

Controller Read me

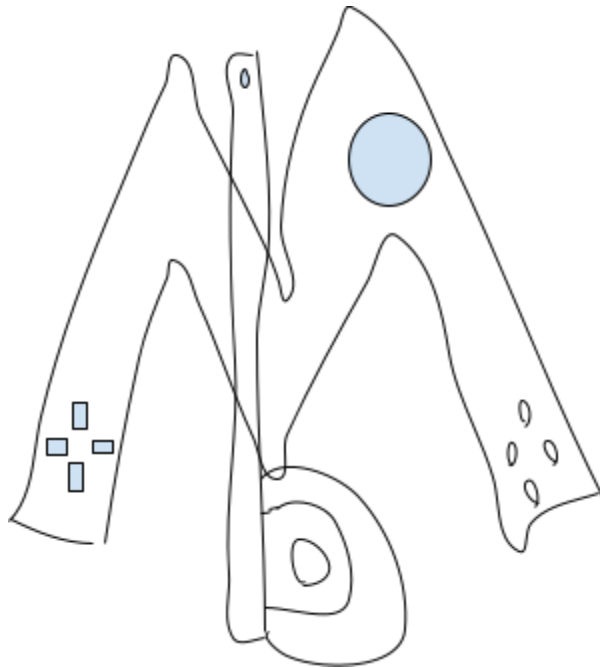
Link to Doc for working Gifs:

<https://docs.google.com/document/d/1xZBi9KXqXY9MyEPhsdJzQ39GcelliGPsm5kTkEw7KSI/edit?usp=sharing>

Brain Storm:

Name Initials: M P

Idea: I want to make a novel controller for the game overcooked, It will use 1 joystick, 8 push buttons, a LED and a buzzer.



I have designed it so that the player can use the right buttons to interact in the game, and the joy stick to move. Since the d-pad is only used to emote the player does not always need to hold on to it. This is why the player only needs to hold two of three places at once. This also has a Led and Buzzer, both of these will flash/buzz when the player is about to run out of time in the game.

Electronics Prototype:

In this prototype the values of the joystick(shown here with two potentiometers) and buttons is compressed into a string and sent through the serial monitor, this can be read by the game and will be used to control the in game character. The game will also post the time value in the serial monitor allowing the LED and Buzzer to flash when the time is low.

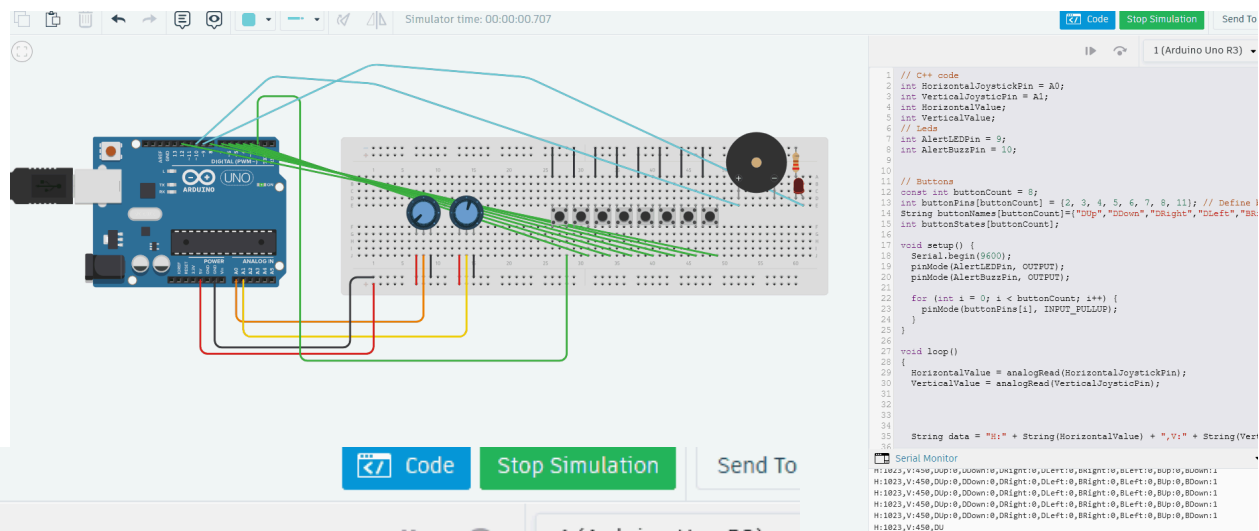
I started by laying out all my components and wiring them to ground and power. The i decided what ports i wanted to use for inputs. I made an array of ints and array of strings which would hold the input pin numbers and name of each pin. Then I set up the output going to the buzzer and LED. As the code starts, the analog pins will be read for the potentiometer values and each push button will send a 0 or 1 depending on if it is pressed. All this data will be put into one string with the name and value of each component so that the unity integration can read it all at once.

Link to TinkerCad:

https://www.tinkercad.com/things/inEeOpSQZla/editel?returnTo=%2Fdashboard&sharecode=wSHrDvNSe9e07swtx5Pu-JUDUWLIBr7Z7Bk_nwysial

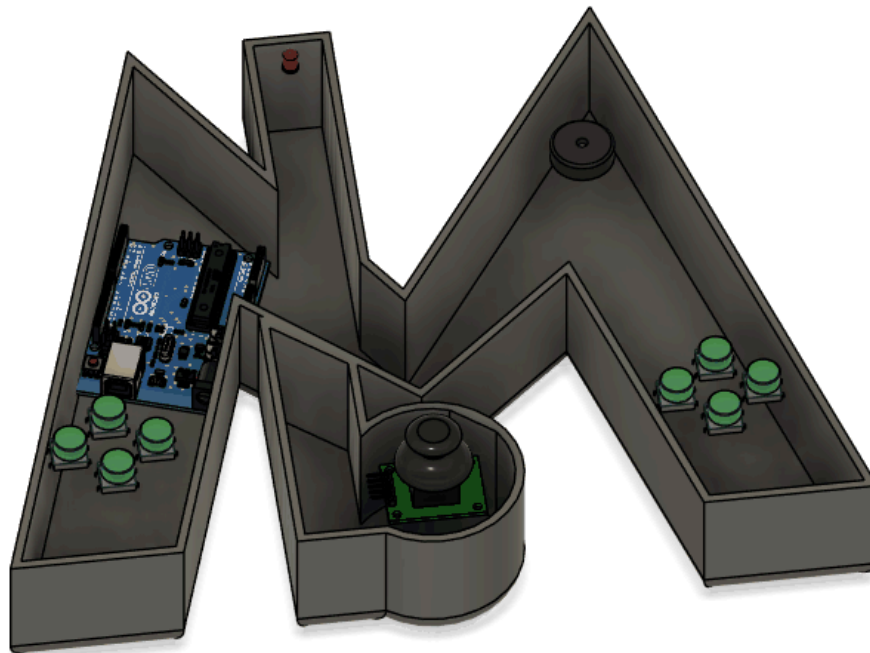
VideoLink: <https://youtu.be/pZZkxIhkBvU>

Gif:



```
1 // C++ code
2 int HorizontalJoystickPin = A0;
3 int VerticalJoystickPin = A1;
4 int HorizontalValue;
5 int VerticalValue;
6 // Leds
7 int AlertLEDPin = 9;
8 int AlertBuzzPin = 10;
9
10
11 // Buttons
12 const int buttonCount = 8;
13 int buttonPins[buttonCount] = {2, 3, 4, 5, 6, 7, 8, 11}; // Define
14 String buttonNames[buttonCount]={"DUp","DDown","DRight","DLeft","BR
15 int buttonStates[buttonCount];
16
17 void setup() {
18   Serial.begin(9600);
19   pinMode(AlertLEDPin, OUTPUT);
20   pinMode(AlertBuzzPin, OUTPUT);
21
22   for (int i = 0; i < buttonCount; i++) {
23     pinMode(buttonPins[i], INPUT_PULLUP);
24   }
25 }
26
27 void loop()
28 {
29   HorizontalValue = analogRead(HorizontalJoystickPin);
30   VerticalValue = analogRead(VerticalJoystickPin);
31
```

Assembly:



VideoLink: https://youtu.be/fw3fxxY9s_4

I started modelling in Fusion by first placing all my components roughly where they were supposed to be and then drawing my rough controller shape over them to make sure that they all fit. While doing this I realized that I would need to make the controller bigger so that I could fit my components inside of it. Especially the Arduino. I then extruded the shape and made an offset so that i could hollow out space for my components.

Exploded View/ Technical drawings/ Bill of Materials:

Parts List			
Item	Qty	Part Number	Material
1	1	Arduino Joystick Module_	Steel
2	1	Aurdino v5	
3	8	MyPushButton v2	
4	1	PKM22EPPH4001-B0	Steel
5	1	LED 5mm	Steel

Dept.	Technical reference	Created by Manvir Punglia 2024-12-05	Approved by		
		Document type Exam Drawing	Document status		
		Title Assembly - Exploded View	DWG No. 01		
		Rev. X	Date of issue 05/12/2024	Sheet 1/2	

Dept.	Technical reference	Created by Manvir Punglia 2024-12-05	Approved by		
		Document type Exam Drawing	Document status		