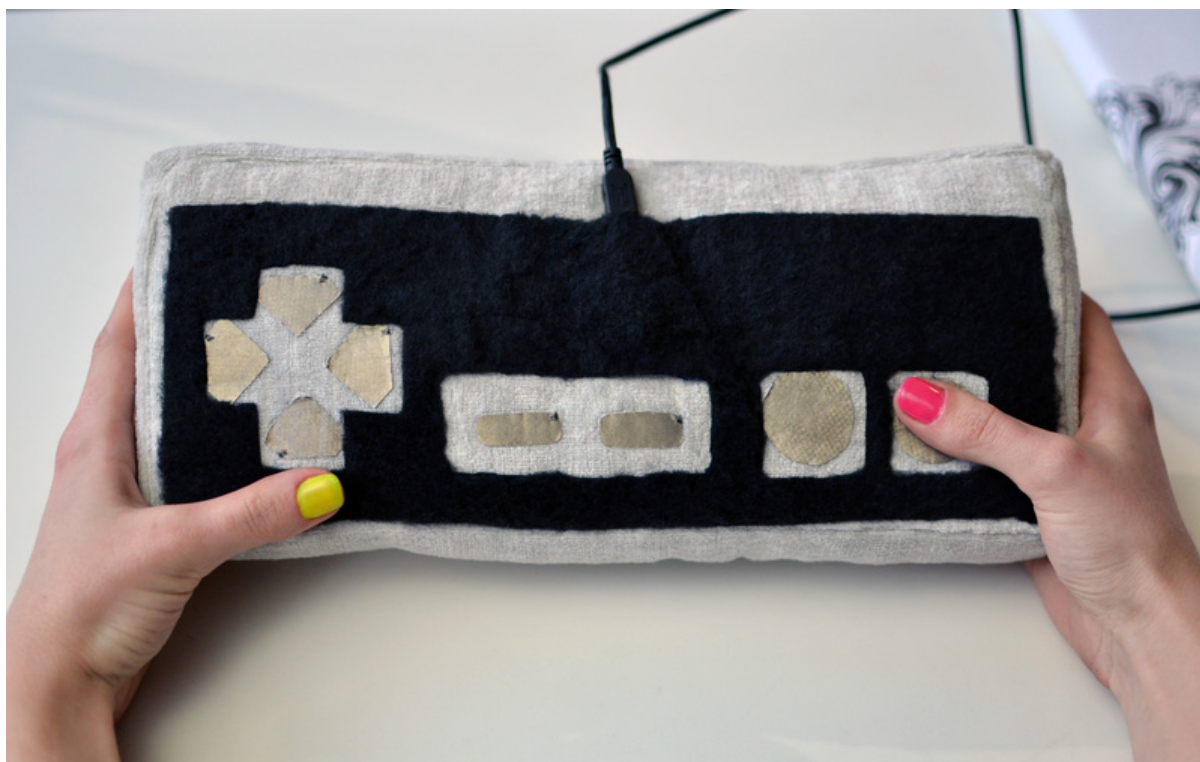




Plush Game Controller

Created by Becky Stern



<https://learn.adafruit.com/plush-game-controller>

Last updated on 2024-06-03 01:16:15 PM EDT

Table of Contents

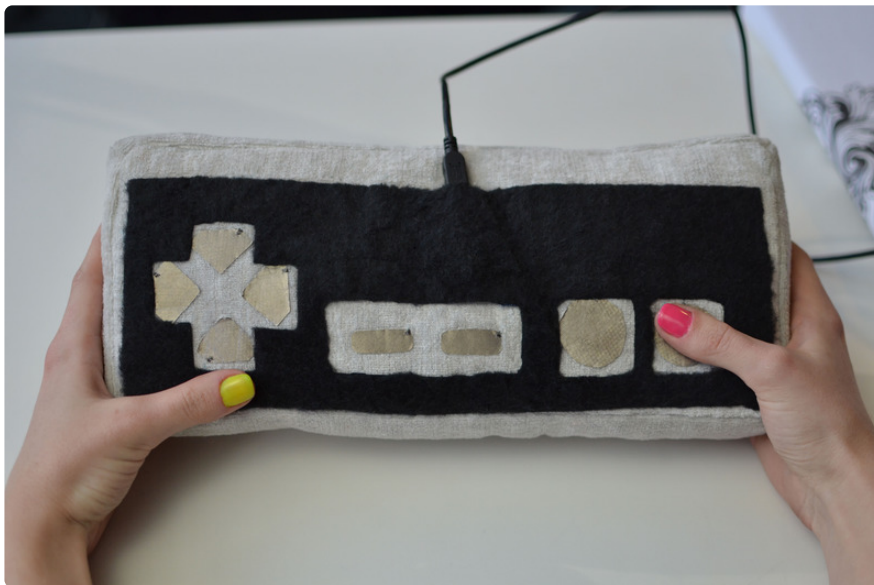
Overview	3
Pattern & Circuit Diagram	4
Stitch Circuit	5
The Code	9
Construct Plushie	12
Use it!	19

Overview

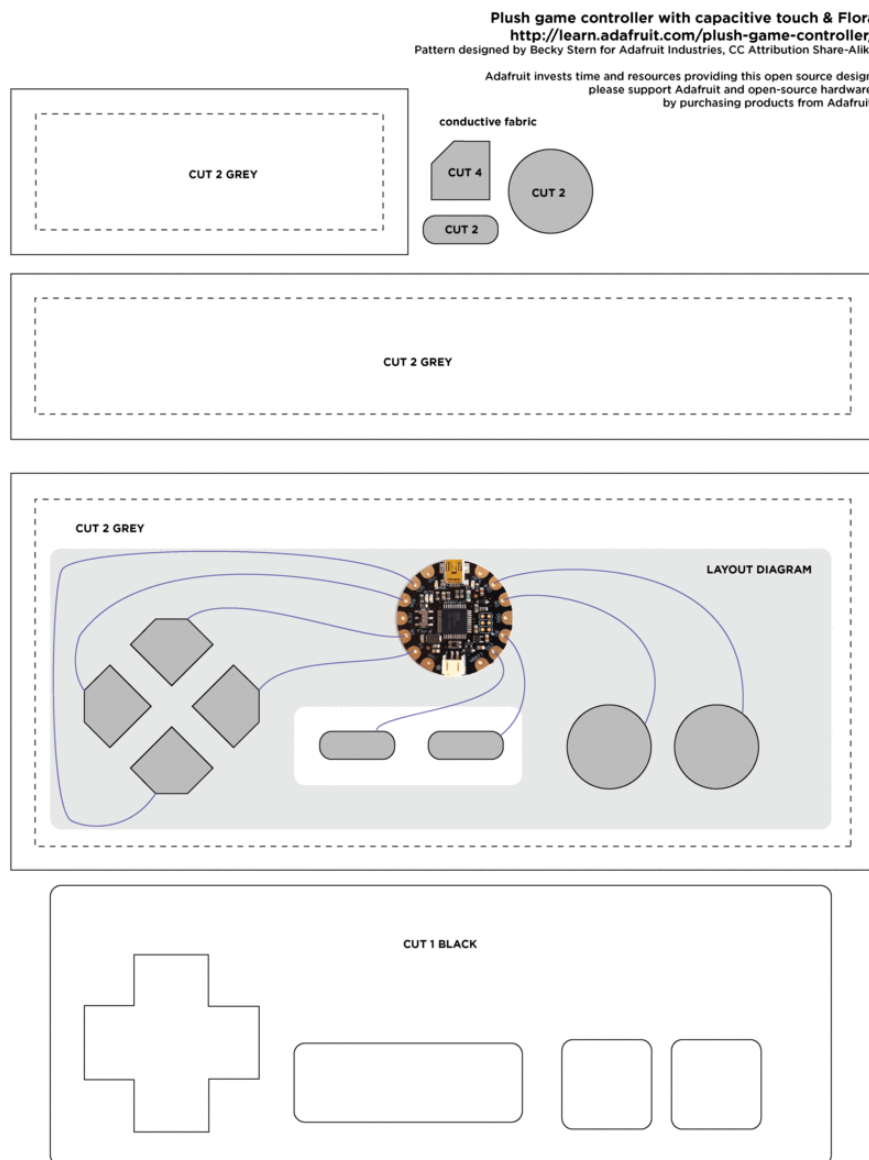
Make an oversize plush game controller that actually works! Use conductive fabric and a Flora board to stitch up a capacitive touch sensing circuit. The controller acts like a computer keyboard, allowing you to play all your favorite old school games on emulator sites online.

This is an all-sew project with no batteries, making it great for beginners and even kids to try. Read on to build your own!

Special assistance creating this project provided by Risa Rose.



Pattern & Circuit Diagram



We've made a PDF pattern for constructing this plush controller. Download the [tiled PDF \(https://adafru.it/c5Y\)](https://adafru.it/c5Y), print it out and tile the four pages together. We've also made available the [pattern Illustrator file \(https://adafru.it/c5Z\)](https://adafru.it/c5Z) if you'd like to make any changes.

Stitch Circuit



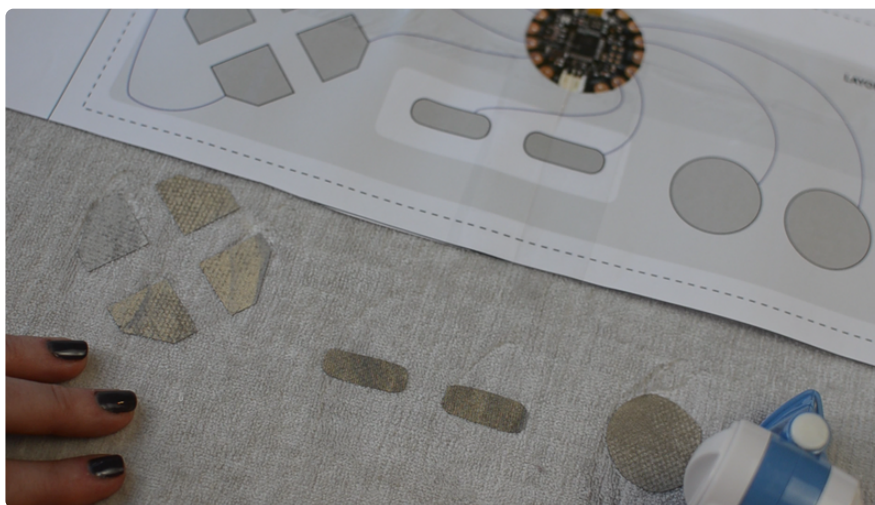
Iron a small piece of [double-sided iron-on interfacing \(https://adafru.it/c60\)](https://adafru.it/c60) to a piece of woven conductive fabric.



I like [this mini iron \(https://adafru.it/c61\)](https://adafru.it/c61) for better control with small pieces.

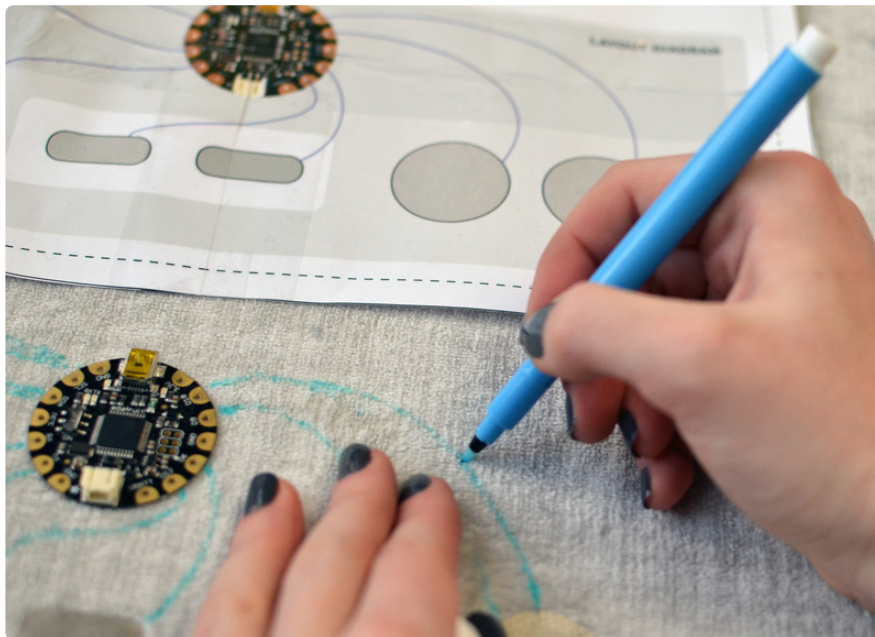


Peel off the paper backing (or don't), and cut out the shapes necessary for your controller buttons - refer to your pattern! You'll have four directional buttons, two option buttons, and two round action buttons.



If you haven't already, peel the backing from the interfacing and then iron the pieces of conductive fabric to some grey fuzzy fabric. Don't place them too close to the edge

of your fabric, or you won't have anything to grip onto with your embroidery hoop in the next steps.



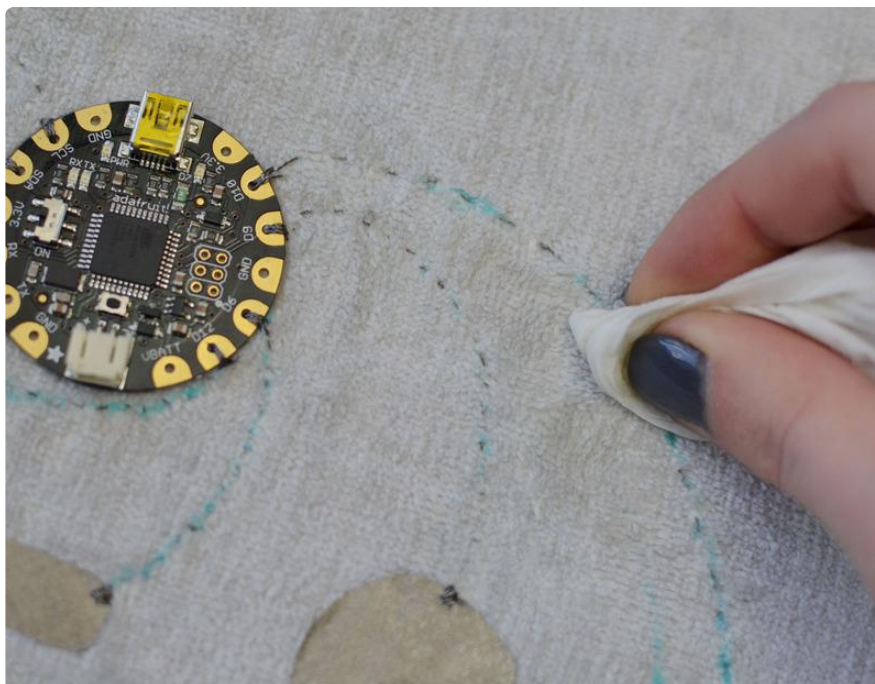
Place the Flora on the fabric, and following the pattern use a [water-soluble embroidery](https://adafru.it/c62) (<https://adafru.it/c62>) marker to sketch the traces that will connect the Flora pads to the conductive fabric buttons.

It's important that these traces are as far apart from each other as possible so they don't accidentally trigger each other's capacitive touch sensing. We've tested this particular pattern and it works great; we also made previous versions with traces closer together that did not work well, so keep that in mind if you're designing your own button layout.

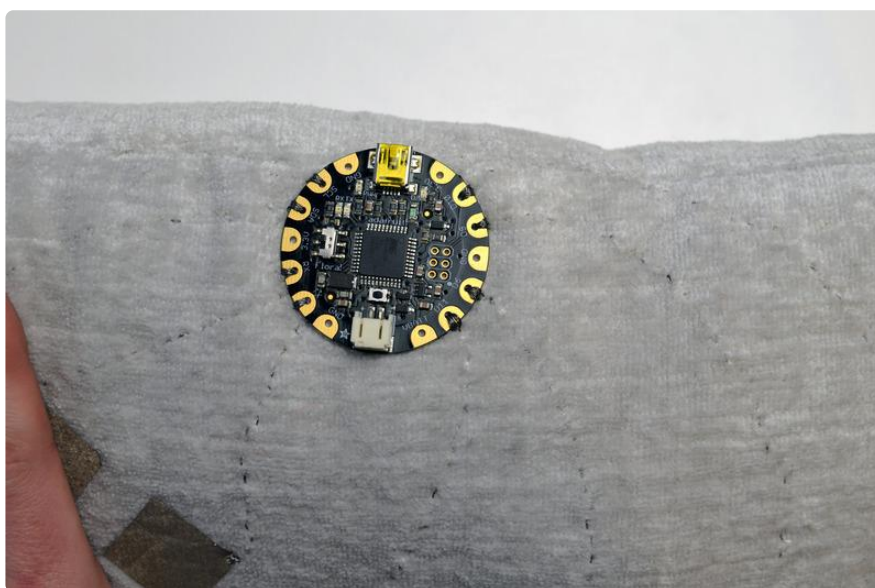


Embroider connections that follow the lines you just drew, wrapping several times around the Flora pads and into the conductive fabric, then knot, seal, and cut the thread at the back. For more information about working with conductive thread, check out our [Conductive Thread](https://adafru.it/aVx) (<https://adafru.it/aVx>) guide.

Remove the circuit from the embroidery hoop and iron out any creases.



Blot the blue traces with a damp paper towel to make it disappear.



The Code

Select Download File to get the project code from GitHub. For more information about programming Flora and to download the IDE, please visit the [Getting Started with FLORA guide](https://adafruit.it/aSZ) (<https://adafruit.it/aSZ>).

You'll also need [Modern Device's capacitive touch sensing library](https://adafruit.it/c66) (<https://adafruit.it/c66>) for Arduino. Refer to the Adafruit Guide [All About Arduino Libraries](https://adafruit.it/dit) (<https://adafruit.it/dit>) for help installing libraries.

```

// SPDX-FileCopyrightText: 2019 Limor Fried for Adafruit Industries
// SPDX-FileCopyrightText: 2019 Becky Stern for Adafruit Industries
//
// SPDX-License-Identifier: MIT

/*
Example code for a Flora game controller with capacitive touch sensing! Full
tutorial and video:
http://learn.adafruit.com/plush-game-controller/
Uses Modern Device's Capacitive Sensing library: https://github.com/moderndevise/
CapSense
Adafruit invests time and resources providing this open source code,
please support Adafruit and open-source hardware by purchasing
products from Adafruit!
Written by Limor Fried & Becky Stern for Adafruit Industries.
BSD license, all text above must be included in any redistribution

*/
#include <CapPin.h>
#include <Keyboard.h>

CapPin cPin_10 = CapPin(10);    // read pin 10 (D10 on Flora) - connect to NES B
CapPin cPin_9 = CapPin(9);      // read pin 9 (D9 on Flora) - connect to NES A
CapPin cPin_6 = CapPin(6);      // read pin 6 (D6 on Flora) - connect to NES Start
CapPin cPin_12 = CapPin(12);    // read pin 12 (D12 on Flora) - connect to NES
Select
CapPin cPin_1 = CapPin(1);      // read pin 1 (TX on Flora) - connect to NES right
CapPin cPin_0 = CapPin(0);      // read pin 0 (RX on Flora) - connect to NES up
CapPin cPin_2 = CapPin(2);      // read pin 2 (SDA on Flora) - connect to NES left
CapPin cPin_3 = CapPin(3);      // read pin 3 (SCL on Flora) - connect to NES down

CapPin pins[] = {cPin_10, cPin_9, cPin_6, cPin_12, cPin_1, cPin_0, cPin_2, cPin_3};
// check http://arduino.cc/en/Reference/KeyboardModifiers for more info on unique
keys

// WASD D-pad, select = Return, start = Space, LeftButton = z, RightButton = x
//char Keys[] = { 'x', 'z', ' ', KEY_RETURN, 'd', 'w',
'a', 's'};

// arrow D-pad, select = Return, start = Space, LeftButton = b, RightButton = a
char Keys[] = { 'a', 'b', ' ', KEY_RETURN, KEY_RIGHT_ARROW,
KEY_UP_ARROW, KEY_LEFT_ARROW, KEY_DOWN_ARROW};

boolean currentPressed[] = {false, false, false, false, false, false, false, false};

// Capacitive touch threshhold, you might want to mess with this if you find its too
// sensitive or not sensitive enough
#define THRESH 500

float smoothed[8] = {0,0,0,0,0,0,0,0};

void setup()
{
  //while (!Serial)
  Serial.begin(115200);
  Serial.println("start");
  Keyboard.begin();
}

void loop()
{
  for (int i=0;i<8;i++) {
    delay(1);
    long total1 = 0;
    long start = millis();
    long total = pins[i].readPin(2000);
  }
}

```

```

// check if we are sensing that a finger is touching the pad
// and that it wasn't already pressed
if ((total > THRESH) && (! currentPressed[i])) {
    Serial.print("Key pressed #"); Serial.print(i);
    Serial.print(" "); Serial.print(Keys[i]); Serial.println("");
    currentPressed[i] = true;

    Keyboard.press(Keys[i]);
}
else if ((total <= THRESH) && (currentPressed[i])) {
    // key was released (no touch, and it was pressed before)
    Serial.print("Key released #"); Serial.print(i);
    Serial.print(" "); Serial.print(Keys[i]); Serial.println("");
    currentPressed[i] = false;

    Keyboard.release(Keys[i]);
}

/*
// simple lowpass filter to take out some of the jitter
// change parameter (0 is min, .99 is max) or eliminate to suit
smoothed[i] = smooth(total, .8, smoothed[i]);
Serial.print(i); Serial.print(": ");
Serial.print(millis() - start);      // time to execute in mS
Serial.print("ms \t");
Serial.print(total);                // raw total
Serial.print("\t->\t");
Serial.println((int) smoothed[i]);   // smoothed
*/
delay(5);
}
}

// simple lowpass filter
// requires recycling the output in the "smoothedVal" param
int smooth(int data, float filterVal, float smoothedVal){

    if (filterVal > 1){          // check to make sure param's are within range
        filterVal = .999999;
    }
    else if (filterVal <= 0){
        filterVal = 0;
    }

    smoothedVal = (data * (1 - filterVal)) + (smoothedVal * filterVal);

    return (int)smoothedVal;
}

```

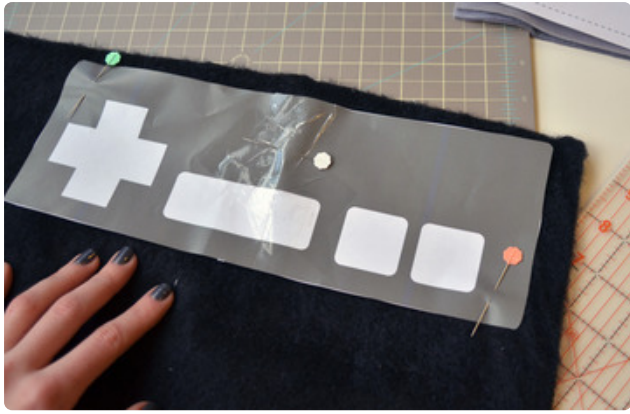


Open a text editor and test your circuit! It should type keypresses into the editor. If your thread traces are too close together, the buttons could interfere with each other, so be sure to follow the pattern for this project and keep the traces from coming unnecessarily close together! Once you can verify the circuit works its time to turn it into a 3D plush toy.

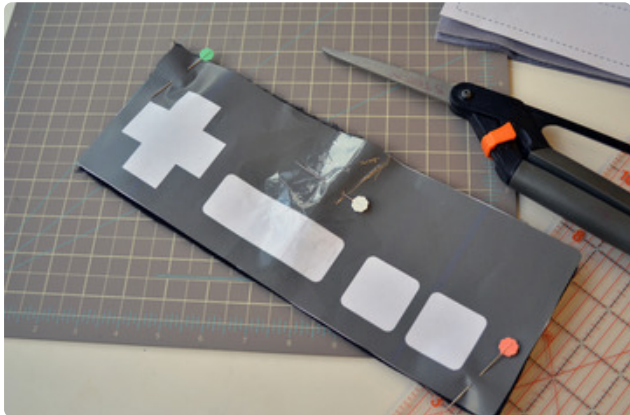
Construct Plushie



Cut out one rectangle for the back of the plushie according to the pattern, and two of each side pieces. We'll cut the front panel in a later step. If your fabric has a nap (soft in one direction and rough the opposite way), be sure to observe it and cut your pieces accordingly.



Use the black overlay template to cut a piece of black felt or other fuzzy fabric. Iron a piece of double-sided interfacing to the back of this piece and repin the pattern piece to the fabric.





Use a ruler and sharp blade to cut out the button windows. For curved corners, you can cut the straight portion of the lines with a knife and then finish off with a small pair of scissors. Felt doesn't fray and if you're using another fabric, the interfacing should help it not fray.



Before cutting the front panel, iron on the black "faceplate" piece, aligning the button windows with the conductive fabric buttons. This fabric insulates the thread traces from your hand (preventing unintended triggers) and also gives the controller that classic look.

I sandwiched in a piece of scrap fabric just big enough to cover the Flora board itself, just so it wouldn't get sticky in case I want to look at it or use it for another project.

make sure the USB port is aligned with the edge of this piece for easy connecting!

Use the front panel pattern piece (same as the back panel pattern piece) to cut off excess fabric around the circuit.



Pin the plushie side pieces together to form a rectangle, making sure the nap of the fabric all goes in the same direction (at this point it does not matter which direction, so long as there is consistency). Machine- or hand-stitch all four seams.



Lay your new rectangle out on top of the back of the plushie, and align and pin the edges, right sides together. If your fabric is especially fluffy with a strong nap, make sure the soft direction goes toward the seams you are about to sew. This will ensure that when the toy is finished, the fur will look and feel nice to the touch in a downward direction.

Stitch all four edge seams with a machine or by hand, leaving a gap on one long edge for stuffing.



Align the bottom part of the plushie to the front piece and pin around the edges, being sure the nap on the back panel matches the nap on the front panel. Stitch around all four edges and trim stray threads.

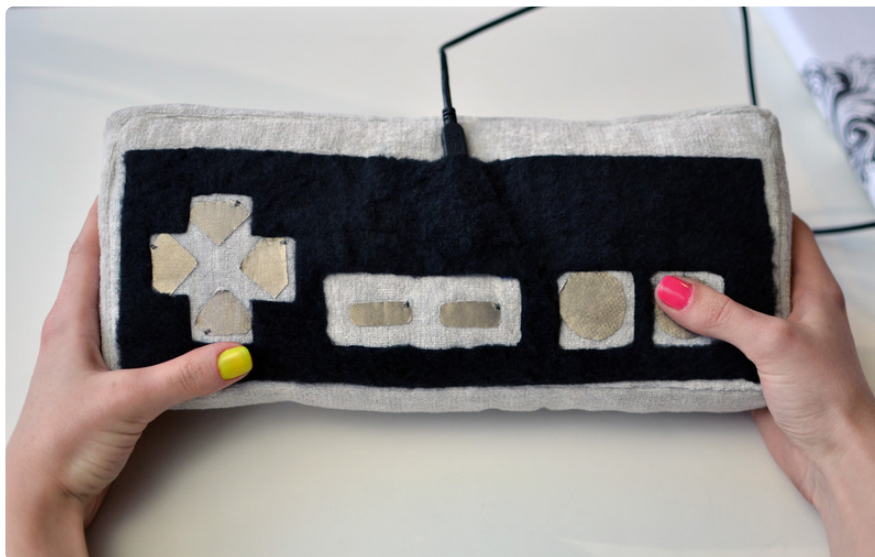


Turn the fabric shell right side out and fill with fiber fill, starting with the corners. Use a long chopstick to stuff the filling into the corners and then fill the remaining portion with small bits of stuffing at a time until it is reasonably firm and not lumpy.



Use a [ladder stitch \(https://adafru.it/c64\)](https://adafru.it/c64) to close up the seam, and your plush controller is complete!

Use it!



Plug the controller into your computer and start playing your favorite games online! We had fun playing the games on [VirtualNES \(https://adafru.it/c65\)](https://adafru.it/c65). You can customize the Arduino sketch to trigger any keyboard actions you'd like, so it's totally customizable to whatever keyboard-interaction game you can find. Happy stitching!