3435
3437
3439
3441
3443
3445
3447
3449
3451
3453
3455
3457
3459
3461
3463
3465
3467
3469
3471
3473
3475
3477
3479
3481
3483
3485
3487
3489
3491
3493
3495
3497
3499
3501
3503
3505
3507
3509
3511
3513
3515
3517
3519
3521
3523
3525
3527
3529
3531
3533
3535
3537
3539
3541
3543
3545
3547
3549
3551
3553
3555
3557
3559
3561
3563
3565
3567
3569
3571
3573
3575
3577
3579
3581
3583
3585
3587
3589
3591
3593
3595
3597
3599
3601
3603
3605
3607
3609
3611
3613
3615
3617
3619
3621
3623
3625
3627
3629
3631
3633
3635
3637
3639
3641
3643
3645
3647
3649
3651
3653
3655
3657
3659
3661
3663
3665
3667
3669
3671
3673
3675
3677
3679
3681
3683
3685
3687
3689
3691
3693
3695
3697
3699
3701
3703
3705
3707
3709
3711
3713
3715
3717
3719
3721
3723
3725
3727
3729
3731
3733
3735
3737
3739
3741
3743
3745
3747
3749
3751
3753
3755
3757
3759
3761
3763
3765
3767
3769
3771
3773
3775
3777
3779
3781
3783
3785
3787
3789
3791
3793
3795
3797
3799
3801
3803
3805
3807
3809
3811
3813
3815
3817
3819
3821
3823
3825
3827
3829
3831
3833
3835
3837
3839
3841
3843
3845
3847
3849
3851
3853
3855
3857
3859
3861
3863
3865
3867
3869
3871
3873
3875
3877
3879
3881
3883
3885
3887
3889
3891
3893
3895
3897
3899
3901
3903
3905
3907
3909
3911
3913
3915
3917
3919
3921
3923
3925
3927
3929
3931
3933
3935
3937
3939
3941
3943
3945
3947
3949
3951
3953
3955
3957
3959
3961
3963
3965
3967
3969
3971
3973
3975
3977
3979
3981
3983
3985
3987
3989
3991
3993
3995
3997
3999
4001
4003
4005
4007
4009
4011
4013
4015
4017
4019
4021
4023
4025
4027
4029
4031
4033
4035
4037
4039
4041
4043
4045
4047
4049
4051
4053
4055
4057
4059
4061
4063
4065
4067
4069
4071
4073
4075
4077
4079
4081
4083
4085
4087
4089
4091
4093
4095
4097
4099
4101
4103
4105
4107
4109
4111
4113
4115
4117
4119
4121
4123
4125
4127
4129
4131
4133
4135
4137
4139
4141
4143
4145
4147
4149
4151
4153
4155
4157
4159
4161
4163
4165
4167
4169
4171
4173
4175
4177
4179
4181
4183
4185
4187
4189
4191
4193
4195
4197
4199
4201
4203
4205
4207
4209
4211
4213
4215
4217
4219
4221
4223
4225
4227
4229
4231
4233
4235
4237
4239
4241
4243
4245
4247
4249
4251
4253
4255
4257
4259
4261
4263
4265
4267
4269
4271
4273
4275
4277
4279
4281
4283
4285
4287
4289
4291
4293
4295
4297
4299
4301
4303
4305
4307
4309
4311
4313
4315
4317
4319
4321
4323
4325
4327
4329
4331
4333
4335
4337
4339
4341
4343
4345
4347
4349
4351
4353
4355
4357
4359
4361
4363
4365
4367
4369
4371
4373
4375
4377
4379
4381
4383
4385
4387
4389
4391
4393
4395
4397
4399
4401
4403
4405
4407
4409
4411
4413
4415
4417
4419
4421
4423
4425
4427
4429
4431
4433
4435
4437
4439
4441
4443
4445
4447
4449
4451
4453
4455
4457
4459
4461
4463
4465
4467
4469
4471
4473
4475
4477
4479
4481
4483
4485
4487
4489
4491
4493
4495
4497
4499
4501
4503
4505
4507
4509
4511
4513
4515
4517
4519
4521
4523
4525
4527
4529
4531
4533
4535
4537
4539
4541
4543
4545
4547
4549
4551
4553
4555
4557
4559
4561
4563
4565
4567
4569
4571
4573
4575
4577
4579
4581
4583
4585
4587
4589
4591
4593
4595
4597
4599
4601
4603
4605
4607
4609
4611
4613
4615
4617
4619
4621
4623
4625
4627
4629
4631
4633
4635
4637
4639
4641
4643
4645
4647
4649
4651
4653
4655
4657
4659
4661
4663
4665
4667
4669
4671
4673
4675
4677
4679
4681
4683
4685
4687
4689
4691
4693
4695
4697
4699
4701
4703
4705
4707
4709
4711
4713
4715
4717
4719
4721
4723
4725
4727
4729
4731
4733
4735
4737
4739
4741
4743
4745
4747
4749
4751
4753
4755
4757
4759
4761
4763
4765
4767
4769
4771
4773
4775
4777
4779
4781
4783
4785
4787
4789
4791
4793
4795
4797
4799
4801
4803
4805
4807
4809
4811
4813
4815
4817
4819
4821
4823
4825
4827
4829
4831
4833
4835
4837
4839
4841
4843
4845
4847
4849
4851
4853
4855
4857
4859
4861
4863
4865
4867
4869
4871
4873
4875
4877
4879
4881
4883
4885
4887
4889
4891
4893
4895
4897
4899
4901
4903
4905
4907
4909
4911
4913
4915
4917
4919
4921
4923
4925
4927
4929
4931
4933
4935
4937
4939
4941
4943
4945
4947
4949
4951
4953
4955
4957
4959
4961
4963
4965
4967
4969
4971
4973
4975
4977
4979
4981
4983
4985
4987
4989
4991
4993
4995
4997
4999
5001
5003
5005
5007
5009
5011
5013
5015
5017
5019
5021
5023
5025
5027
5029
5031
5033
5035
5037
5039
5041
5043
5045
5047
5049
5051
5053
5055
5057
5059
5061
5063
5065
5067
5069
5071
5073
5075
5077
5079
5081
5083
5085
5087
5089
5091
5093
5095
5097
5099
5101
5103
5105
5107
5109
5111
5113
5115
5117
5119
5121
5123
5125
5127
5129
5131
5133
5135
5137
5139
5141
5143
5145
5147
5149
5151
5153
5155
5157
5159
5161
5163
5165
5167
5169
5171
5173
5175
5177
5179
5181
5183
5185
5187
5189
5191
5193
5195
5197
5199
5201
5203
5205
5207
5209
5211
5213
5215
5217
5219
5221
5223
5225
5227
5229
5231
5233
5235
5237
5239
5241
5243
5245
5247
5249
5251
5253
5255
5257
5259
5261
5263
5265
5267
5269
5271
5273
5275
5277
5279
5281
5283
5285
5287
5289
5291
5293
5295
5297
5299
5301
5303
5305
5307
5309
5311
5313
5315
5317
5319
5321
5323
5325
5327
5329
5331
5333
5335
5337
5339
5341
5343
5345
5347
5349
5351
5353
5355
5357
5359
5361
5363
5365
5367
5369
5371
5373
5375
5377
5379
5381
5383
5385
5387
5389
5391
5393
5395
5397
5399
5401
5403
5405
5407
5409
5411
5413
5415
5417
5419
5421
5423
5425
5427
5429
5431
5433
5435
5437
5439
5441
5443
5445
5447
5449
5451
5453
5455
5457
5459
5461
5463
5465
5467
5469
5471
5473
5475
5477
5479
5481
5483
5485
5487
5489
5491
5493
54
```

```

1195     text( "Angle", 17, 787);
1197 void anglesRight()
1198 {
1199     stroke(0);
1200     fill(clickHighlight);
1201     rect(950, 5, 225, 60, 7, 7, 7, 7);
1202     textSize(25);
1203     fill(rightLeg);
1204     text("Right Hip", 960, 30);
1205     text("Right Knee", 960, 60);
1207
1208     fill(clickHighlight);
1209     rect(0, 740, 120, 60, 7, 7, 0, 0);
1210     fill(0);
1211     textSize(12);
1212     text("Right Click For", 17, 760);
1213     textSize(30);
1214     text( "Angle", 17, 787);
1215 }
1217 void anglesLeft()
1218 {
1219     stroke(0);
1220     fill(clickHighlight);
1221     rect(950, 5, 225, 60, 7, 7, 7, 7);
1222     textSize(25);
1223     fill(leftLeg);
1224     text("Left Hip", 960, 30);
1225     text("Left Knee", 960, 60);
1227
1228     fill(clickHighlight);
1229     rect(0, 740, 120, 60, 7, 7, 0, 0);
1230     fill(0);
1231     textSize(12);
1232     text("Right Click For", 17, 760);
1233     textSize(30);
1234     text( "Angle", 17, 787);
1235 }
1237 void anglesSide()
1238 {
1239     stroke(0);
1240     fill(clickHighlight);
1241     rect(950, 5, 225, 120, 7, 7, 7, 7); //765,
1242     5
1243     textSize(25);
1244     fill(rightLeg);
1245     text("Right Hip", 960, 30);
1246     text("Right Knee", 960, 60);
1247     fill(leftLeg);
1248     text("Left Hip", 960, 90);
1249     text("Left Knee", 960, 120);
1250
1251     fill(clickHighlight);
1252     rect(0, 740, 120, 60, 7, 7, 0, 0);
1253     fill(0);
1254     textSize(12);
1255     text("Right Click For", 17, 760);
1256     textSize(30);
1257     text( "Angle", 17, 787);
1258 }
1259 void anglesRightSide()
1260 {
1261     stroke(0);
1262     fill(clickHighlight);
1263     rect(950, 70, 225, 60, 7, 7, 7, 7);
1264     textSize(25);
1265     fill(rightLeg);
1266     text("Right Hip", 960, 95);
1267     text("Right Knee", 960, 125);
1268
1269     fill(clickHighlight);
1270     rect(0, 740, 120, 60, 7, 7, 0, 0);
1271     fill(0);
1272     textSize(12);
1273     text("Right Click For", 17, 760);
1274     textSize(30);
1275     text( "Angle", 17, 787);
1276 }
1277 void anglesLeftSide()
1278 {
1279     stroke(0);
1280     fill(clickHighlight);
1281     rect(950, 70, 225, 60, 7, 7, 7, 7);
1282     textSize(25);
1283     fill(leftLeg);
1284     text("Left Hip", 960, 95);
1285     text("Left Knee", 960, 125);
1286
1287     fill(clickHighlight);
1288     rect(0, 740, 120, 60, 7, 7, 0, 0);
1289     fill(0);
1290     textSize(12);
1291     text("Right Click For", 17, 760);
1292     textSize(30);
1293     text( "Angle", 17, 787);
1294 }
1295 void blackAngles()
1296 {
1297     fill(0);
1298     strokeWeight(0);
1299     rect(940, 0, 300, 200);
1300 }
1301 void serialEvent(Serial port) {
1302     String inData = port.readStringUntil('\n'); // cut off white space (carriage return)
1303     inData = trim(inData);
1304     // println(inData);
1305     // if (frameCount % 4 == 0) {
1306
1307         if (inData.charAt(0) == 'S') { //S
1308             inData = inData.substring(1);
1309             rightHipX1 = float(inData);
1310         }
1311         if (inData.charAt(0) == 'T') { //T
1312             inData = inData.substring(1);
1313             rightHipY1 = float(inData);
1314         }
1315         if (inData.charAt(0) == 'U') { //U
1316             inData = inData.substring(1);
1317             rightHipZ1 = float(inData);
1318         }
1319         if (inData.charAt(0) == 'V') { //V
1320             inData = inData.substring(1);
1321             rightHipX2 = float(inData);
1322         }
1323     }

```

```

1325   if (inData.charAt(0) == 'W') {           //W      1391
1326     inData = inData.substring(1);
1327     rightHipY2 = float(inData);
1328   }
1329   if (inData.charAt(0) == 'X') {           //X      1393
1330     inData = inData.substring(1);
1331     rightHipZ2 = float(inData);
1332   }
1333   //if (frameCount % 4 == 1){
1334   if (inData.charAt(0) == 'M') {           1395
1335     inData = inData.substring(1);
1336     rightLegX1 = float(inData);
1337   }
1338   if (inData.charAt(0) == 'N') {           1397
1339     inData = inData.substring(1);
1340     rightLegY1 = float(inData);
1341   }
1342   if (inData.charAt(0) == 'O') {           1399
1343     inData = inData.substring(1);
1344     rightLegZ1 = float(inData);
1345   }
1346   if (inData.charAt(0) == 'P') {           1401
1347     inData = inData.substring(1);
1348     rightLegX2 = float(inData);
1349   }
1350   if (inData.charAt(0) == 'Q') {           1403
1351     inData = inData.substring(1);
1352     rightLegY2 = float(inData);
1353   }
1354   if (inData.charAt(0) == 'R') {           1405
1355     inData = inData.substring(1);
1356     rightLegZ2 = float(inData);
1357   }
1358   //}
1359   //if (frameCount % 4 == 2) {
1360   if (inData.charAt(0) == 'D') {           1407
1361     inData = inData.substring(1);
1362     leftHipX1 = float(inData);
1363   }
1364   if (inData.charAt(0) == 'E') {           1409
1365     inData = inData.substring(1);
1366     leftHipY1 = float(inData);
1367   }
1368   if (inData.charAt(0) == 'F') {           1411
1369     inData = inData.substring(1);
1370     leftHipZ1 = float(inData);
1371   }
1372   if (inData.charAt(0) == 'A') {           1411
1373     inData = inData.substring(1);
1374     leftHipX2 = float(inData);
1375   }
1376   if (inData.charAt(0) == 'B') {           1411
1377     inData = inData.substring(1);
1378     leftHipY2 = float(inData);
1379   }
1380   if (inData.charAt(0) == 'C') {           1411
1381     inData = inData.substring(1);
1382     leftHipZ2 = float(inData);
1383   }
1384   //}
1385   //if (frameCount % 4 == 3){
1386   if (inData.charAt(0) == 'G') {           1411
1387     inData = inData.substring(1);
1388     leftLegX1 = float(inData);
1389   }

```

3. Discussion

One of LegSense's primary purposes is to aid both novice and experienced powerlifters. Powerlifting does not only require strength and skill, but also technique. When lifting, there is a high risk of injury. Technique is crucial to avoid getting hurt. However, with practice and proper movement, one can minimize this risk.

In terms of weight lifting, if LegSense were available for those who are not given the opportunity to learn the proper technique they could now have access to a personal coach. If someone does not have the means to afford or have access to proper training, they can use LegSense to learn the fundamentals of the lift prior to using the weights. By learning the movement prior to using weight, the user can lower their risk of injury.

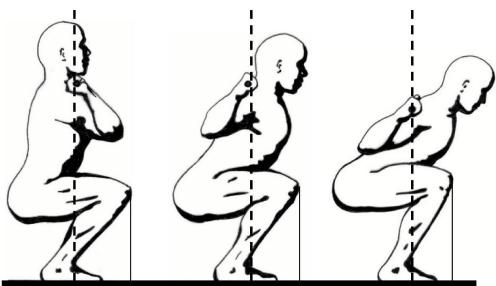


FIG. 13. Squat Variations

Additionally in terms of weight lifting, if one were to wear LegSense while they perform a lift, they would be able to analyze the exact motion of the lift to increase efficiency and minimize risk of injury. As an example, if someone were squatting and was not achieving proper depth in the motion, they could analyze the lift and adjust their depth on later repetitions and lifts.

Another main purpose of LegSense is to aid in physical therapy. During physical therapy, a doctor or specialist looks for key identifiers to determine the progression of a patient. They may ask a patient, "How much does this motion hurt?" or "How far can you extend your leg?" but if this information was numerical rather than verbal the doctor's perception of the injury would likely be more accurate. Rather than asking the patient about their flexibility and range of motion, the doctor can analyze the angles provided by LegSense. Using this data, the doctor could even take this a step further by storing the angles during a specific exercise during each session of physical therapy, and compare the patient's progression from session to session. Furthermore, the doctor can then plot these angles on a graph and provide visual evidence of the patient's increased or diminished range of motion from session to session.

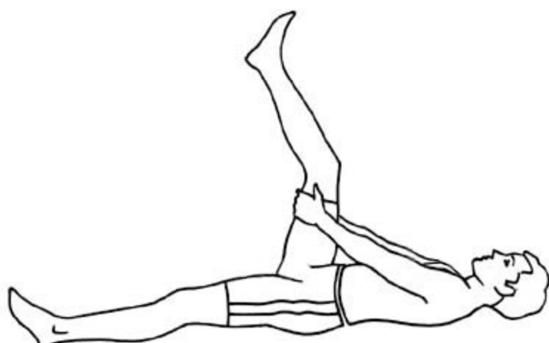


FIG. 14. Hamstring Stretch

For example, a common injury for dozens of sports and athletes is pulling or tearing a hamstring. In rehab, one stretch that can determine the progress the patient has made in their flexibility and recovery is the lying down hamstring stretch, as shown in figure 14. Assuming the patient is wearing LegSense, a doctor could store the angle of the injured leg relative to the ground or core of the patient to determine their flexibility and range of motion. Once the angle is stored, the doctor could create charts such as tables and graphs to accurately analyze the user's progression.

This practical application of LegSense in physical therapy extends just beyond hamstring injuries. There are

many motions doctors analyze before concluding on a patient's condition. If a patient were to wear LegSense for the entire duration of a session the doctor could, after the session, go back and rewatch the motions of the user during a specific exercise and analyze that data and even potentially compare it to the same exercise a previous session. Such as motions in figure 15.

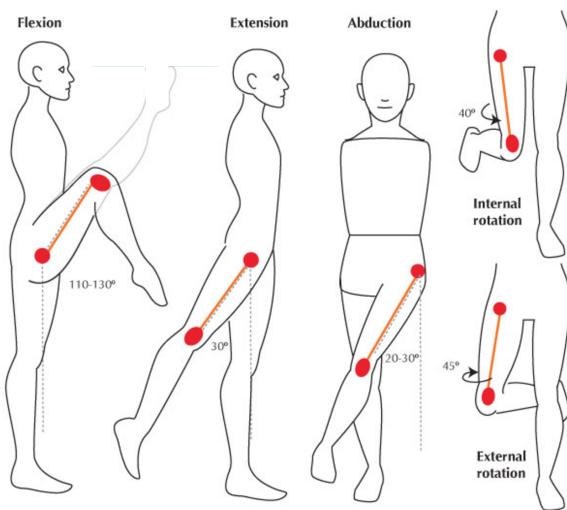


FIG. 15. Physical Therapy Motions

4. Conclusion

In conclusion there are a lot of corrections and progress that can be made if the project were taken to the next level. LegSense would include a much more sophisticated user interface for weight lifting, preprogrammed with dozens of various weightlifting motions. Furthermore, the product itself would improve with the addition of back sensors, in addition to other limbs such as arms and forearms.

Additionally, LegSense's interface would include a separate section dedicated to rehabilitation in which a user could identify a specific exercise they were doing and track their range of motion on a session to session basis.

The product itself and interface have come a long way since they were first created, yet still have a ways to go before it is put on the market.

Bibliography

- [1] Paul Beznis. London 2012 olympics triple jump men slowmo. <https://www.youtube.com/watch?v=xS6NTz3qXNw>, July 2014. London Olympics Triple Jump.

- [2] Adafruit Industries. Adafruit industries ez-link bluefruit shield. <https://learn.adafruit.com/adafruit-bluefruit-ez-link-shield/programming-with-arduino?view=all>. All of the information pertaining to the EZ-Link Bluefruit Shield by Adafruit Industries.
- [3] Adafruit Industries. Adafruit industries tca select device. <https://learn.adafruit.com/adafruit-tca9548a-1-to-8-i2c-multiplexer-breakout/overview>. All of the information pertaining to the TCA Select device by Adafruit Industries.
- [4] Mark Lelinwalla. How usa swimming trains with bmw's motion tracking system for the 2016 olympics. <https://techcrunch.com/2016/07/18/how-usa-swimming-trains-with-bmw-s-motion-tracking-system-for-the-2016-olympics/>, July 2016. BMW working with motion tracking and U.S.A. Olympic swimmers.
- [5] SparkFun. Lsm9ds1 information from sparkfun. <https://www.sparkfun.com/products/13284>. All of the information pertaining to the LSM9DS1 Breakout Board by SparkFun.