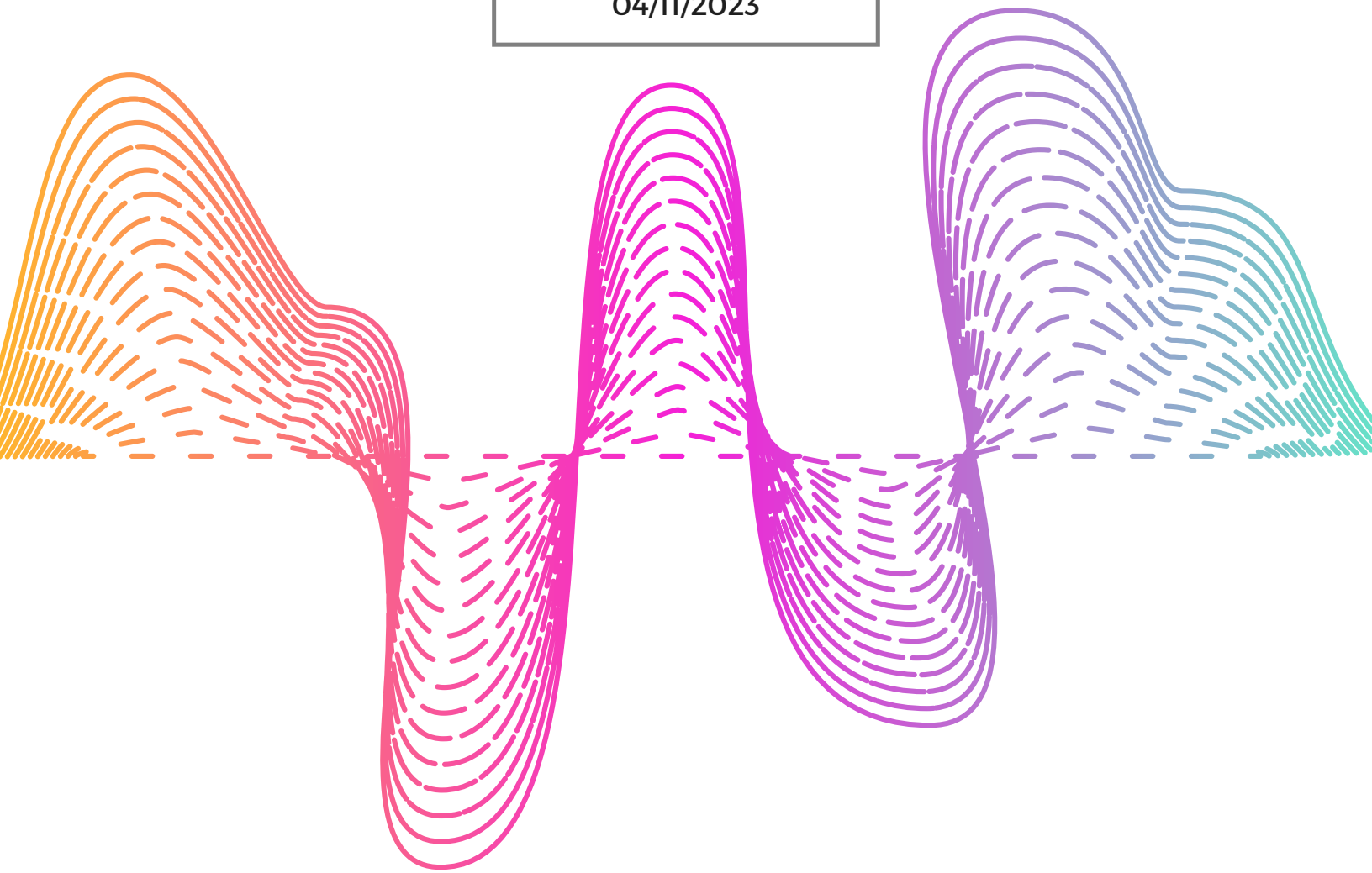
DIGF-2015-502: ATELIER II

FINAL PROJECT DOCUMENTATION

"DIY DJ"

04/11/2023



Prepared For :

DIGF-2015-502
Atelier II: Collaboration
Zoma Maduekwe
Adam Tindale

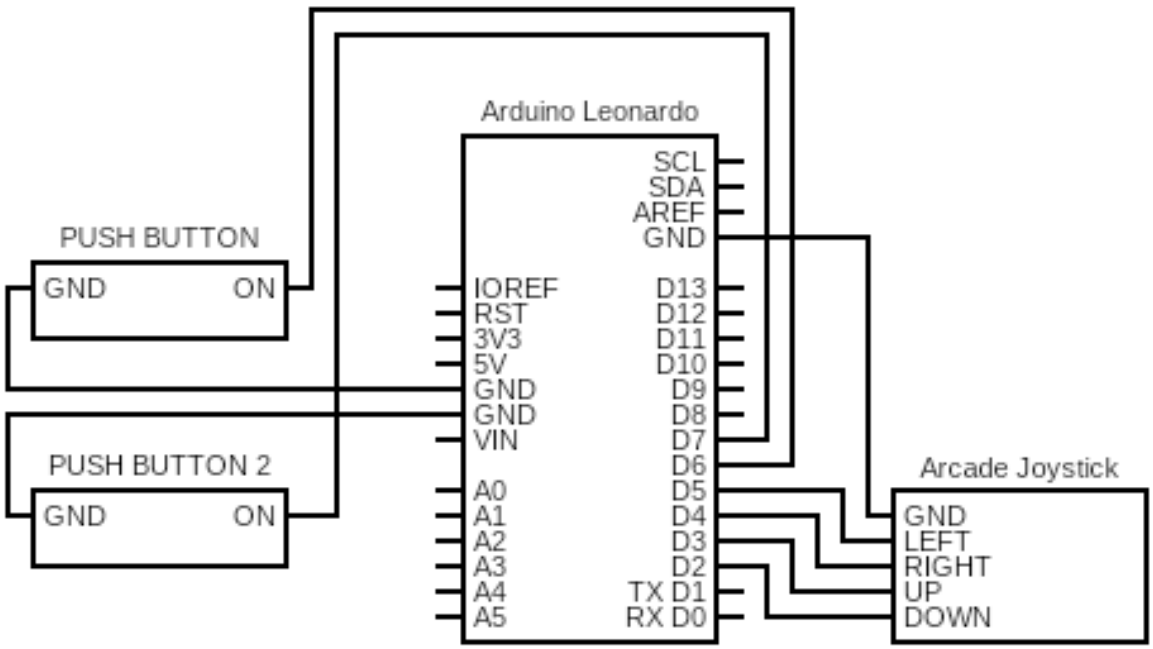
Created By:

Angelina Do, Jason Fung, Liam Gonzales, Minh Nguyen, Khalifa Seck

Project Description

“DIY DJ” or “Do-It-Yourself DJ” is an interactive music visualizer game using controllers and buttons that will be displayed onto the player’s webcam using AR. Players are encountered with the challenge to piece back together $ a musical puzzle. The joystick allows you to navigate through the player menu while the buttons manipulate musical elements such as growl, reverb, and pitch. Using webcam tracking, when buttons of the controller are pressed an audio visualizer with the shape of the face (deformation based on offset controls) will be projected by tracking the player’s webcam (augmented reality). The clarity of the player's image will depict on how "hot" or "cold" they are from putting the game back to it's original audio and imagery. “DIY DJ” software was created via TouchDesigner while hardware include Arduino components and a 3D printed casing encompassing the design of a game controller.

Circuit Diagram



Pseudocode

- Load: Joystick Chop.
- Select6: Y-axis input.
- Fan2 out: y-axis input into 3 channels (0, 1, 2).
- Select12: channel 1.
- Select13: channel 2.
- Count10: channel 1 input, off to on at an increase.
- Count9: channel 2 input, off to on at a decrease.
- Math6: ADD channel 1 count and channel 2 count.
- Limit1: loop Sum value from 0 to 3.
- Fan1 out: sum value into 3 channels (1, 2, 3).
- Select9: channel 1.
- selGr: channel 1.
- Logic2: if selGr is equal to 1 AND null11 is equal to 1, output 1 to logic7.
- Logic7: if logic2 is equal to 1 AND select5 is equal to 1, output 1 to count6.
- Count6: Increase value on “off to on” and output to math1.
- Select10: channel 1.
- selFo: channel 1.
- Logic3: if selFo is equal to 1 AND null11 is equal to 1, output 1 to logic8.
- Logic8: if logic3 is equal to 1 AND select5 is equal to 1, output 1 to count7.
- Count7: Increase value on “off to on” and output to math2.
- Select11: channel 1.
- selRe: channel 1.
- Logic4: if selRe is equal to 1 AND null11 is equal to 1, output 1 to logic9.
- Logic9: if logic2 is equal to 1 AND select5 is equal to 1, output 1 to count6.
- Count8: Increase value on “off to on” and output to math1.
- Select9: channel 1.
- selGr: channel 1.
- Logic2: if selGr is equal to 1 AND null11 is equal to 1, output 1 to logic1.
- Logic1: if logic2 is equal to 1 AND select4 is equal to 1, output 1 to count1.
- Count1: Decrease value on “off to on” and output to math1.
- Select10: channel 1.
- selFo: channel 1.
- Logic3: if selFo is equal to 1 AND null11 is equal to 1, output 1 to logic5.
- Logic5: if logic3 is equal to 1 AND select4 is equal to 1, output 1 to count2.
- Count2: decrease value on “off to on” and output to math2.
- Select11: channel 1.
- selRe: channel 1.
- Logic4: if selRe is equal to 1 AND null11 is equal to 1, output 1 to logic6.
- Logic6: if logic4 is equal to 1 AND select5 is equal to 1, output 1 to count5.
- Count5: Decrease value on “off to on” and output to math3.
- Select8: Button1 input and Button2 input.
- Select5: Button 1 input
- Select4: Button 2 input
- cntrlMenu: Select Xaxis input.
- Logic11: set to output 1 when value is greater than 0.
- Count12: Increase value on “off to on”, limit loops from minimum 0 to maximum 1, increment value set to value of cntrlMenu1.
- Null11: Output to logic2, logic3, logic4.
- Math1: Output value to Math4.
- Math4: ADD math1 to grBase, output to null1.
- Null1: Export to Value Send in ableGrowl.
- Math2: Output value to Math7.
- Math7: ADD math2 to foBase, output to null2.
- Null2: Export to Value Send in ableForce.
- Math3: Output value to Math8.
- Math8: ADD math1 to reBase, output to null3.
- Null3: Export to Value Send in ableReverb

Link To In Game Python Code on GitHub

List of Components

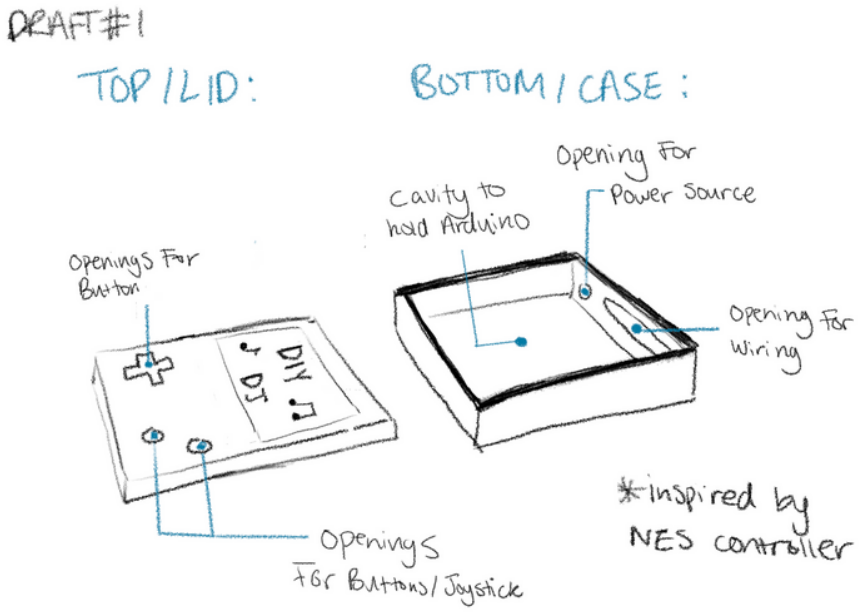
Software

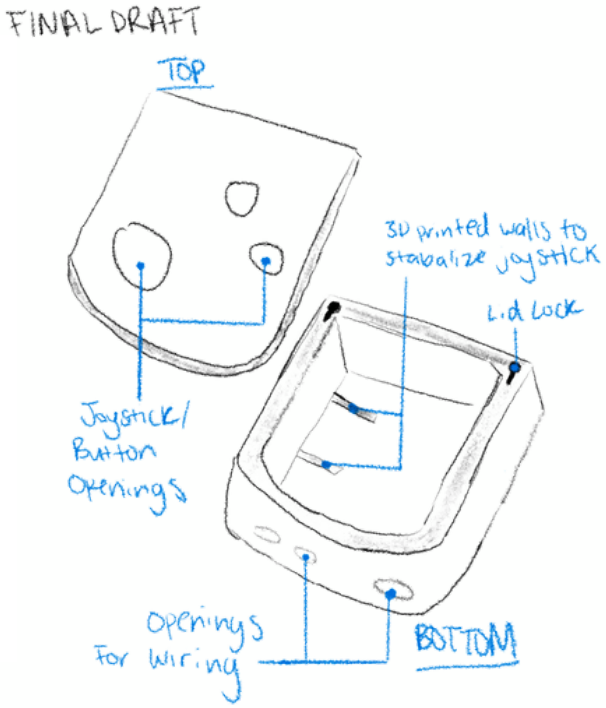
- Applications: TouchDesigner and TDAbleton Live
- Protocols: MIDI, Python, Python
- 3D Modelling: Blender and Tinkercad
- 3D Printing: Prusa Slicer

Hardware

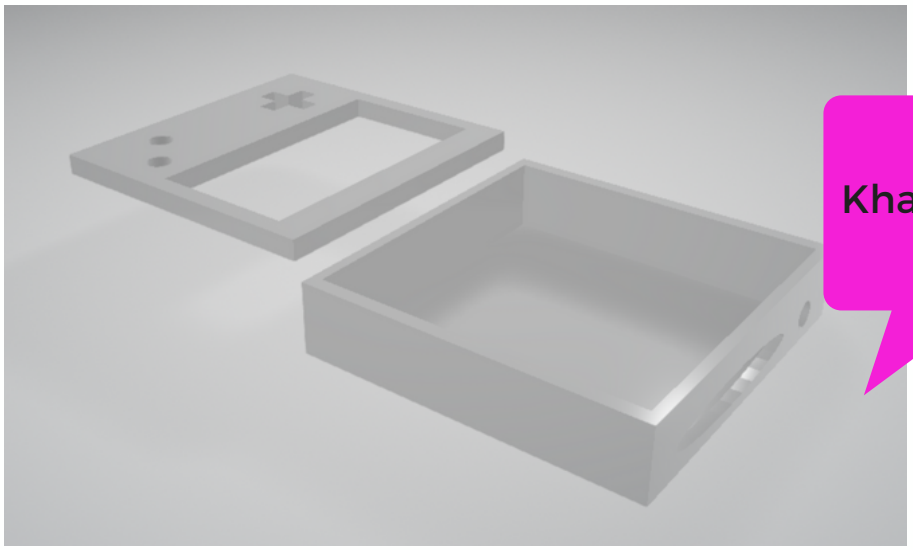
- Arduino Leonardo
- 4-Way Arcade Joystick
- 2 Buttons
- Jumper Cables

Building Sketches

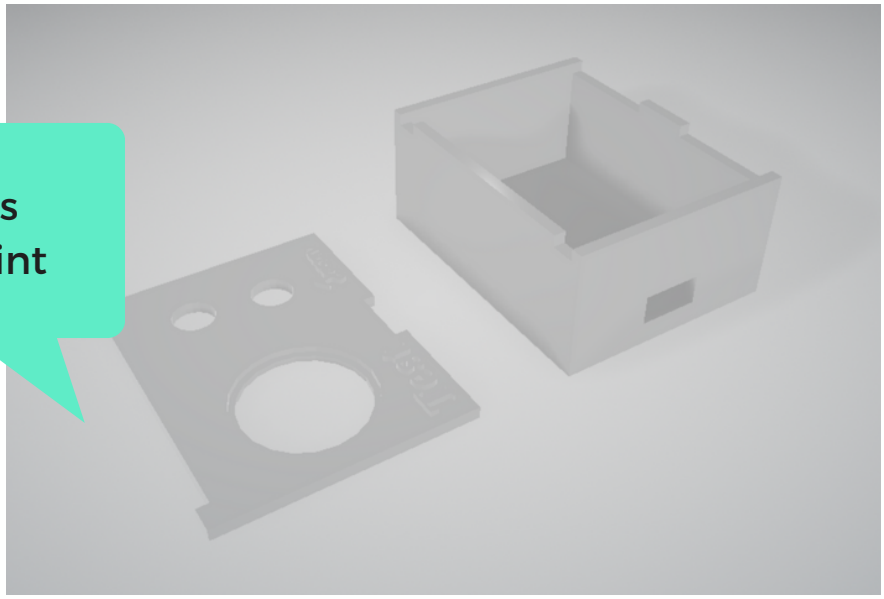




3D Test Print Models

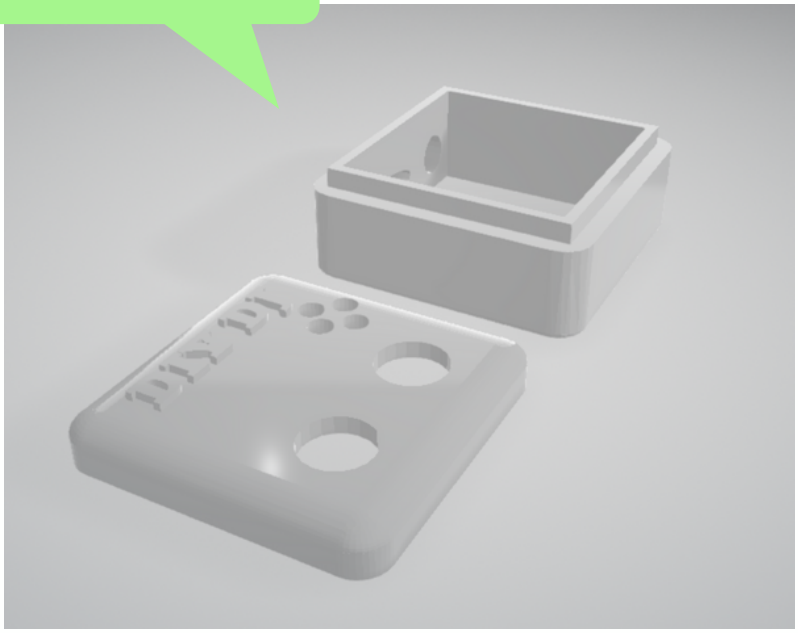


Khalifa 1st Draft



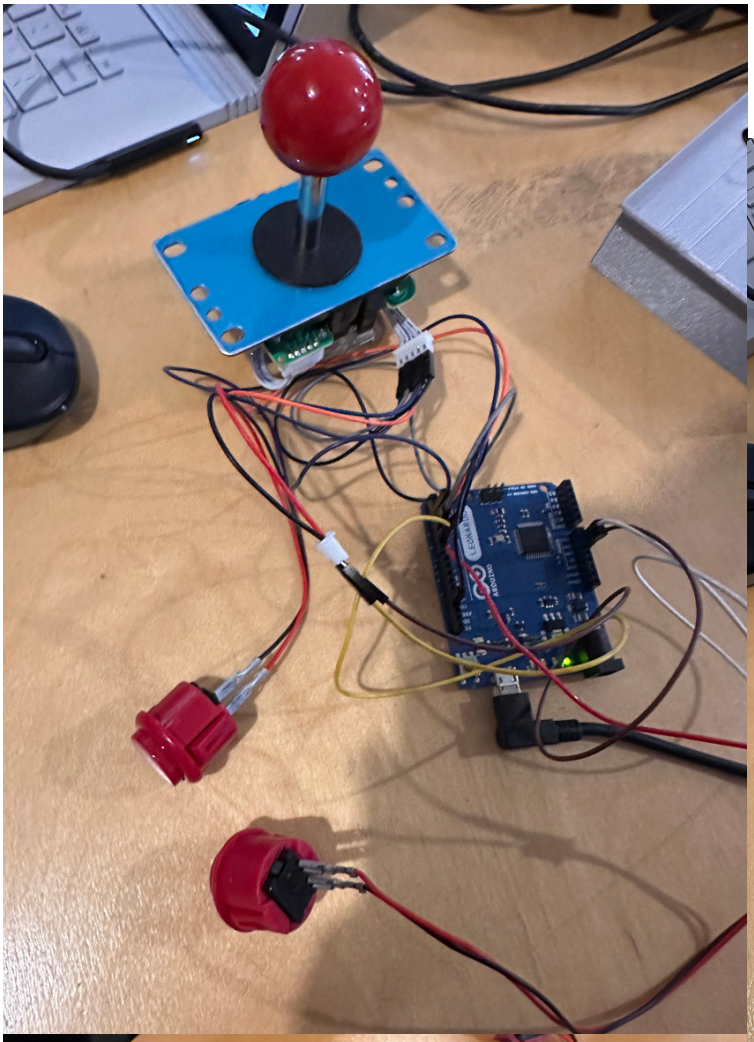
Jason's
Test Print

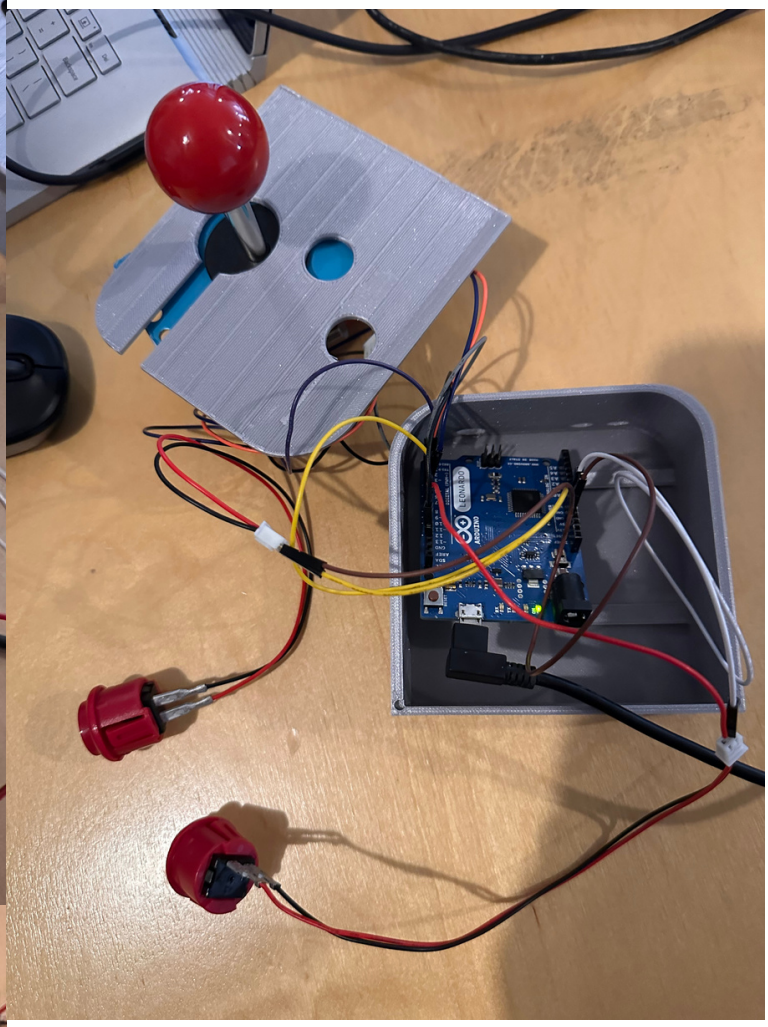
Angelina's
Test Print

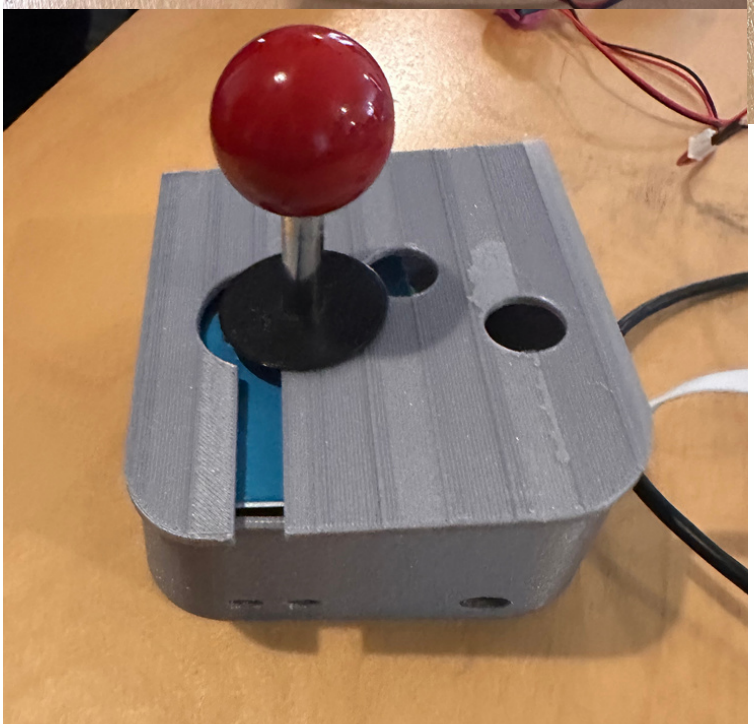


The controller will be printed in PLA. Since it is imagined that audio visualizations are quite colourful and vibrant, the colour of the controller will be neutral, similar to the original Nintendo NES colour scheme.

Building Process

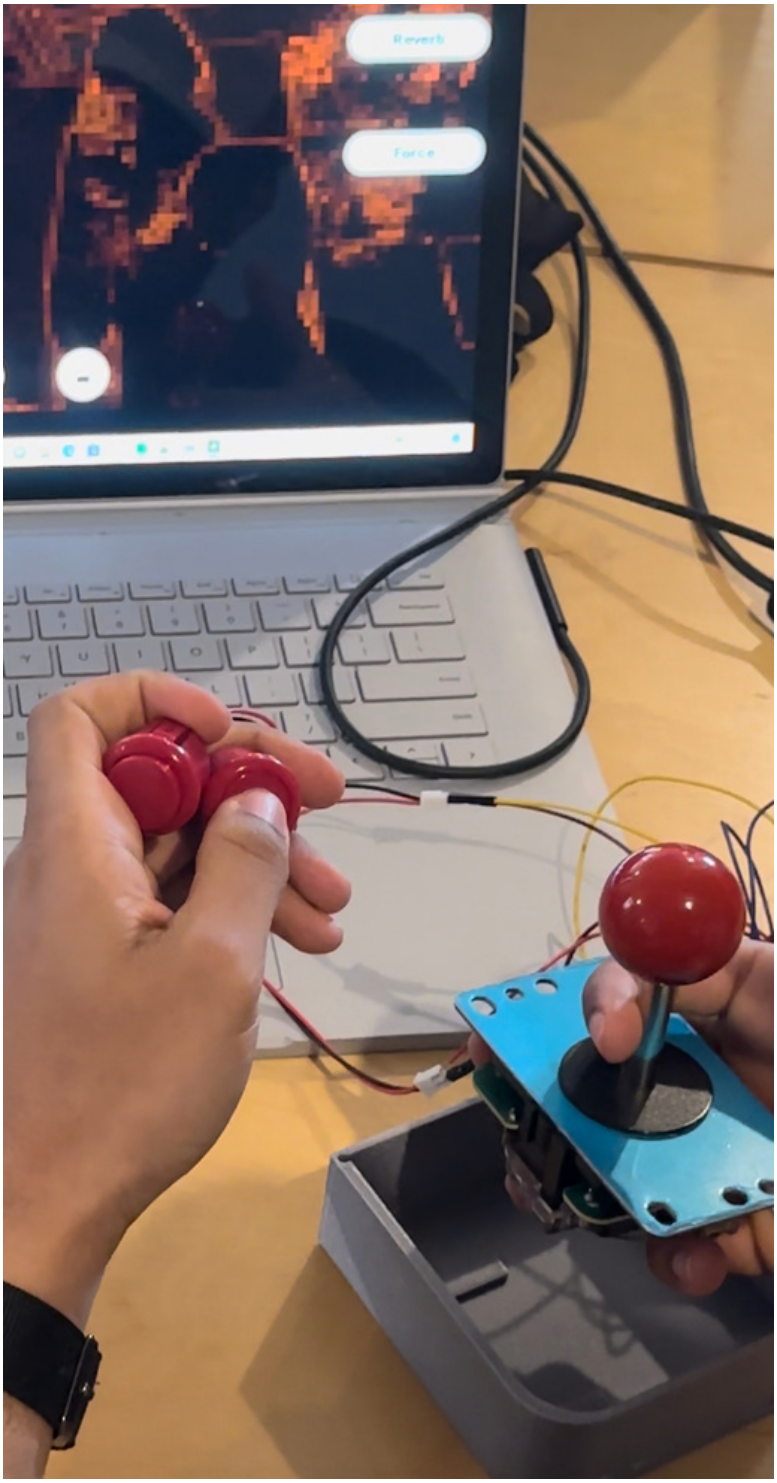


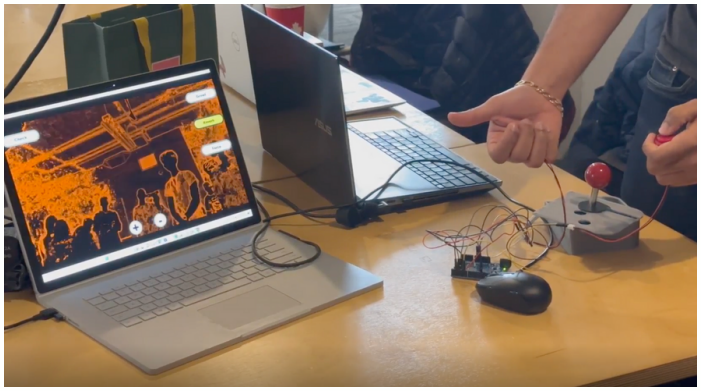




Device in Practice

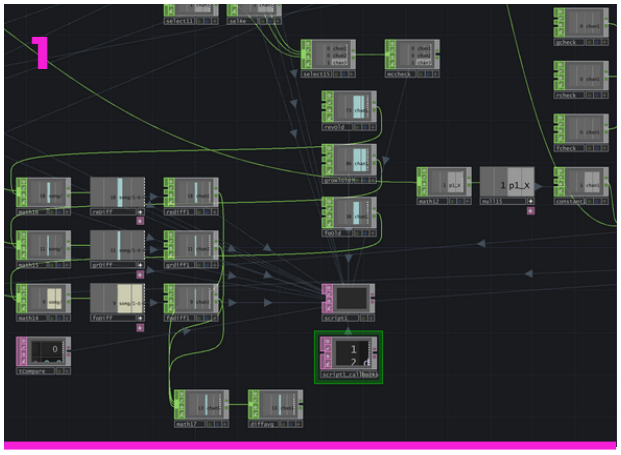
Link To Video DEMO



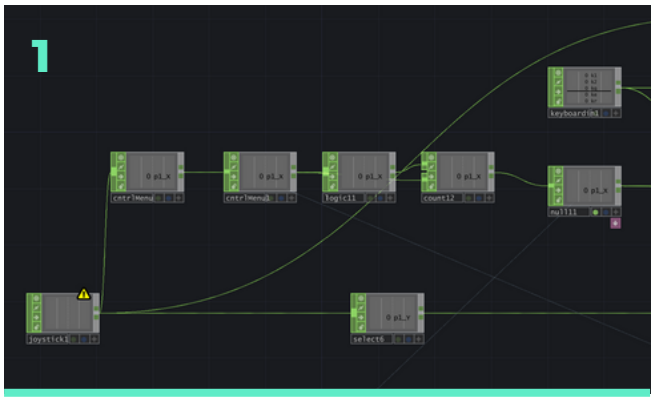


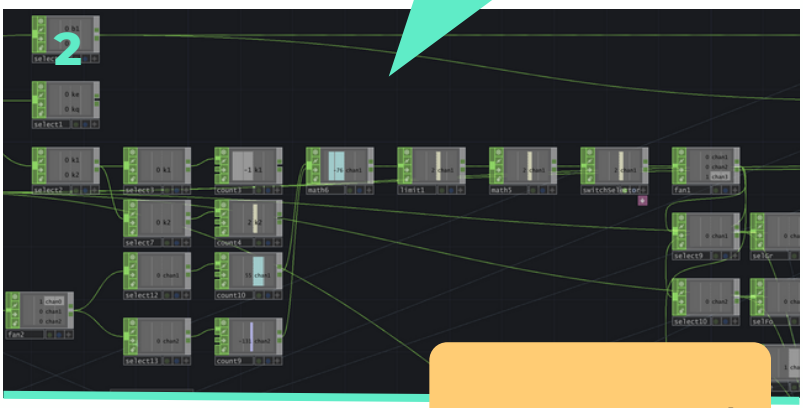
TouchDesigner File

Logic Code Snippets

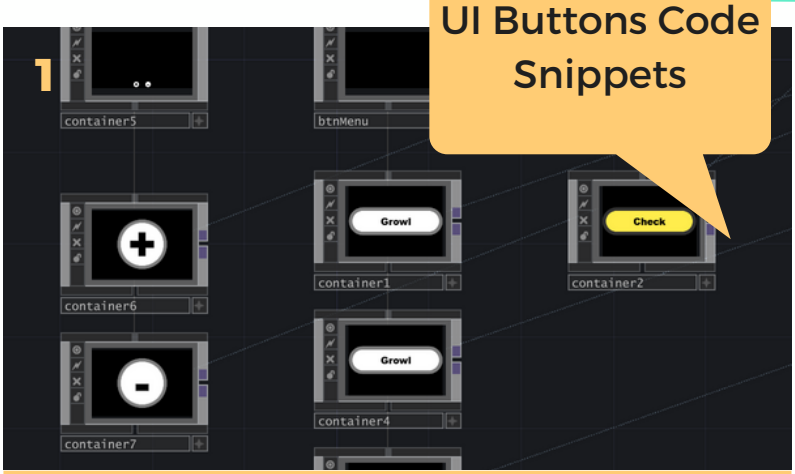


Controls Code Snippets

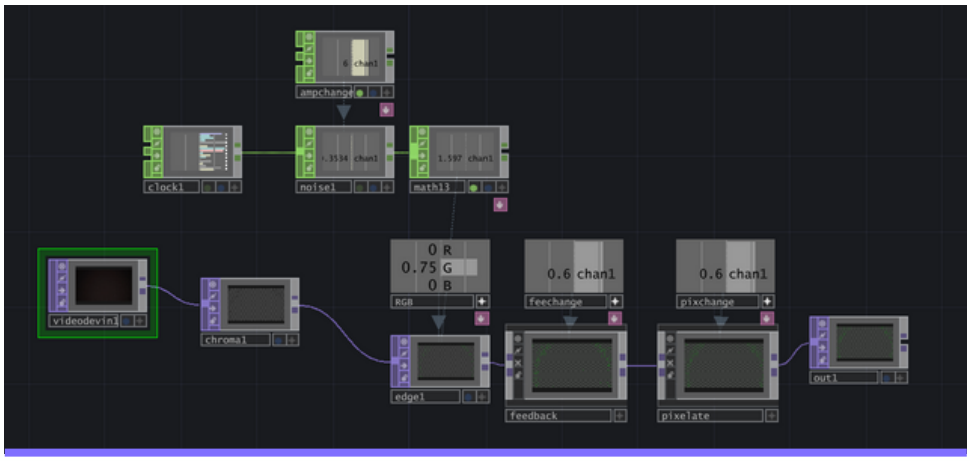




UI Buttons Code Snippets

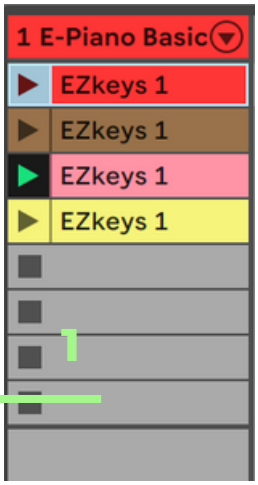


TouchDesigner File Cont.

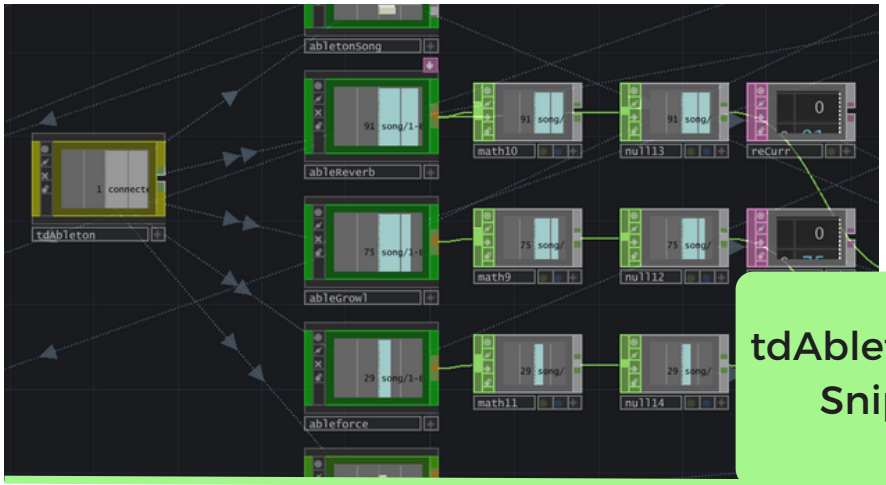


AR Webcam Code Snippets

2

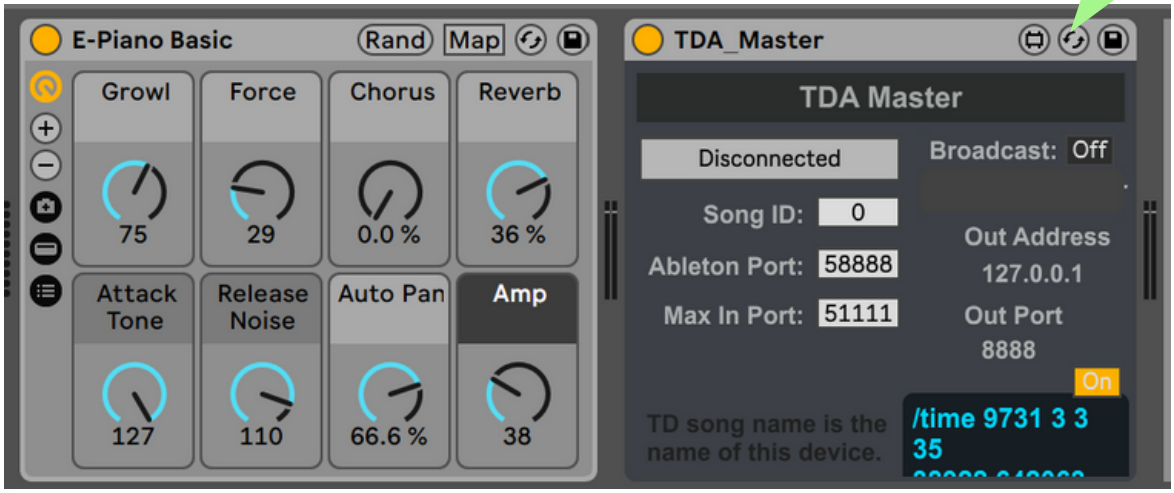


1

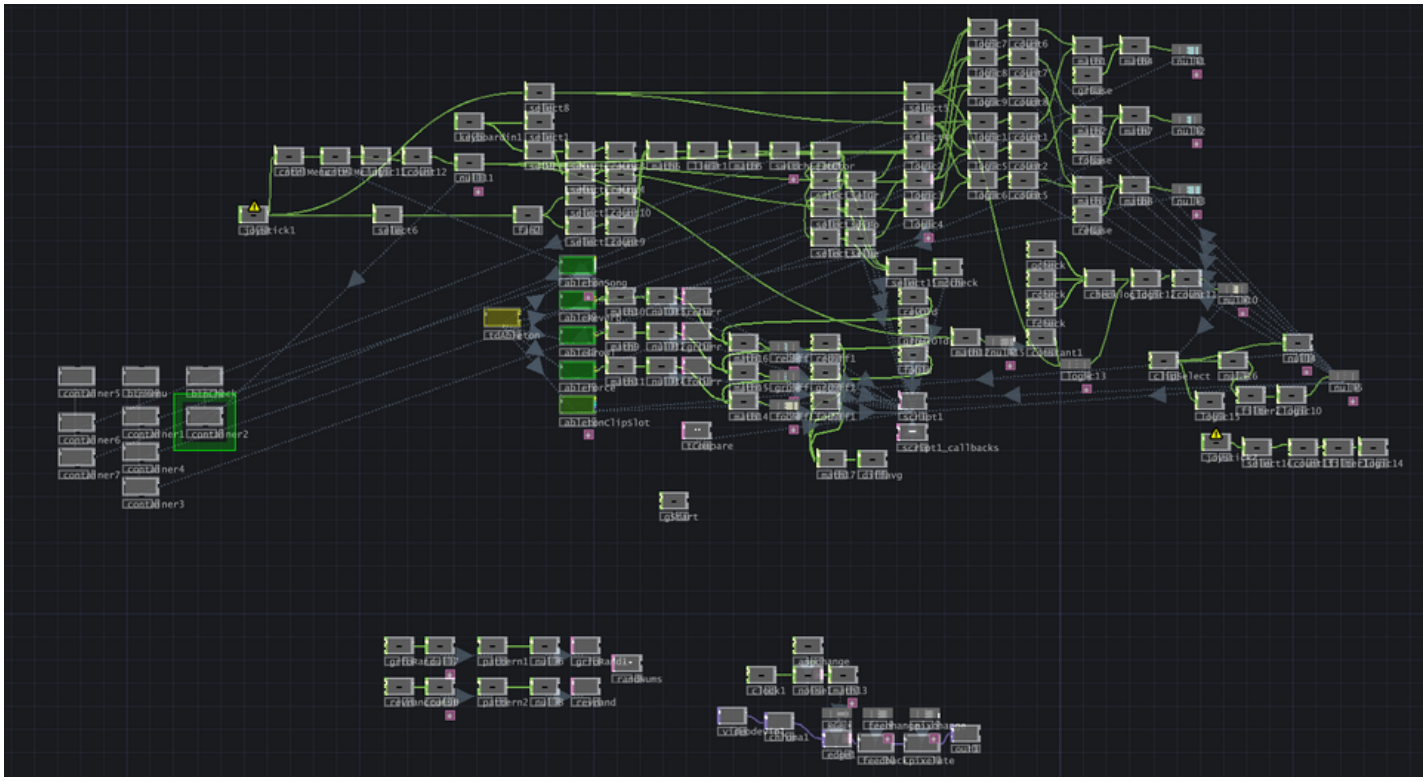


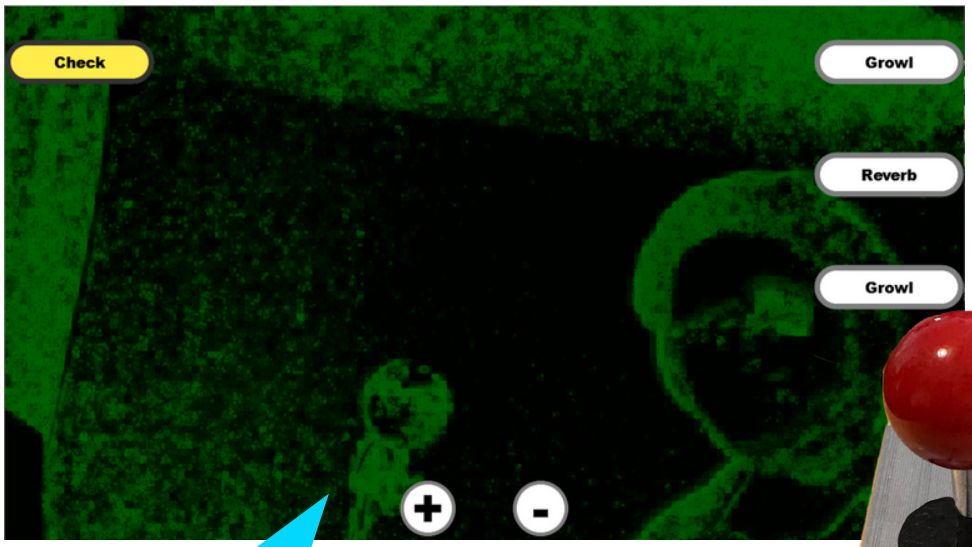
tdAbleton Code Snippets

3



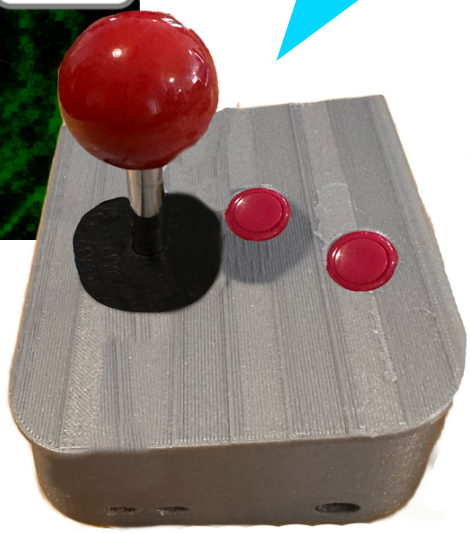
Final TouchDesigner File





Final Product (TD present mode)

Final 3D Printed Controller



Reference Material

3D Printed Radio Controller [1]
https://www.youtube.com/watch?v=YMF5NXeHOnk&ab_channel=Electronoobs

This video features a 3D printed RC controller, this is taken as inspiration as it features a 3D printed case for an Arduino UNO/NANO encases joysticks and buttons which is the premise of what we would like to create for our audio visualisation game.

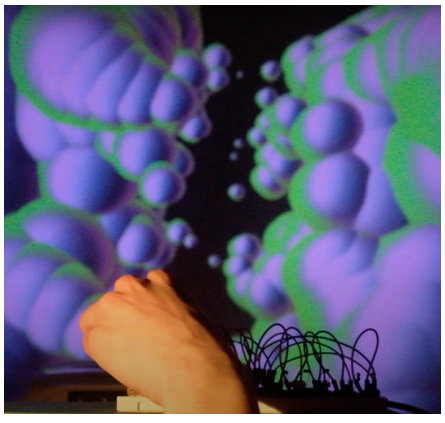
Controlling visuals in real time / Arduino MIDI + Touchdesigner [2]
https://www.youtube.com/watch?v=SVgqv7m1iAk&ab_channel=parametrip

This resource features audio reactive visuals controlled by MIDI and TouchDesigner which may be interesting to explore for the musical component.

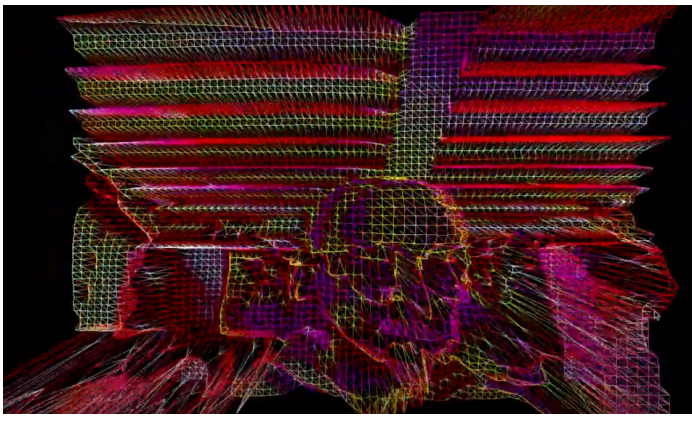
Biometric Music Systems with Audio reactive Webcam [3]
https://www.youtube.com/watch?v=td0hbSbGpGY&ab_channel=THSOUNDART

This resource features webcam tracking with audio reactivity. Although not done through TouchDesigner and Arduino, we are inspired by this concept and hope to apply this to our own project.

[1]



[2]



[3]

Summary and Reflection

The purpose of this project was to create a "musical instrument" or audiovisual electronic device. Throughout this process, we faced a variety of challenges including using Python in TouchDesigner and even the final fabrication of the 3D model. Overall, we were able to complete a successful video game with functioning components with minimal errors. The task at hand was completed as both the hardware and software communicate with each other.

In terms of next steps, we would definitely want to incorporate a completed prototype of the 3D model if more time was available. Additionally, during the critique Kyle mentioned we could have tried using a different microcontroller more suited towards arcade style games which would have made our process a little bit easier.