// define the pins im using

#define x2 A0

#define y1 A1

#define x1 A2

#define y2 A3

// add libraries

#include "Mouse.h"

// initialize variables

int x = 0;

int y = 0;

int z1 = 0;

int z2 = 0;

int pressed = 0;

int posbuffer[8];

int posbufferadj[8];

int startpos[2];

int changebuffer[8];

int xadj = 0;

int yadj = 0;

int badjumpsize = 10;

int tomove[2];

int sens = 3;

void setup()

{

// setup whatever i need to setup

Serial.begin(9600);

Mouse.begin();

}

void loop()

{

// grab the x position

pinMode(y1,INPUT);

pinMode(y2,INPUT);

digitalWrite(y2,LOW);

pinMode(x1,OUTPUT);

digitalWrite(x1,HIGH);

pinMode(x2,OUTPUT);

digitalWrite(x2,LOW);

// y = analogRead(y1);

y = analogRead(y1) \* -1;

// grab the y position

pinMode(x1,INPUT);

pinMode(x2,INPUT);

digitalWrite(x2,LOW);

pinMode(y1,OUTPUT);

digitalWrite(y1,HIGH);

pinMode(y2,OUTPUT);

digitalWrite(y2,LOW);

// x = analogRead(x1) \* -1;

x = analogRead(x1);

// find pressure

pinMode(y1,OUTPUT);

digitalWrite(y1,HIGH);

pinMode(x2,OUTPUT);

digitalWrite(x2,LOW);

pinMode(x1,INPUT);

pinMode(y2,INPUT);

z1 = analogRead(x1);

z2 = analogRead(y2);

if (z2 != 1023)

{

pressed = true;

}

else

{

pressed = false;

x = 0;

y = 0;

}

// continue the buffer

posbuffer[7] = posbuffer[5];

posbuffer[6] = posbuffer[4];

posbuffer[5] = posbuffer[3];

posbuffer[4] = posbuffer[2];

posbuffer[3] = posbuffer[1];

posbuffer[2] = posbuffer[0];

posbuffer[1] = y;

posbuffer[0] = x;

// adjust x and y

xadj = (posbuffer[0]+posbuffer[2]+posbuffer[4]+posbuffer[6])/4;

yadj = (posbuffer[1]+posbuffer[3]+posbuffer[5]+posbuffer[7])/4;

posbufferadj[7] = posbufferadj[5];

posbufferadj[6] = posbufferadj[4];

posbufferadj[5] = posbufferadj[3];

posbufferadj[4] = posbufferadj[2];

posbufferadj[3] = posbufferadj[1];

posbufferadj[2] = posbufferadj[0];

posbufferadj[1] = yadj;

posbufferadj[0] = xadj;

// touching me?

if (pressed)

{

// first touch?

if (posbuffer[2] == 0)

{

startpos[0] = x;

startpos[1] = y;

}

// continued touchy

else

{

// move along the change buffer

changebuffer[7] = changebuffer[5];

changebuffer[6] = changebuffer[4];

changebuffer[5] = changebuffer[3];

changebuffer[4] = changebuffer[2];

changebuffer[3] = changebuffer[1];

changebuffer[2] = changebuffer[0];

changebuffer[1] = posbufferadj[1] - posbufferadj[3];

changebuffer[0] = posbufferadj[0] - posbufferadj[2];

// remove de baddies

if (changebuffer[0] > badjumpsize or changebuffer[0] < badjumpsize\*-1)

{

changebuffer[0] = 0;

}

if (changebuffer[1] > badjumpsize or changebuffer[2] < badjumpsize\*-1)

{

changebuffer[1] = 0;

}

tomove[0] = changebuffer[0]\*2;

tomove[1] = changebuffer[1]\*2;

}

}

else

{

changebuffer[7] = changebuffer[5];

changebuffer[6] = changebuffer[4];

changebuffer[5] = changebuffer[3];

changebuffer[4] = changebuffer[2];

changebuffer[3] = changebuffer[1];

changebuffer[2] = changebuffer[0];

if (changebuffer[1] < 0)

{

changebuffer[1] + 10;

if (changebuffer[1] > 0)

{

changebuffer[1] = 0;

}

}

else

{

changebuffer[1] - 10;

if (changebuffer[1] < 0)

{

changebuffer[1] = 0;

}

}

if (changebuffer[0] < 0)

{

changebuffer[0] + 10;

if (changebuffer[0] > 0)

{

changebuffer[0] = 0;

}

}

else

{

changebuffer[0] - 10;

if (changebuffer[0] < 0)

{

changebuffer[0] = 0;

}

}

tomove[0] = changebuffer[0];

tomove[1] = changebuffer[1];

}

// put outs

Mouse.move(tomove[0] \* sens, tomove[1] \* sens);

// polling rate

delay(2);

// delay(1000);

}

// define the pins im using

#define x2 A0

#define y1 A1

#define x1 A2

#define y2 A3

// add libraries

#include "Mouse.h"

// initialize variables

int x = 0;

int y = 0;

int z1 = 0;

int z2 = 0;

int pressed = 0;

int posbuffer[8];

int posbufferadj[8];

int startpos[2];

int changebuffer[8];

int xadj = 0;

int yadj = 0;

int badjumpsize = 10;

int tomove[2];

float sens = 3.5;

float mouseaccel = 1;

int pressuresens = 110;

void setup()

{

// setup whatever i need to setup

Serial.begin(9600);

Mouse.begin();

}

void loop()

{

// grab the x position

pinMode(y1,INPUT);

pinMode(y2,INPUT);

digitalWrite(y2,LOW);

pinMode(x1,OUTPUT);

digitalWrite(x1,HIGH);

pinMode(x2,OUTPUT);

digitalWrite(x2,LOW);

// y = analogRead(y1);

y = analogRead(y1) \* -1;

// grab the y position

pinMode(x1,INPUT);

pinMode(x2,INPUT);

digitalWrite(x2,LOW);

pinMode(y1,OUTPUT);

digitalWrite(y1,HIGH);

pinMode(y2,OUTPUT);

digitalWrite(y2,LOW);

// x = analogRead(x1) \* -1;

x = analogRead(x1);

// find pressure

pinMode(y1,OUTPUT);

digitalWrite(y1,HIGH);

pinMode(x2,OUTPUT);

digitalWrite(x2,LOW);

pinMode(x1,INPUT);

pinMode(y2,INPUT);

z1 = analogRead(x1);

z2 = analogRead(y2);

if (z1 > pressuresens)

{

pressed = true;

}

else

{

pressed = false;

x = 0;

y = 0;

}

// continue the buffer

posbuffer[7] = posbuffer[5];

posbuffer[6] = posbuffer[4];

posbuffer[5] = posbuffer[3];

posbuffer[4] = posbuffer[2];

posbuffer[3] = posbuffer[1];

posbuffer[2] = posbuffer[0];

posbuffer[1] = y;

posbuffer[0] = x;

// adjust x and y

// xadj = (posbuffer[0]+posbuffer[2]+posbuffer[4]+posbuffer[6])/4;

// yadj = (posbuffer[1]+posbuffer[3]+posbuffer[5]+posbuffer[7])/4;

posbufferadj[7] = posbufferadj[5];

posbufferadj[6] = posbufferadj[4];

posbufferadj[5] = posbufferadj[3];

posbufferadj[4] = posbufferadj[2];

posbufferadj[3] = posbufferadj[1];

posbufferadj[2] = posbufferadj[0];

// posbufferadj[1] = yadj;

// posbufferadj[0] = xadj;

posbufferadj[1] = y;

posbufferadj[0] = x;

// touching me?

if (pressed)

{

// first touch?

if (posbuffer[2] == 0)

{

startpos[0] = x;

startpos[1] = y;

}

// continued touchy

else

{

// move along the change buffer

changebuffer[7] = changebuffer[5];

changebuffer[6] = changebuffer[4];

changebuffer[5] = changebuffer[3];

changebuffer[4] = changebuffer[2];

changebuffer[3] = changebuffer[1];

changebuffer[2] = changebuffer[0];

changebuffer[1] = posbufferadj[1] - posbufferadj[3];

changebuffer[0] = posbufferadj[0] - posbufferadj[2];

// remove de baddies

if (changebuffer[0] > badjumpsize or changebuffer[0] < badjumpsize\*-1)

{

changebuffer[0] = 0;

}

if (changebuffer[1] > badjumpsize or changebuffer[2] < badjumpsize\*-1)

{

changebuffer[1] = 0;

}

if (changebuffer[0] < 0)

{

tomove[0] = pow((changebuffer[0] \* -1),mouseaccel) \* -1;

}

else

{

tomove[0] = pow(changebuffer[0],mouseaccel);

}

if (changebuffer[1] < 0)

{

tomove[1] = pow((changebuffer[1] \* -1),mouseaccel) \* -1;

}

else

{

tomove[1] = pow(changebuffer[1],mouseaccel);

}

}

}

else

{

changebuffer[7] = changebuffer[5];

changebuffer[6] = changebuffer[4];

changebuffer[5] = changebuffer[3];

changebuffer[4] = changebuffer[2];

changebuffer[3] = changebuffer[1];

changebuffer[2] = changebuffer[0];

if (changebuffer[1] < 0)

{

changebuffer[1] + 10;

if (changebuffer[1] > 0)

{

changebuffer[1] = 0;

}

}

else

{

changebuffer[1] - 10;

if (changebuffer[1] < 0)

{

changebuffer[1] = 0;

}

}

if (changebuffer[0] < 0)

{

changebuffer[0] + 10;

if (changebuffer[0] > 0)

{

changebuffer[0] = 0;

}

}

else

{

changebuffer[0] - 10;

if (changebuffer[0] < 0)

{

changebuffer[0] = 0;

}

}

// tomove[0] = changebuffer[0];

// tomove[1] = changebuffer[1];

tomove[0] = 0;

tomove[1] = 0;

}

// put outs

Mouse.move(tomove[0] \* sens, tomove[1] \* sens);

// Serial.print(z1);

// Serial.print(", ");

// Serial.print(z2);

// Serial.print("\n");

// polling rate

delay(1);

// delay(1000);

}