



# Giant LEGO NES Controller

Create your own giant NES Controller out of LEGOs. Use it to play silly games on your computer.

Written By: Joshua



## INTRODUCTION

This variation on our [NES Controller project](#) makes use of a very large amount of LEGO bricks in order to make a very large USB game controller in a classic NES style. While fundamentally no different then the original project, it does require a gross amount of LEGO parts in order to build.

On the upside it's really fun to play Super Mario Brothers on a massive oversized game controller.



### TOOLS:

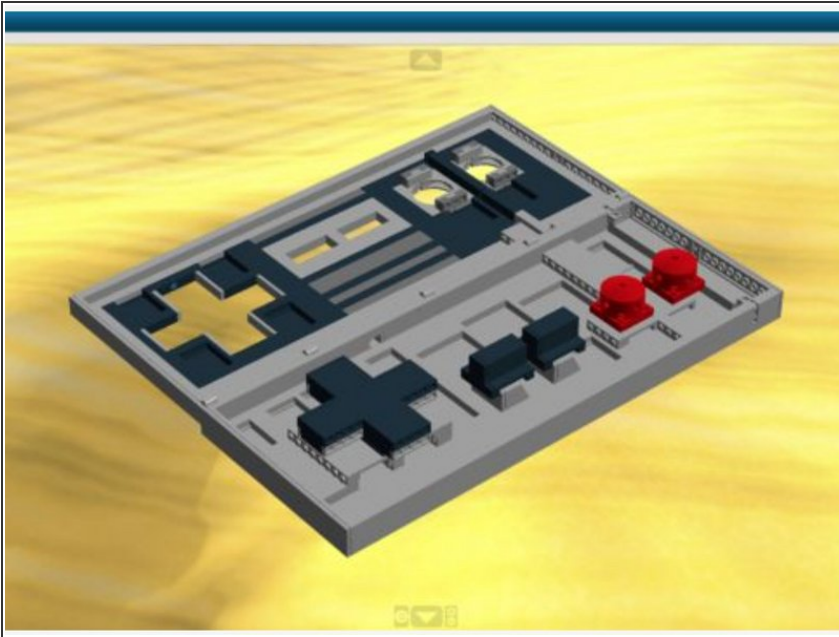
- [Scissors](#) (1)
- [Computer](#) (1)



### PARTS:

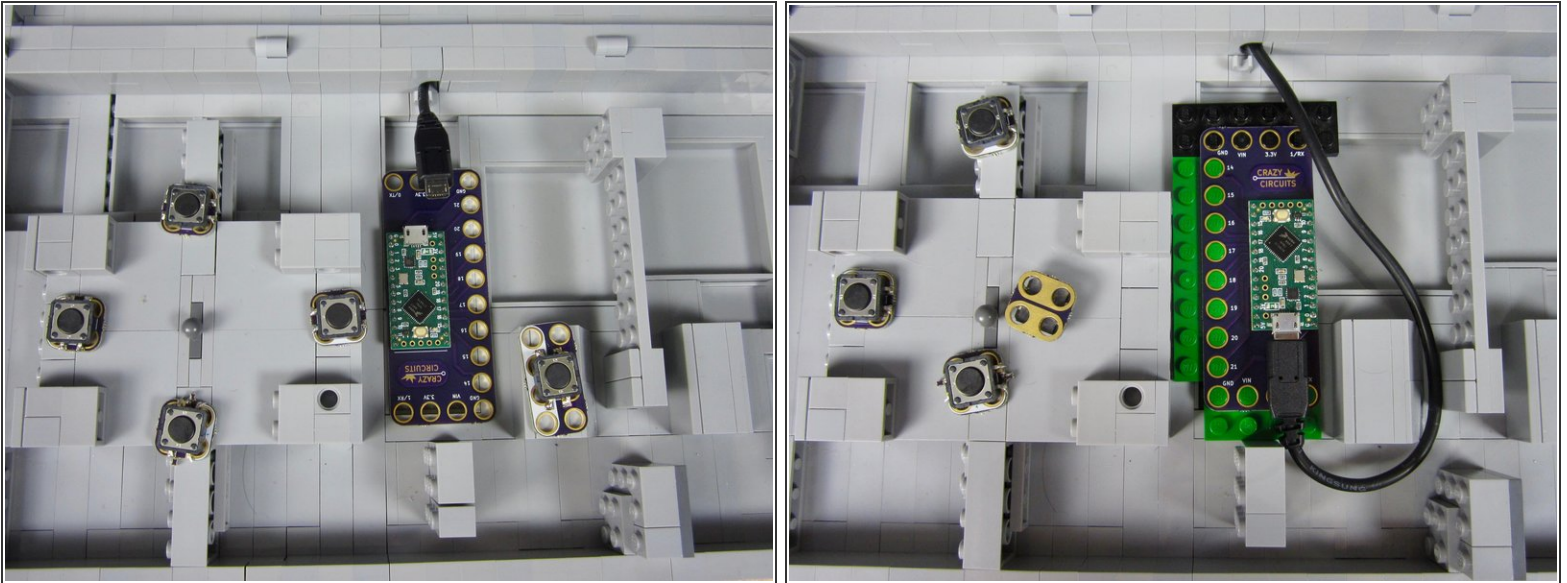
- [Crazy Circuits Touch Board](#) (1)
- [Jumbo Pushbutton Chip](#) (6)
- [LEGO Bricks](#) (1)

## Step 1 — Build the Controller



- Download the free [LEGO Digital Designer](#) software.
- Download our [Giant LEGO NES Controller](#) file for use with LEGO Digital Designer. (Click the link and then click the Download button found on the site.)
- Within LEGO Digital Designer you can grab a parts list. Use this list and buy way too many bricks from [BrickOwl](#) or [BrickLink](#).
- Once all your parts show up use the "Generate Steps" feature built into Digital Designer to create your own directions.
- ① We spent around \$125 buying up all the bricks for this project. This makes the project a bit more expensive than most LEGO kits, but quite a bit less than the big kits being sold these days.

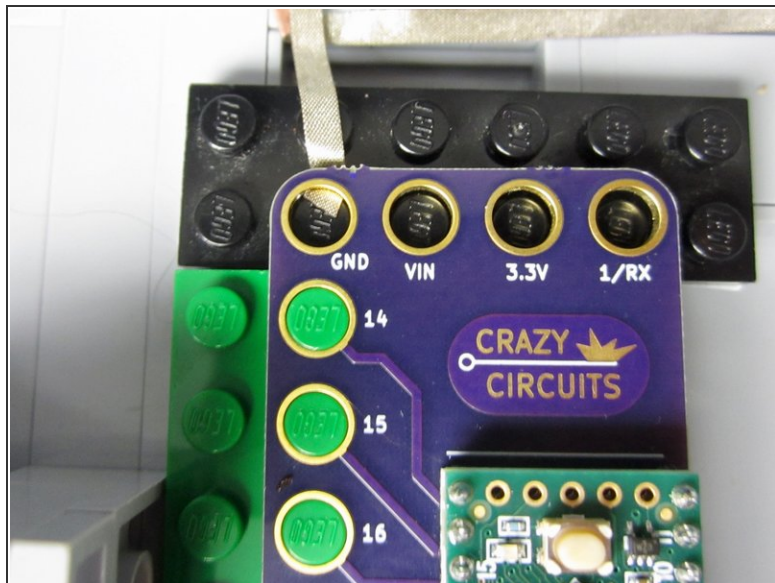
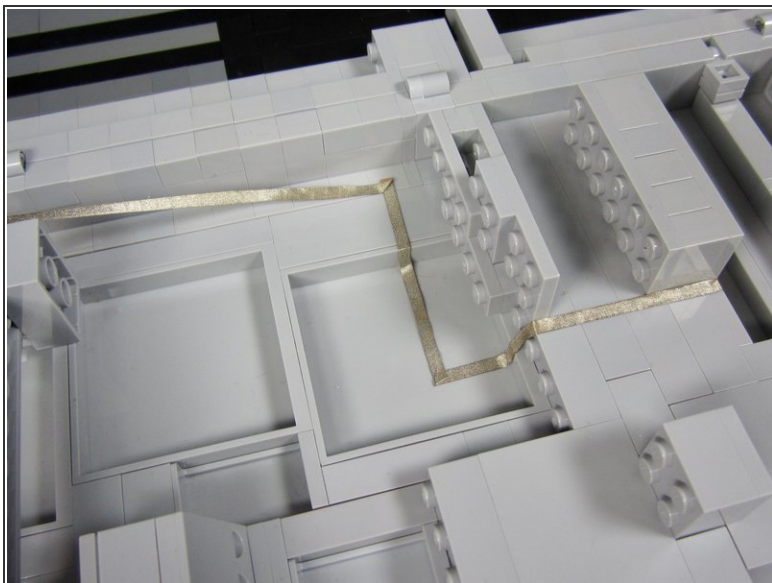
## Step 2 — Lay Out Your Parts



- Figure out where you're going to put everything inside your controller.
- ⓘ We used some slightly smaller sized Jumbo Pushbuttons for the D Pad. You can get your own made up by grabbing files from our [GitHub](#) and then getting them made from [OSHPark](#). We'd just recommend using our standard ones.
- ⚠ Since the inside of our controller is smooth we cheated a bit and taped down some plates so that the Touch Board was more secure. We also used double sided tape to secure down all the buttons.
- ⚠ We used a slightly different prototype version of our Touch Board for this project, since we were worried about spacing. However our standard board would also fit just fine in the space available.

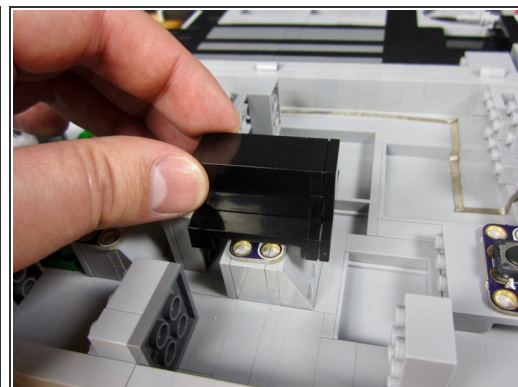
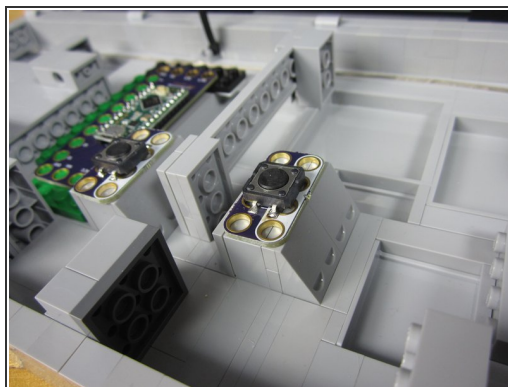
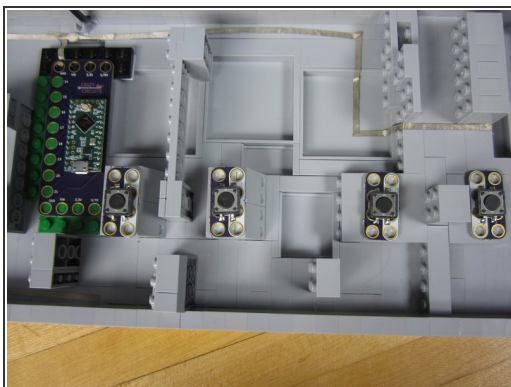


### Step 3 — Run a Common Ground



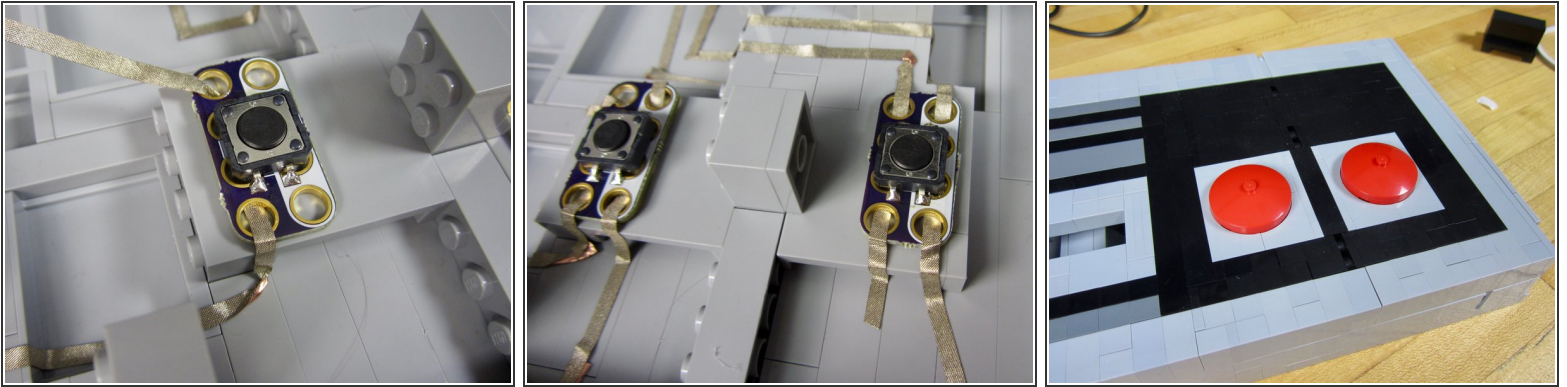
- Run a very long line of tape from a Ground connection on the Touch Board all the way to the end of the board.
- ① We'll be using this to hook up Ground for several buttons.

### Step 4 — Tape Down the Buttons



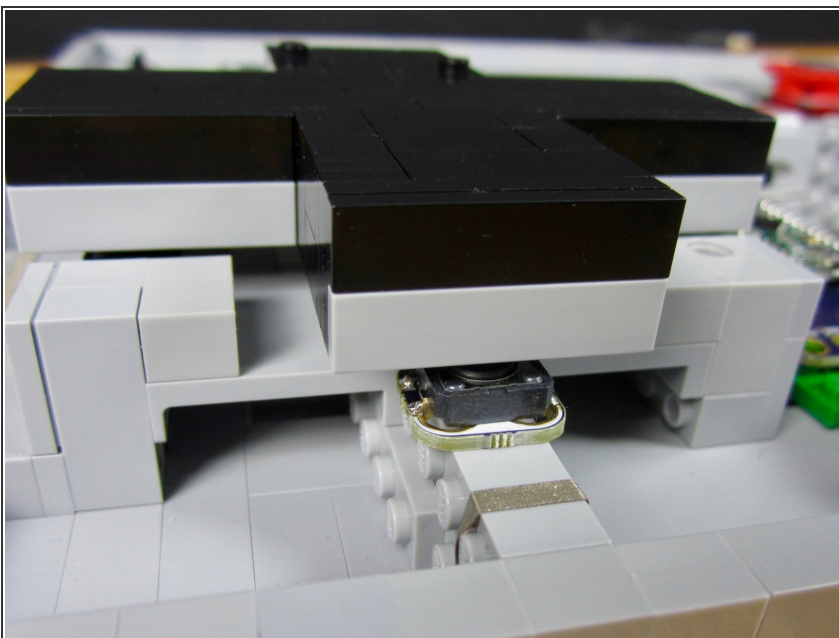
- Use a small piece of double sided foam tape to secure down all the Pushbuttons.
- Our current code doesn't allow for a "Start" and "Select" button. None the less, we wired them in anyways so we can eventually add functionality when we update the code.

## Step 5 — Connect A & B to the Touch Board



- Connect the colored side of the B button to Pin 0.
  - Connect the colored side of the A button to Pin 1.
  - Connect the white side of each button to the Common Ground.
- i* The Pins you use don't matter. You can easily change that in the code.
- i* We weaved the tape through the holes to make a solid connection. It helps to test your connections with a Multimeter.

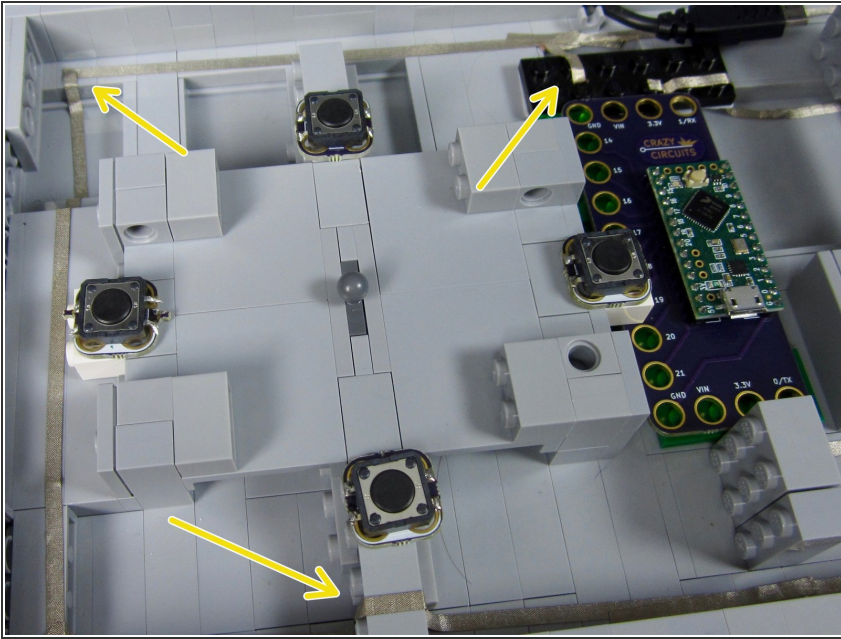
## Step 6 — Tape Down the D Pad



- Position and tape down the four D pad buttons.
- Put your LEGO D Pad in place and test to make sure each button is being pressed. You should hear a little "click" then pressed.

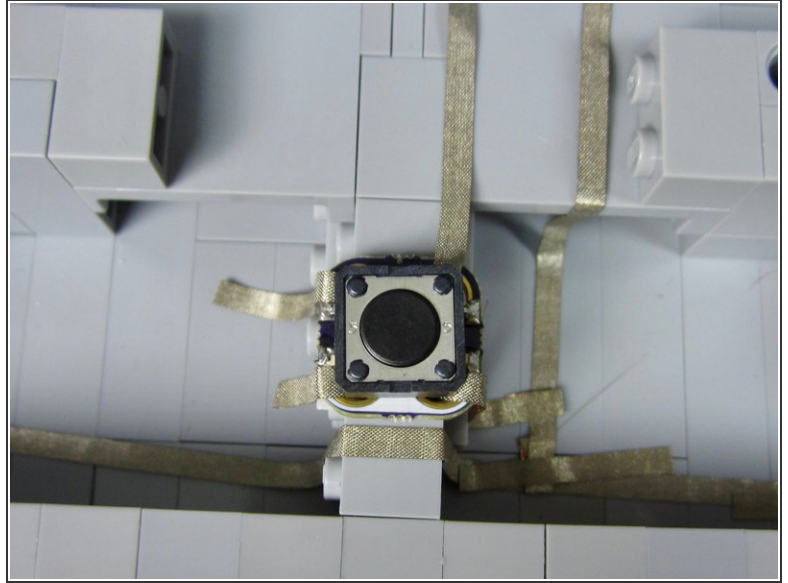
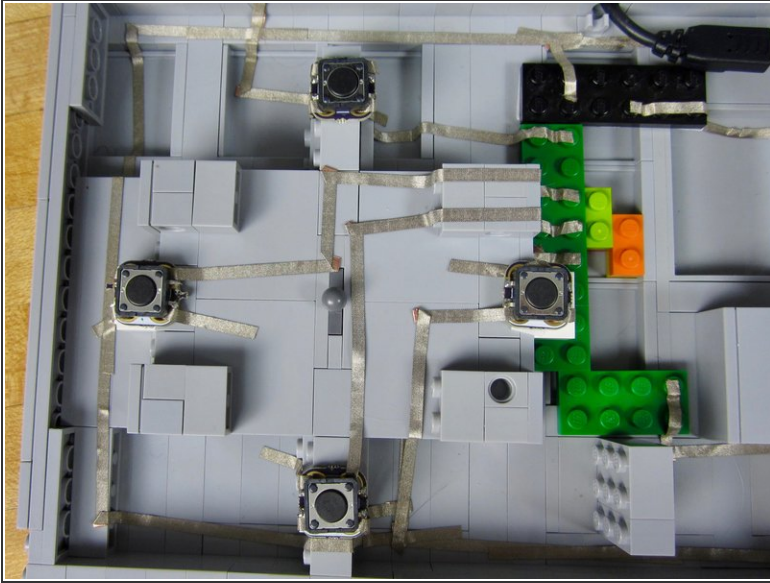


## Step 7 — Make a Common Ground



- Make a Common Ground line from your Touch Board around your D Pad.
- Feel free to attach it to the original Common Ground you made. Test with a Multimeter to make sure all your connections are strong.

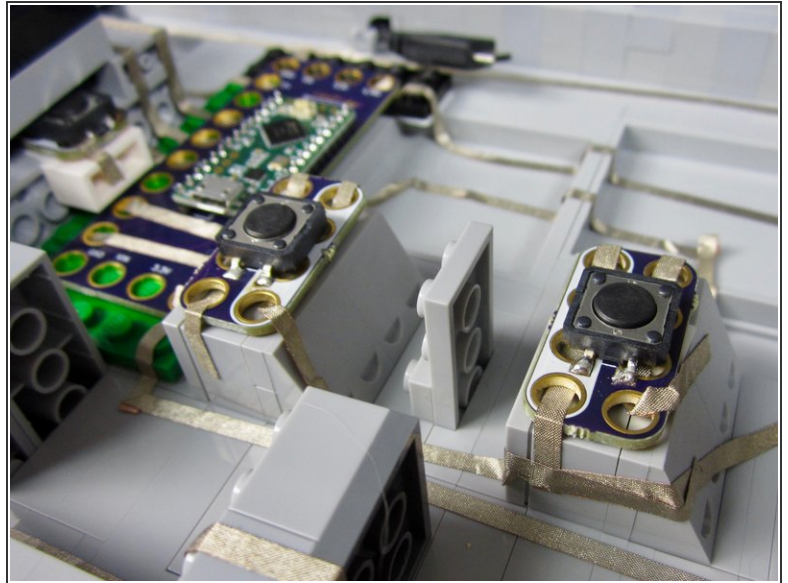
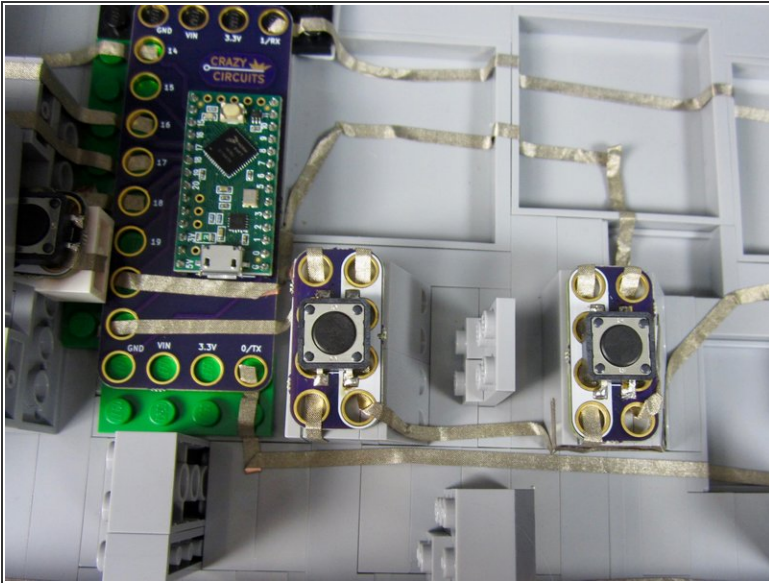
## Step 8



- Connect each of the D Pad buttons to the Touch Board.
  - We used Pins 14, 16, 17, and 18. Again, it doesn't matter which numbered pins you use as we can change things in the code.
- ⚠ Don't overlap lines of tape. This can cause issues shorts and mess up signals. When in doubt just redo a line. The tape should peel up and be reusable.**



## Step 9 — Optional: Wire up Start and Select



- ❗ Our current software doesn't have code for the Start and Select buttons. Yet.
- If you want to wire in the Start and Select buttons do so now.

## Step 10 — Upload the Code

```


11
12
13 // set the pins
14 int buttonDownPin = 15;
15 int buttonLeftPin = 16;
16 int buttonUpPin = 17;
17 int buttonRightPin = 18;
18 int buttonBPin = 19;
19 int buttonAPin = 20;
20 int buttonCPin = 14;
21 int buttonDPin = 23;
22
23 // set LED pin
24 int LEDPin = 13;
25
26 // set a slight delay
27 int keyDelay = 3;
28
29 void setup() {
30 // set all of our pins for input

```

```

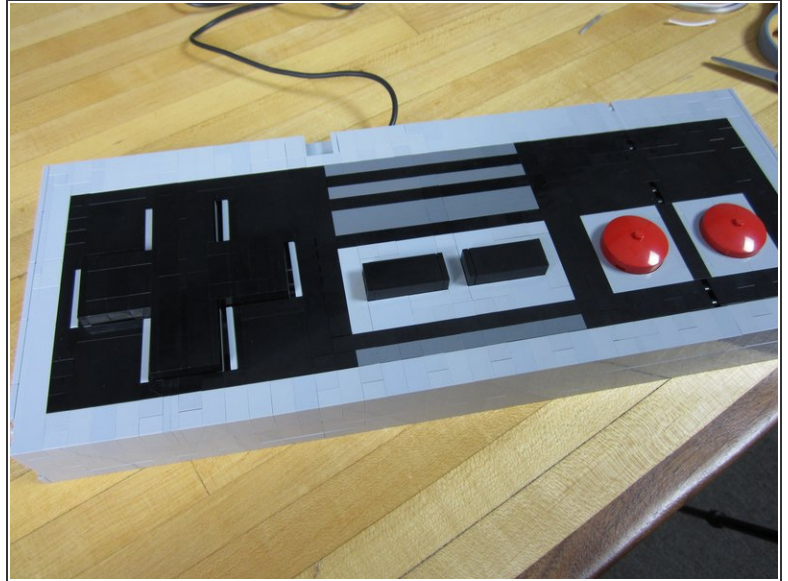
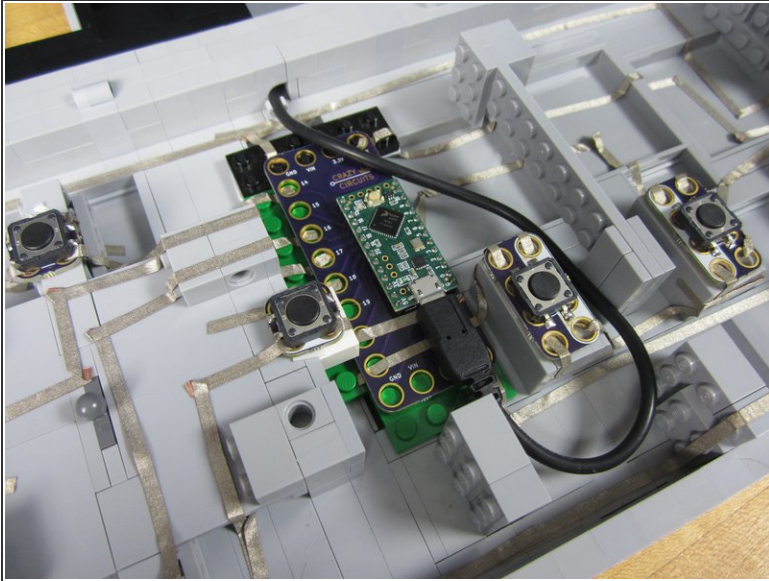
95
96
97 // check for B button
98 if (digitalRead(buttonBPin) == LOW) {
99   Keyboard.set_key5(KEY_B);
100   Keyboard.send_now();
101 }
102 else {
103   Keyboard.set_modifier(0);
104   Keyboard.set_key5(0);
105   Keyboard.send_now();
106 }
107
108
109 // check for A button
110 if (digitalRead(buttonAPin) == LOW) {
111   Keyboard.set_key6(KEY_A);
112   Keyboard.send_now();
113 }
114 else {
115   Keyboard.set_modifier(0);
116   Keyboard.set_key6(0);
117   Keyboard.send_now();
118 }
119

```

 If you've never used the Touch Board before, STOP! Read the [Touch Board User Guide](#). You need to install both the Arduino Software as well as some additional software for programming.

- Open up Arduino. Open a new project window. [Copy and Paste in this code](#).
- You're probably going to need to change a but of code. Lines 14-21 let you change which Pin is which button.
- For instance, in Line 14 says that Pin 15 is the "Down" button. If you connected your "Down" button to a different pin you need to change that number. Same thing for all the other pins.
- Our code makes your computer think that our Touch Board is a USB KeyBoard. When you press one of your Pushbuttons your computer types a keyboard key. You can change which keys are pressed in the code.
- For example, line 111 control shows that the "A" Pushbutton will in fact type the letter A when pressed. If you change **Key\_A** to **Key\_F** that button will now type the letter F. The other buttons are controlled the same way.

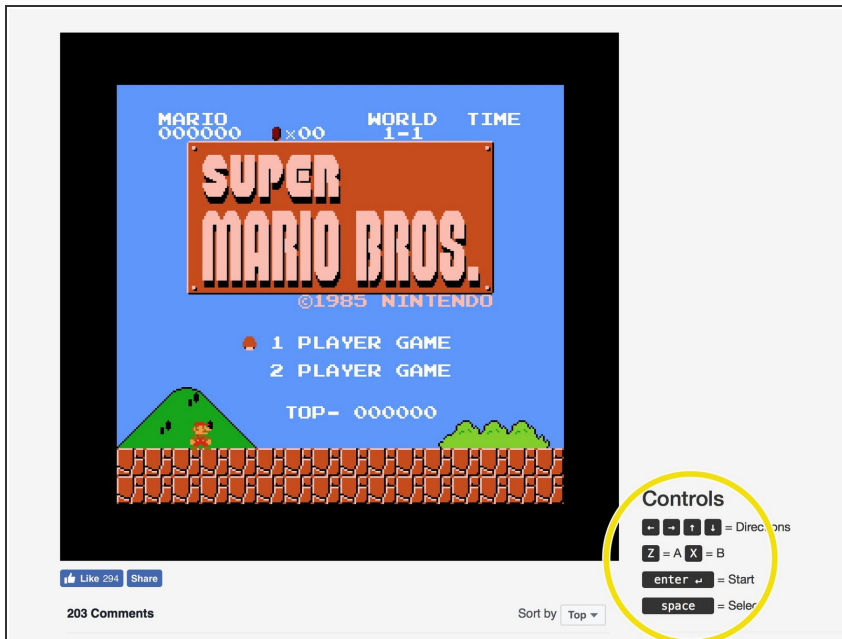
## Step 11 — Test Everything... Again



- Seriously. Test everything again with a multimeter.
- Make sure your double sided tape is holding your buttons down.
- Make sure you code is correct. When in doubt, open up a text document and start pressing buttons. Does the "Up" make your curser go up? Does the "A" button type the letter A?



## Step 12 — Playing Mario in a Browser



- Use [this link](#) to play Super Mario Brothers in your web browser.
- Notice how this website requires you to use the Z and X keyboard keys to control Mario. You'll need to slightly modify the code and re upload.
- ❗ There are many many places online that allow you to play classic video games in a web browser or on your computer.
- If you've got access to a Raspberry Pi you can use your controller with that system as well.