Berkleee College of Music

The Invisible Carnival:

An Ecosystem of Custom Laser Controlled Devices

Final Paper

Submitted in Partial Fulfillment of the Degree of

Master of Music in Production, Technology, and Innovation

Supervisor: Nacho Marco, Marta Verde

by Juan Pablo Gomez

Valencia Campus, Spain

December 2020

Table of Contents

Abstract

The project aims to create a visual ecosystem of analogue devices that will be used by Fortuno(the artistic name of the author) in live performances and during production. Through the development of two MIDI laser controlled devices, the artist will be able to use lasers to control various parameters of the sound. The project also aims to demonstrate Fortuno's capabilities as a sound designer, an artist, a sound engineer, and a live engineer. Hence, he will use these devices for his electronic production and then utilize the sounds in his set. An Arduino microchip will be used to program the laser response for each device. The laser MIDI controller will be created mainly through the use of programing using C++/Arduino. Each device will be molded separately with different types of casings used. Each device will be tested and then implemented in the sound and image of Fortuno. Ultimately, making him stand out as a unique Dj and musician.

Keywords: MIDI, music sound design, live performance, digital production, electronic production, C++, Microchip, ecosystem.

1. Introduction

The main purpose of the project is for Fortuno to innovate himself as an artist/DJ and as an engineer. To achieve this goal, he had to break down both categories, and dissect aspects of past innovations from both fields.

As an engineer, he looked towards synthesis and hardware devices as his most unique skills come from a deep understanding of electronic components and audio processing. Thus, Fortuno looked at some of the most innovative hardware synthesizers and the history of modular as a whole. He realized that to innovate he would not only have to improve existing hardware but add a unique twist to it. Hence, Fortuno decided in creating an ecosystem of laser musical devices that have both a visual and a practical benefit.

As a DJ he looked towards his biggest inspirations such as Stephan Bodzin and Eric Prydz. Realizing that a huge part of their innovation was implementing unique devices to their performances. Stephan Bodzin developed his own MIDI controller[1] and Eric Prydz(along with various artists) created the holosphere(A 3D visual holographic sphere)[2]. Thus, Fortuno decided to build upon these ideas and take a look into what was missing in his setup. He ultimately came to the conclusion that he would need an ecosystem of devices. At the same time this gear would have a visual aspect which would entertain to the audience. The idea of creating this ecosystem of devices is to provide a sense of continuity and a unique complete image in the mind of the viewers.

[1] V.A, “Stephan Bodzin's PO10 LiveController,” Stephan Bodzin's PO10 LiveController | Equipboard®, accessed December 20, 2020, https://equipboard.com/pros/stephan-bodzin/po10-livecontroller.

[2] Dani Deahl, “Eric Prydz Is Going to DJ inside a Giant Glowing Sphere - Here's How It Was Made,” The Verge (The Verge, July 9, 2019), https://www.theverge.com/2019/7/9/20683461/dj-eric-prydz-epic-6-0-holosphere-tomorrowland.

Below are the figures of the initial intended devices for this project. The first image is the front panel of the module rack, called the Sequen-Vicity 5059. The images after that are of the intended MIDI laser controller originally called the MIDI 2 U Invisi-Controller.

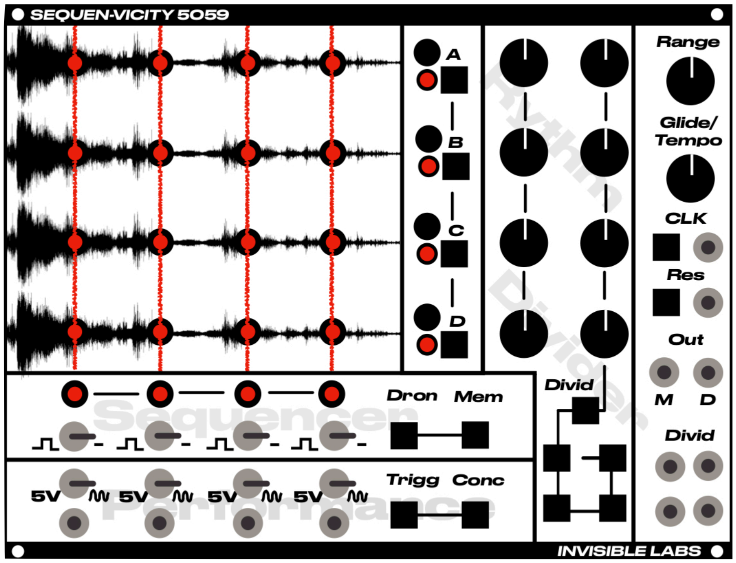


Figure 1. Front panel of the Sequen-Vicity 5059

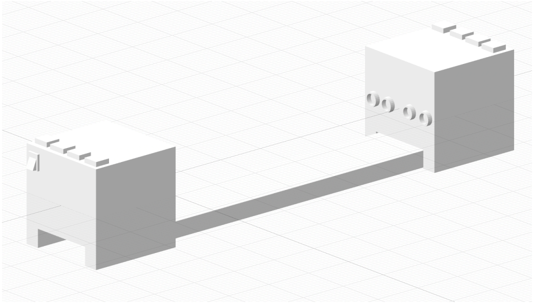


Figure 2. MIDI 2 U Invisi-Controller Top

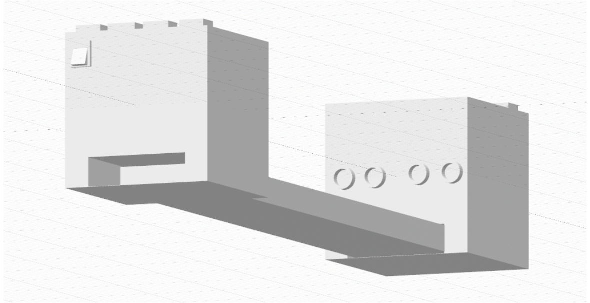


Figure 3. MIDI 2 U Invisi-Controller Bottom

Ultimately, this project aims to make Fortuno the only DJ and engineer to design by himself a varied ecosystem of complex analogue equipment that drive his visual and audio presence forward. He will find unique ways to perform as no one will have the devices he has.

2. Review of the State of the Art

There were many inspirations that affected the development of the project. These include artists such as Stephan Bodzin, Boris Brechja, Ben Bohmer and Jean-Michel Jarre. They also include synthesizers and modular racks such as the Invisi-Synth, the Moog Grandmother, Maths and Pressure Points.

To innovate, and to push himself forward, Fortuno had to look at the past; both as an engineer and an artist. That is why the first inspirations for this project were the Invisi-Synth[3] and Stephan Bodzin[4]. Stephan Bodzin inspired him to change the typical DJ setup with unique personality. Fortuno started implementing this mentality by using his own MIDI controllers on top of the DJ controller, however he desired to go beyond that as it currently was not as innovative. Moreover, by having researched about controllers and their usage of lasers, Fortuno concluded that a device with lasers would still be very unique today. Hence he decided to build these devices with the idea of lasers in mind. At the same time Fortuno started to produce more music in the style of Ben Bohmer[5] and Boris Brechja[6]. Inspired by the way Ben Bohmer uses synths in his sets and Boris Brechja's style of music, the artist decided to use his Moog Grandmother more

[3] Invisi-Synth Research Paper: Juan Gomez, â€œSound and Technology,â€ Portfolio, 2020, https://totumag.wixsite.com/portfolio/copy-of-circuit-design

[4] Stephan Bodzin: Resident Advisor, "Stephan Bodzin," Resident Advisor, June 18, 2021, https://www.residentadvisor.net/DJ/stephanbodzin/biography.

[5] Ben Bohmer: Anjunadeep, "Ben Böhmer - Breathing," Anjunadeep, 2019, https://anjunadeep.com/ben-bohmer/.

[6] Boris Brechja: Boris Brechja , "Boris Brejcha," Boris Brejcha, 2018, https://www.borisbrejcha.de/.

extensively and really explore its intricate possibilities. There began to be connections to his past as an electrical engineer and from there he invested his time in modular synthesis.

A strong passion with modular became apparent and the project shifted into creating a personalized modular rack that uses lasers to design interesting sounds live. An important rack for this development is Maths by Make Noise[7]. It affects the CV voltage in mathematical ways and derives, integrates and sums multiple signals. Moreover, it is the most popular rack built today. Thus by learning about its success and Make Noise Pressure Points[8] the essence of a good module was determined; to provide a creative and unique way of modulating sound. Nacho Marco at the time also recommended to look into Jean Michel Jarre[9]. He inspired the idea of adding a basic laser MIDI synth in the performance to create unity and be visually stunning.

Moreover, some preliminary research was made to conclude if the module and performance would be unique. After looking at various artists and gear it was concluded that there is no modular rack that uses lasers and no DJ that uses his/her own built/designed modular rack.

The combined inspirations of the artists and the gear led to the success of the idea.

[7] Math Make Noise: Make Noise, "MATHS," Make Noise Co., 2012, https://www.makenoisemusic.com/modules/maths.

[8] Pressure Points Make Noise: Make Noise, "Pressure Points," Make Noise Co., 2012, https://www.makenoisemusic.com/modules/pressure-points.

[9] Jean-Michel Jarre: VA, "Biography.," JeanMichel Jarre, 2020, https://jeanmicheljarre.com/biography.

3. Description

However as time progressed the project evolved and changed form, The invisible carnival ended up consisting of two laser controlled MIDI devices. Each controller designed for a specific task. The first controller called the Mystery Box is an all in one MIDI controller. Please refer to figure 16 to see the full body of the Mystery Box on page 22. The front face consists of five separate knobs. The orientation of the knobs follows a Moog filter setup with two knobs on the side. On the top there are six buttons that can be customized for any purpose.

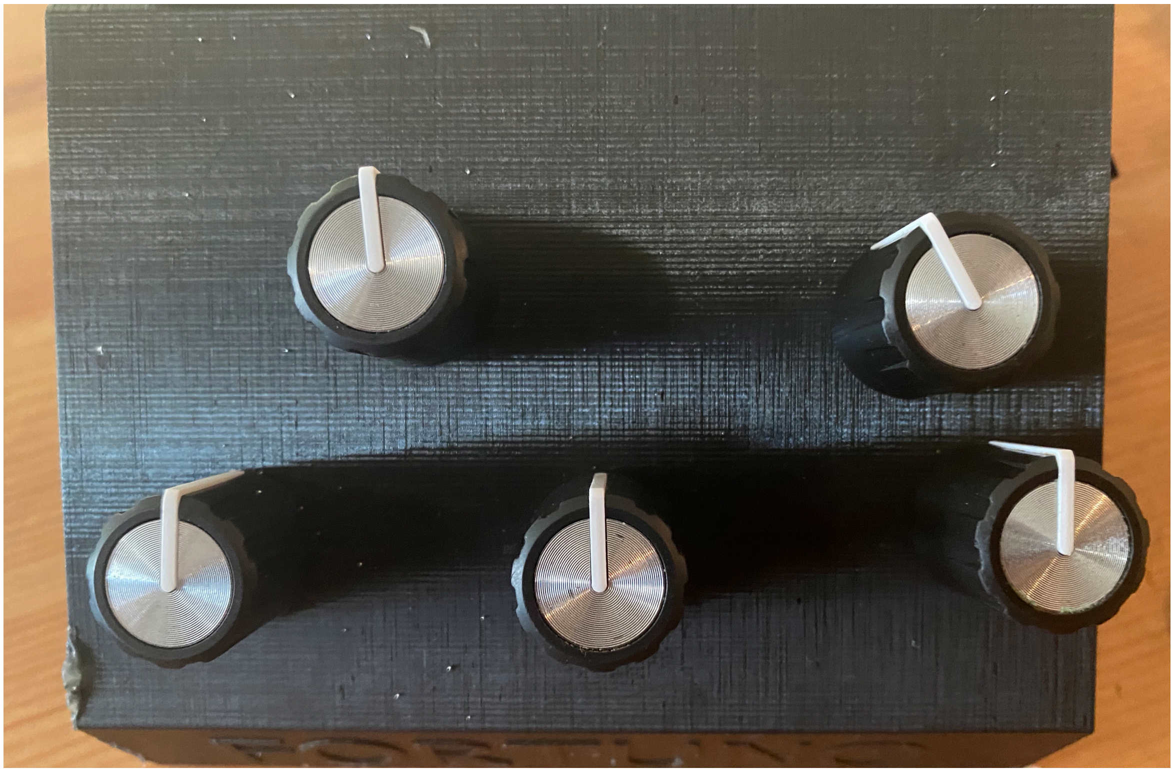


Figure 4. Front panel of the Mystery Box

The two buttons on the side ideally work in conjunction with the knobs below and the other four buttons in conjunction with the four lasers.

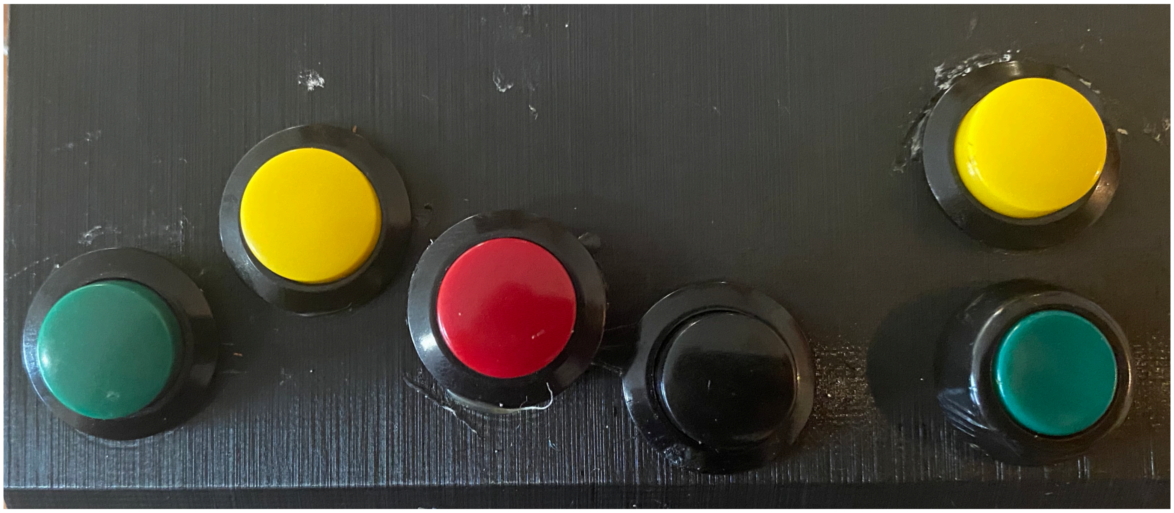


Figure 5. Top panel of the Mystery Box

The last part of the Mystery Box that works as a MIDI control are the four lasers that. follow. To improve the responsiveness of the device it was decided that the two lasers on each side work together. Thus in order to activate the effect both of the lasers have to be intercepted.

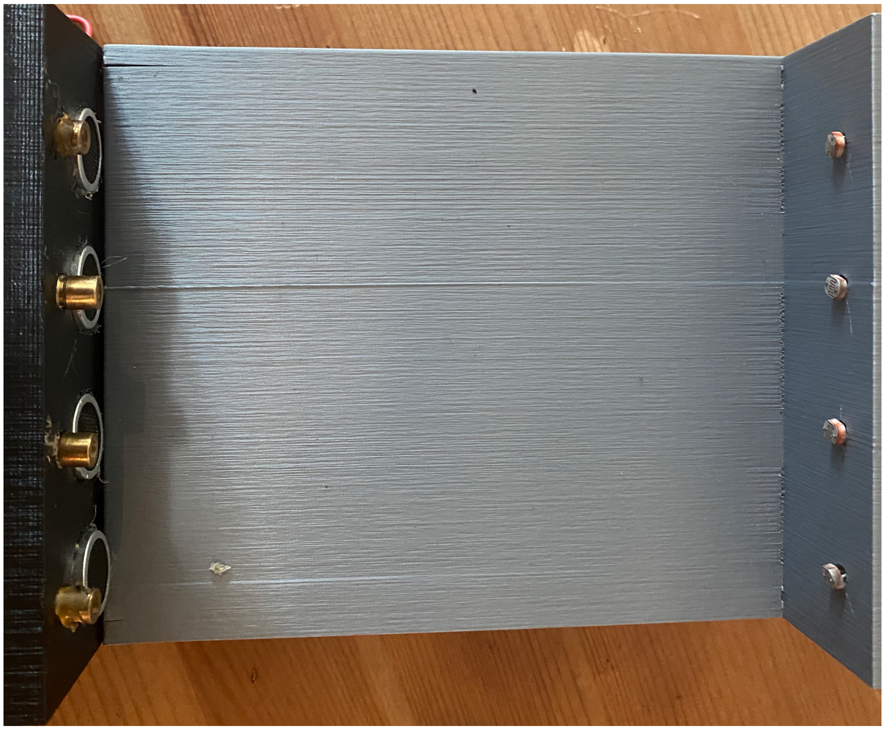


Figure 6. Top laser panel of the Mystery Box

Below is an image of the MIDI controller in the dark with the lasers activated.

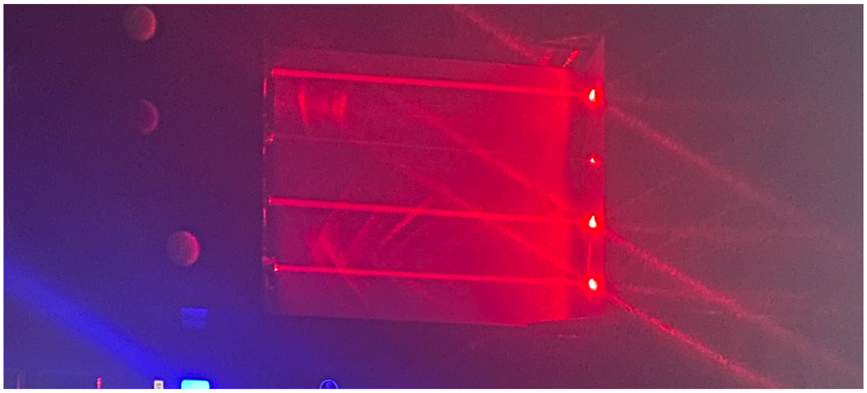


Figure 7. Top panel of the Mystery Box lasers on

The second MIDI controller called the Ring of Fortune is a ring device that uses lasers to control MIDI. Below is an image of the Ring both on and off.

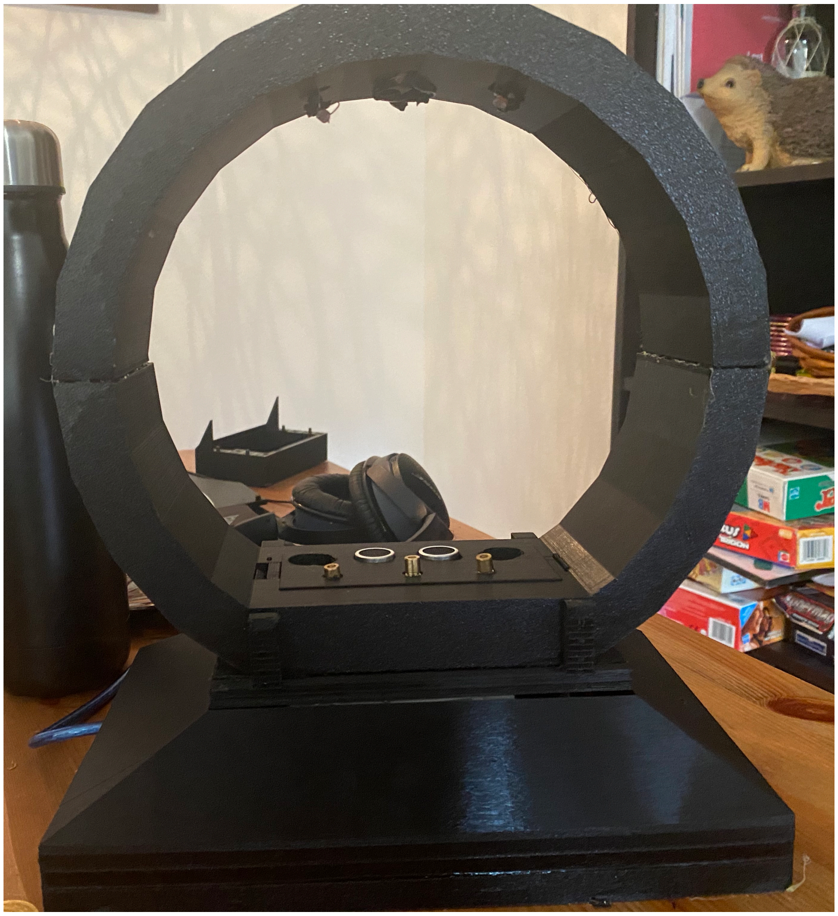


Figure 8. Front view of the Ring of Fortune

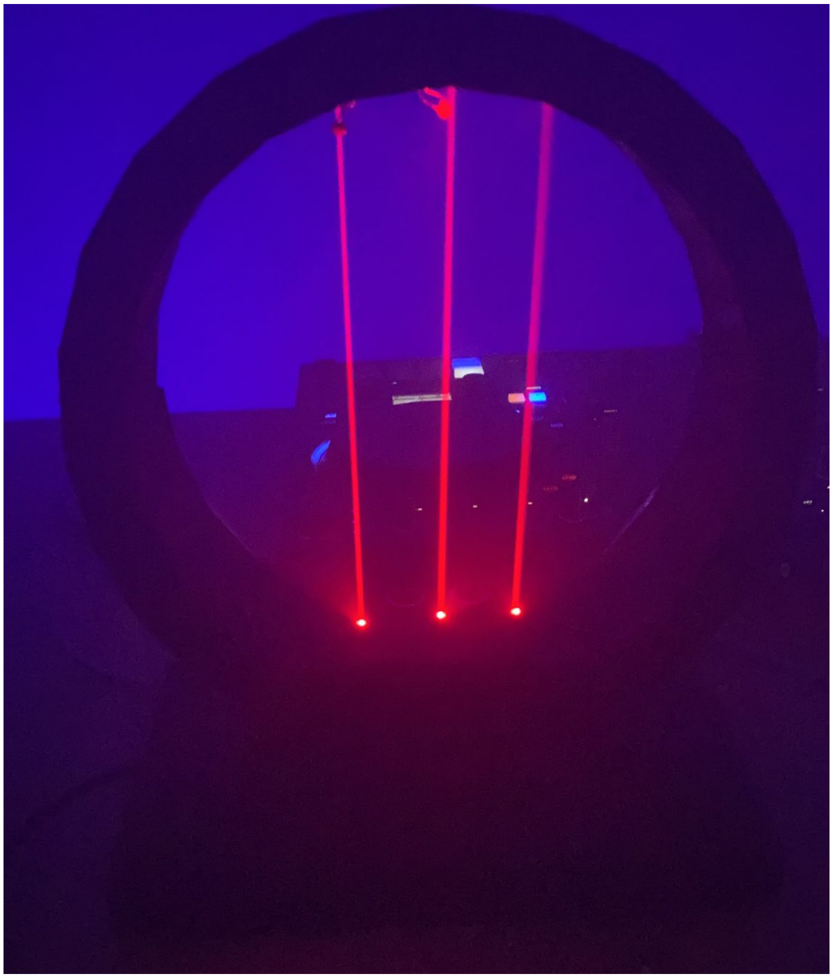


Figure 9. Front view of the Ring of Fortune lasers on

This controller is made specifically for the live stage performance. The laser in the middle controls a low/pass filter. The laser on the right activates a reverb effect. The effect on the left activates a delay effect. All three can be used at the same time and mixed and matched. Please refer to figure 14 and 15 for images of this device being used in a live performance.

The project consists of using both laser MIDI controllers together to perform during a set. And to use the mystery box both in live sets and when producing.

Both MIDI controllers where developed using Arduino as the processing unit. Many other types of smart chips were researched such as the Raspberry PI. The Arduino was chosen for various reasons. One of them being substantially cheaper. The raspberry PI sits at 60 euros and above while the Arduino Leonardo at around 22 euros. The Arduino Leonardo is also a chip specifically capable of MIDI usb, other Arduino chips such as the Arduino Uno are not capable of this communication protocol. The usb MIDI library found in the Arduino website is also simpler to use than that of the Raspberry PI.

The mystery box uses potentiometers as knob MIDI controllers. They are the basic tool used in all analogue gear to control these types of parameters. The buttons used are simple switches that either let or deny the passing of current. These type of buttons are standard. There was no need of pressure controls for this device. Moreover a multiplexer was used to have the required amount of analog entries.

To construct the laser system used in the Mystery Box and the Ring of Fortune a laser diode, a light diode and a distance sensor was used. The laser dioxide chosen is a red emoting laser. This was chosen as it is the cheapest option and smallest in size. The green diodes are bigger and can cause more eyesight damage. The diode sensors are the standard when creating devices that react with light. The distance sensor chosen was decided from two sensors, a sonic distance and an IR distance. The IR sensors had a more precise reading of distance, however when used with smoke the sonic sensors would out perform the IR sensors. Ultimately leading to choosing the Sonic sensors. Both of the designs were also chosen strategically. The mystery box was designed to be Fortuno's multi-use device. To be what Fortuno was lacking in front of him when performing live. The Ring of fortune was designed to be a visual spectacle and visually pleasing. Making the viewers want to play with it. Having to put your hand in the ring is also on purpose, this creates a sense of power and control. Glue was used to setup most of the circuits in place, next time use magnets.

However to make it all work, a lot of code had to be developed. Fortuno started by using a MIDI hub function that makes the Arduino work in MIDI and be able to send MIDI messages. From there Fortuno learned to use boolean functions to create statements, and mapping functions to make the ranges change such as that of the potentiometer that went from 0-1048 to 0-127 MIDI. Boolean functions and if/else statements were used to develop the button switches, as well as the laser interception. Other functions such as that of the multiplexer and distance function were used to make the device usable. Lastly, as there are two devices connected, Fortuno had to realize that each device had to have their own separate MIDI messages that do not intercept. After this was realized, both devices worked perfectly in unison. You can see the fully developed code in the appendix from page 16 to 21.

4. Innovative Aspect

There are various ways of innovating. One of them is to combine two simple ideas into one. Constructing something new by taking pre existing ideas and developing them to take a unique and successful form. That is the essence of what Fortuno has created and what follows. The invisible carnival is an ecosystem of endless possibilities. Fortuno has only scratched what is possible with MIDI laser devices. The straits that have tried implementing lasers into their performances do it large scale, and use them mainly to play notes. Moreover these are not devices but installations that work with lasers. Fortuno took these ideas and compacted them. Moreover, a person had developed a laser controller in the shape of an X that was very weird and had very little success. My laser MIDI controller scream simplicity and effectiveness. The idea came from combining a laser harp with a MIDI controller. Ultimately Fortuno is the first to blend these two ideas into one simple, and visually stunning idea. Moreover as mentioned the invisible carnival has a lot of space to grow. The idea also of being the DJ that develops and constructs his own MIDI controllers is unique. There are Djs who have custom built controllers made by other third parties of manufacturers.

5. Skills Acquired

Throughout the process of developing these new MIDI controllers and Fortuno many skills were gained. The biggest skill is prototyping and developing a fully functional device. Advanced coding techniques such as Boolean functions. Designing in 3D and printing in 3D. Music production skills such as parallel mixing, resonant eq-ing, eq distortion, and sound designing. Moreover even though the sequencer idea fell through, skills such as how to construct an Arduino analogue pitch control module and an Arduino LFO module were gained. With these skills different module can be designed.

6. Expected and unexpected difficulties

Throughout the process of developing the project, many expected and unexpected issues arose. One of the biggest unexpected challenges was what that there was no way for the Arduino to send a voltage between 0-5 at a constant DC. This meant that there was no way of using the lasers' digital sensors to control the pitch of a sequencer properly. It could reach voltages in that range with an AC current which created issues and made it not reliable. Once this issue was noticed, the focus on the modular gear was decreased. And hence a bigger focus was made on the laser aspects of the MIDI controllers. This revelation served as the birth point of the Ring of Fortune. There were issues with the mystery box where cables would touch each other and interfere. Fortuno had to reopen the device and resolver the connections as well as place rubber holders to not make them interfere. An expected issue arose after having done the first demo of the laser MIDI devices with smoke. The smoke made it very hard for the performer to be seen. Hence there is now a fan involved in the process.

7. Future Ramifications

Moving forward, the invisible carnival’s main goal is to grow. Second iterations of the devices will be built. The second Mystery Box will have magnetic connections and more soldered parts. The second iteration of the Ring of Fortune will have a string of LEDs around the ring. A button will be pressed the lights will shine in the shape of a semi circle they will light up changing the effect of the laser at random depending on which led light the it stops on. Moreover more devices will be built with different functions such as a laser synth. It is desired to continue exploring the idea of a carnival ecosystem and connecting it with nature in future iterations. Steps will then be taken to patent the laser MIDI idea to maintain exclkusitivity. After using it for the Fortuno brand, it will be sold with the Fortuno Tech part of the brand. Fortuno is in talk with a hardware audio provider in Brasil (Boomerang Audio) to develop a Laser MIDId controller.

8. Conclusion

After having used the devices in a performance, the reaction Fortuno was expecting from the ecosystem was achieved. Every person who watched the Mystery Box and the Ring of Fortune in action would immediately smile, say it's cool, and proceed to either talk about it or suggest to use it in a way. This is the reaction Fortuno was hoping for and by such it is a success. Fortuno has always been a creator since a young age, to combine this with his musical journey and himself as an engineer is where the success of all of it comes from. To demonstrate that the unification of these worlds to develop Fortuno's brand is possible. This project has also bee successful at making Fortuno unique and pushing him to create better sounding music. Now he can use these skills and devices to further his career and in the future build upon the ecosystem of lasers.

Appendix

Table 1. Projected Timeline

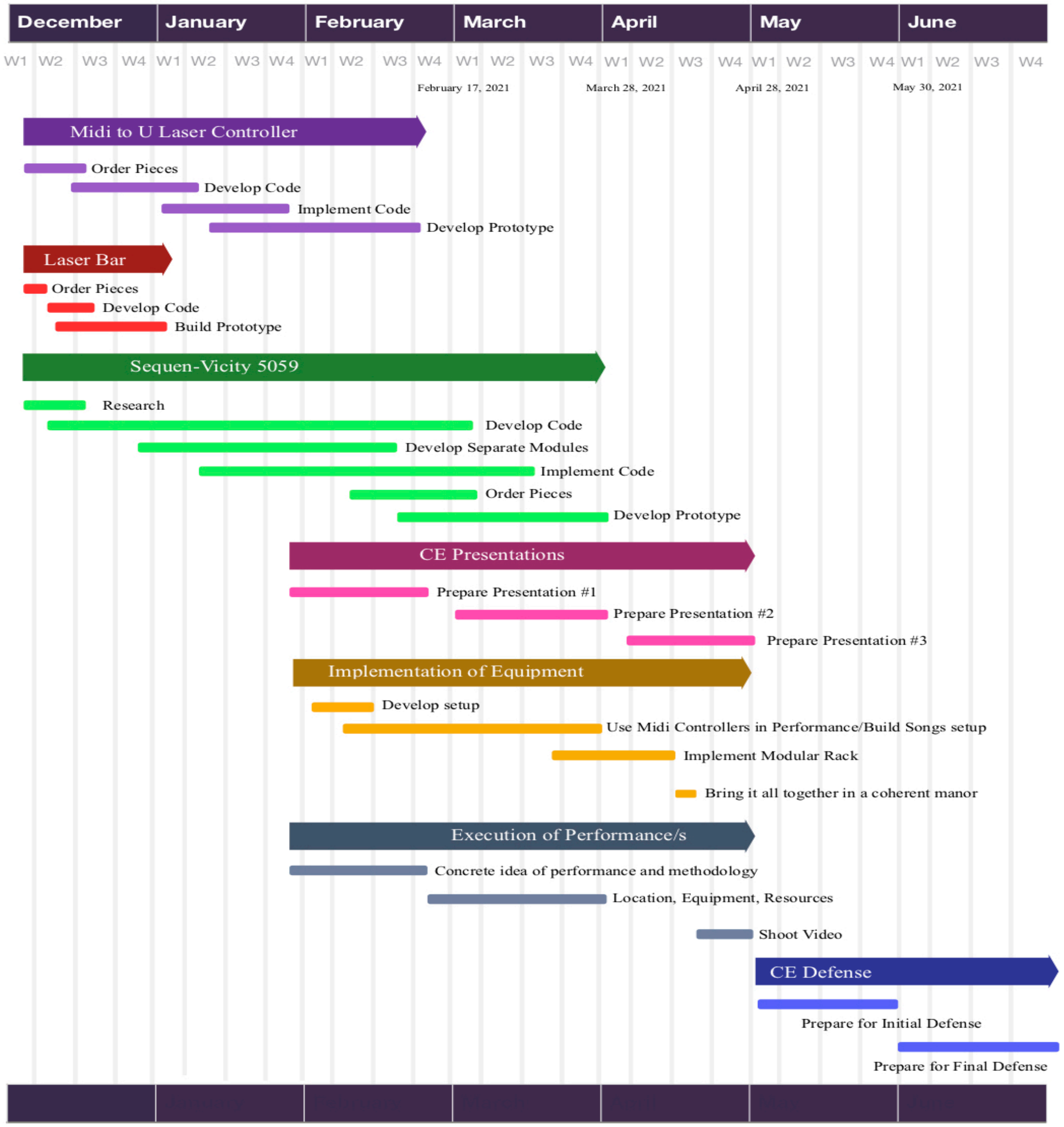


Table 2. Projected Budget

ITEM	Amount	PROPOSED	REAL
MATERIALS (disposables)			
Hard Drives		$400.00	$0.00
Memory sticks		$25.00	$0.00
Potentiometers	12	$3.00	$0.00
Electric Paint		$20.00	$0.00
Arduino/Microchip	3	$300.00	$200.00
Laser	10	$6.00	$6.00
Receiver	10	$6.00	$0.00
Sound Echo		$5.00	$5.00
Op amp	4	$2.00	$2.00
Switches	8	$2.50	$2.50
LED	24	$2.50	$2.50
Digital buttons	19	$2.50	$2.50
3.55 mm Audio ports	12	$5.00	$5.00
PCBs	3	$15.00	$0.00
Cables	20	$5.00	$0.00
EQUIPMENT			
HARDWARE			
Oscilloscope		$380.00	$0.00
DMM		$30.00	$0.00
Soldering Rack and material		$50.00	$0.00
3D Printer and material		$1000.00	$0.00
INTERFACE (purchase) 120 days		$400.00	$0.00
COMPUTER (purchase) 120 days		$2,500	$0.00
CAMERA (rental) 10 days		$1,200	$0.00
SOFTWARE			
ABLETON (purchase) 120 days		$400.00	$0.00
PERSONNEL			
MUSICIANS # of players x rate x hrs			
Videographers 1		$200.00	$0.00
Videographers 2		$200.00	$0.00
DIRECTOR		$300.00	$0.00
MAKEUP		$50.00	$0.00
Studio/Lab			
BERKLEE daily x 30 of days		$1800.00	$0.00
HOME daily x 60 of days		$1000.00	$0.00
CATERING			
MEALS cost/1 person *3 meals a day	3 Months	$800.00	$800.00
OVERHEAD			
RENT	3 Months	$1500.00	$1500.00
POWER	3 Months	$100.00	$100.00
WATER	3 Months	$50.00	$50.00
GAS	3 Months	$50.00	$50.00
INTERNET	3 Months	$0.00	$0.00
PHONE	3 Months	$20.00	$20.00
TOTALS		$12,829.50	$2,745.50

Figure 10. Arduino Code for the Mystery Box

```
 #include "MIDIUSB.h"


 #define S0 10                  /* Assign Multiplexer pin S0 connect to pin D0 */
   #define S1 11                  /* Assign Multiplexer pin S1 connect to pin D1 */
   #define S2 12                  /* Assign Multiplexer pin S2 connect to pin D2 */
   #define S3 13                  /* Assign Multiplexer pin S3 connect to pin D3 */
   #define SIG A0



   int decimal = 2;               // Decimal places of the sensor value outputs
   int sensor0;                /* Assign the name "sensor0" as analog output value from
Channel C0 */
   int sensor1;                /* Assign the name "sensor1" as analog output value from
Channel C1 */
   int sensor2;                /* Assign the name "sensor2" as analog output value from
Channel C2 */
   int sensor3;                /* Assign the name "sensor3" as analog output value from
Channel C3 */
   int sensor4;                /* Assign the name "sensor4" as analog output value from
Channel C4 */
   int sensor5;                /* Assign the name "sensor5" as analog output value from
Channel C5 */
   int sensor6;                /* Assign the name "sensor6" as analog output value from
Channel C6 */
   int sensor7;                /* Assign the name "sensor7" as analog output value from
Channel C7 */
   int sensor8;                /* Assign the name "sensor8" as analog output value from
Channel C8 */
   int sensor9;                /* Assign the name "sensor9" as analog output value from
Channel C9 */
   int sensor10;               /* Assign the name "sensor10" as analog output value from
Channel C10 */
   int sensor11;               /* Assign the name "sensor11" as analog output value from
Channel C11 */
   int sensor12;               /* Assign the name "sensor12" as analog output value from
Channel C12 */
   int sensor13;               /* Assign the name "sensor13" as analog output value from
Channel C13 */
   int sensor14;               /* Assign the name "sensor14" as analog output value from
Channel C14 */
   int sensor15;               /* Assign the name "sensor15" as analog output value from
Channel C15 */

int prev5;
int prev6;
int prev7;
int prev8;
int prev9;

int triggerPin = 7;    // Trigger
int echoPin = 6;    // Echo
long duration;
int distance;
```

```
int triggerPin2 = 9;    // Trigger
int echoPin2 = 8;    // Echo
long duration2;
int distance2;



int prev1;
boolean buttonsave1 = false;
const int buttonPin1 = 1;
int buttonState1 = 0;


int prev2;
boolean buttonsave2 = false;
const int buttonPin2 = 2;
int buttonState2 = 0;


int prev3;
boolean buttonsave3 = false;
const int buttonPin3 = 5;
int buttonState3 = 0;

int prev4;
boolean buttonsave4 = false;
const int buttonPin4 = 3;
int buttonState4 = 0;

void setup() {

Serial.begin(9600);

  pinMode(S0,OUTPUT);                /* Define digital signal pin as output to the Multi-
plexer pin SO */
    pinMode(S1,OUTPUT);                /* Define digital signal pin as output to the Multi-
plexer pin S1 */
    pinMode(S2,OUTPUT);                /* Define digital signal pin as output to the Multi-
plexer pin S2 */
    pinMode(S3,OUTPUT);                /* Define digital signal pin as output to the Multi-
plexer pin S3 */
    pinMode(SIG, INPUT);               /* Define analog signal pin as input or receiver
from the Multiplexer pin SIG */

  pinMode(buttonPin1, INPUT);
    pinMode(buttonPin2, INPUT);
    pinMode(buttonPin3, INPUT);
    pinMode(buttonPin4, INPUT);

  pinMode(triggerPin, OUTPUT);
    pinMode(echoPin, INPUT);
    pinMode(triggerPin2, OUTPUT);
    pinMode(echoPin2, INPUT);

}

void controlChange(byte channel, byte control, byte value) {
  MIDIEventPacket_t event = {0x0B, 0xB0 | channel, control, value};
  MIDIUSB.sendMIDI(event);
}

void loop() {

    // Channel 0 (C0 pin - binary output 0,0,0,0)
    digitalWrite(S0,LOW); digitalWrite(S1,LOW); digitalWrite(S2,LOW);
digitalWrite(S3,LOW);
    sensor0 = analogRead(SIG);

    // Channel 1 (C1 pin - binary output 1,0,0,0)
    digitalWrite(S0,HIGH); digitalWrite(S1,LOW); digitalWrite(S2,LOW);
digitalWrite(S3,LOW);
    sensor1 = analogRead(SIG);

    // Channel 2 (C2 pin - binary output 0,1,0,0)
    digitalWrite(S0,LOW); digitalWrite(S1,HIGH); digitalWrite(S2,LOW);
digitalWrite(S3,LOW);
    sensor2 = analogRead(SIG);

    // Channel 3 (C3 pin - binary output 1,1,0,0)
    digitalWrite(S0,HIGH); digitalWrite(S1,HIGH); digitalWrite(S2,LOW);
digitalWrite(S3,LOW);
    sensor3 = analogRead(SIG);

    // Channel 4 (C4 pin - binary output 0,0,1,0)
    digitalWrite(S0,LOW); digitalWrite(S1,LOW); digitalWrite(S2,HIGH);
digitalWrite(S3,LOW);
    sensor4 = analogRead(SIG);

    // Channel 5 (C5 pin - binary output 1,0,1,0)
    digitalWrite(S0,HIGH); digitalWrite(S1,LOW); digitalWrite(S2,HIGH);
digitalWrite(S3,LOW);
    sensor5 = analogRead(SIG);

    // Channel 6 (C6 pin - binary output 0,1,1,0)
    digitalWrite(S0,LOW); digitalWrite(S1,HIGH); digitalWrite(S2,HIGH);
digitalWrite(S3,LOW);
    sensor6 = analogRead(SIG);

    // Channel 7 (C7 pin - binary output 1,1,1,0)
    digitalWrite(S0,HIGH); digitalWrite(S1,HIGH); digitalWrite(S2,HIGH);
digitalWrite(S3,LOW);
    sensor7 = analogRead(SIG);

    // Channel 8 (C8 pin - binary output 0,0,0,1)
    digitalWrite(S0,LOW); digitalWrite(S1,LOW); digitalWrite(S2,LOW);
digitalWrite(S3,HIGH);
    sensor8 = analogRead(SIG);

    // Channel 9 (C9 pin - binary output 1,0,0,1)
    digitalWrite(S0,HIGH); digitalWrite(S1,LOW); digitalWrite(S2,LOW);
digitalWrite(S3,HIGH);
    sensor9 = analogRead(SIG);

    // Channel 10 (C10 pin - binary output 0,1,0,1)
    digitalWrite(S0,LOW); digitalWrite(S1,HIGH); digitalWrite(S2,LOW);
digitalWrite(S3,HIGH);
    sensor10 = analogRead(SIG);

    // Channel 11 (C11 pin - binary output 1,1,0,1)
    digitalWrite(S0,HIGH); digitalWrite(S1,HIGH); digitalWrite(S2,LOW);
digitalWrite(S3,HIGH);
    sensor11 = analogRead(SIG);

    // Channel 12 (C12 pin - binary output 0,0,1,1)
    digitalWrite(S0,LOW); digitalWrite(S1,LOW); digitalWrite(S2,HIGH);
digitalWrite(S3,HIGH);
    sensor12 = analogRead(SIG);

    // Channel 13 (C13 pin - binary output 1,0,1,1)
    digitalWrite(S0,HIGH); digitalWrite(S1,LOW); digitalWrite(S2,HIGH);
digitalWrite(S3,HIGH);
    sensor13 = analogRead(SIG);

    // Channel 14 (C14 pin - binary output 0,1,1,1)
    digitalWrite(S0,LOW); digitalWrite(S1,HIGH); digitalWrite(S2,HIGH);
digitalWrite(S3,HIGH);
    sensor14 = analogRead(SIG);

    // Channel 15 (C15 pin - binary output 1,1,1,1)
    digitalWrite(S0,HIGH); digitalWrite(S1,HIGH); digitalWrite(S2,HIGH);
digitalWrite(S3,HIGH);
    sensor15 = analogRead(SIG);

    Serial.print("Sensor 0 : ");Serial.print(sensor0);          /* state value for sensor 0 */
    Serial.print(" . Sensor 5 : ");Serial.print(sensor5);
    Serial.print(" . Sensor 2 : ");Serial.print(sensor2);
    Serial.print(" . Sensor 3 : ");Serial.print(sensor3);
    Serial.print(" . Sensor 4 : ");Serial.println(sensor4);/* state value for sensor 1 */
          /* state value for sensor 4 */
         /* state value for sensor 15 */

    //delay(1000);                                              // Read the value every second



  int sensorValue1 = (sensor0);
  int sensorValue2 = (sensor5);
  int sensorValue3 = (sensor2);
  int sensorValue4 = (sensor3);
  int sensorValue5 = (sensor4);
int MIDIValue5 = map(sensorValue1, 0, 1024, 127, 0);

  if (MIDIValue5 != prev5){
 controlChange(3, 6, MIDIValue5);   // Channel 0, middle C, normal velocity
  MIDIUSB.flush();

  }

  prev5 = MIDIValue5;


int MIDIValue6 = map(sensorValue2, 0, 1024, 127, 0);

  if (MIDIValue6 != prev6){
 controlChange(3, 7, MIDIValue6);   // Channel 0, middle C, normal velocity
  MIDIUSB.flush();

  }

  prev6 = MIDIValue6;

  int MIDIValue7 = map(sensorValue3, 0, 1024, 127, 0);

  if (MIDIValue7 != prev7){
 controlChange(3, 8, MIDIValue7);   // Channel 0, middle C, normal velocity
  MIDIUSB.flush();

  }

  prev7 = MIDIValue7;

  int MIDIValue8 = map(sensorValue4, 0, 1024, 127, 0);

  if (MIDIValue8 != prev8){
 controlChange(3, 9, MIDIValue8);   // Channel 0, middle C, normal velocity
  MIDIUSB.flush();

  }

  prev8 = MIDIValue8;

   // Channel 0, middle C, normal velocity
  MIDIUSB.flush();

buttonState1 = digitalRead(buttonPin1);
  buttonState2 = digitalRead(buttonPin2);
   buttonState3 = digitalRead(buttonPin3);
    buttonState4 = digitalRead(buttonPin4);

  // check if the pushbutton is pressed. If it is, the buttonState is HIGH:



int MIDIValue1 = map(buttonState1, 1, 0, 0, 127);
  if (MIDIValue1 != prev1){
    if (MIDIValue1 == 0){
      buttonsave1 = !buttonsave1;
      if (buttonsave1) {
         controlChange(3, 10, 127);   // Channel 0, middle C, normal velocity
  MIDIUSB.flush();
      }
      else {
         controlChange(3, 10, 0);   // Channel 0, middle C, normal velocity
  MIDIUSB.flush();

      }
    }

}

prev1 = MIDIValue1;

int MIDIValue2 = map(buttonState2, 1, 0, 0, 127);
  if (MIDIValue2 != prev2){
    if (MIDIValue2 == 0){
      buttonsave2 = !buttonsave2;
      if (buttonsave2) {
         controlChange(3, 12, 127);   // Channel 0, middle C, normal velocity
  MIDIUSB.flush();

      }
      else {
         controlChange(3, 12, 0);   // Channel 0, middle C, normal velocity
  MIDIUSB.flush();


      }
    }

}

prev2 = MIDIValue2;


int MIDIValue3 = map(buttonState3, 1, 0, 0, 127);
  if (MIDIValue3 != prev3){
    if (MIDIValue3 == 0){
      buttonsave3 = !buttonsave3;
      if (buttonsave3) {
         controlChange(3, 13, 127);   // Channel 0, middle C, normal velocity
  MIDIUSB.flush();
      }
      else {
         controlChange(3, 13, 0);   // Channel 0, middle C, normal velocity
  MIDIUSB.flush();

      }
    }
}

prev3 = MIDIValue3;

int MIDIValue4 = map(buttonState4, 1, 0, 0, 127);
  if (MIDIValue4 != prev4){
    if (MIDIValue4 == 0){
      buttonsave4 = !buttonsave4;
      if (buttonsave4) {
         controlChange(3, 1, 127);   // Channel 0, middle C, normal velocity
  MIDIUSB.flush();

      }
      else {
         controlChange(3, 1, 0);   // Channel 0, middle C, normal velocity
  MIDIUSB.flush();



      }
    }
  }

        prev4 = MIDIValue4;

  int sensorValue6 = analogRead(A4);
Serial.print(sensorValue6);
  delay(1);



  digitalWrite(triggerPin, LOW);
  delayMicroseconds(5);
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);

  pinMode(echoPin, INPUT);
  duration = pulseIn(echoPin, HIGH);

  // Convert the time into a distance
  distance = duration * 0.0343 / 2;



int MIDIValue10 = map(distance, 0, 13, 0, 127);

  delay(10);

Serial.print(" ");
Serial.println(MIDIValue10);

  if (sensorValue6 < 200){
controlChange(3, 4, MIDIValue10);   // Channel 0, middle C, normal velocity
  MIDIUSB.flush();

     }
int sensorValue7 = analogRead(A5);
Serial.print(sensorValue7);
  delay(1);



  digitalWrite(triggerPin2, LOW);
  delayMicroseconds(5);
  digitalWrite(triggerPin2, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin2, LOW);

  pinMode(echoPin2, INPUT);
  duration2 = pulseIn(echoPin2, HIGH);

  // Convert the time into a distance
  distance2 = duration2 * 0.0343 / 2;



int MIDIValue11 = map(distance2, 0, 13, 0, 127);

Serial.print(" ");
Serial.println(MIDIValue11);

  delay(10);

  if (sensorValue7 < 200){
controlChange(3, 5, MIDIValue11);   // Channel 0, middle C, normal velocity
  MIDIUSB.flush();

  }
}

void noteOn(byte channel, byte pitch, byte velocity) {
  MIDIEventPacket_t noteOn = {0x09, 0x90 | channel, pitch, velocity};
  MIDIUSB.sendMIDI(noteOn);
}


void noteOff(byte channel, byte pitch, byte velocity) {
  MIDIEventPacket_t noteOff = {0x08, 0x80 | channel, pitch, velocity};
  MIDIUSB.sendMIDI(noteOff);
}
```

Figure 11. Arduino Code for the Ring of Fortune

```
#include "MIDIUSB.h"


int triggerPin2 = 9;    // Trigger
int echoPin2 = 8;    // Echo
long duration2;
int distance2;

void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);

  pinMode(triggerPin2, OUTPUT);
  pinMode(echoPin2, INPUT);



}

void controlChange(byte channel, byte control, byte value) {
  MIDIEventPacket_t event = {0x0B, 0xB0 | channel, control, value};
  MIDIUSB.sendMIDI(event);
}


void loop() {
  // read the input on analog pin 0:


int sensorValue7 = analogRead(A0);

  delay(1);



  digitalWrite(triggerPin2, LOW);
  delayMicroseconds(5);
  digitalWrite(triggerPin2, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin2, LOW);

  pinMode(echoPin2, INPUT);
  duration2 = pulseIn(echoPin2, HIGH);

  // Convert the time into a distance
  distance2 = duration2 * 0.0343 / 2;
```

```
int MIDIValue11 = map(distance2, 0, 14, 0, 127);



  delay(10);

  if (sensorValue7 < 300){
controlChange(3, 15, MIDIValue11);   // Channel 0, middle C, normal velocity
  MIDIUSB.flush();



  }




int sensorValue8 = analogRead(A1);




  if (sensorValue8 < 300){
controlChange(3, 16, MIDIValue11);   // Channel 0, middle C, normal velocity
  MIDIUSB.flush();


}

int sensorValue9 = analogRead(A2);

  if (sensorValue9 < 300){
controlChange(3, 17, MIDIValue11);   // Channel 0, middle C, normal velocity
  MIDIUSB.flush();



}


}
void noteOn(byte channel, byte pitch, byte velocity) {
  MIDIEventPacket_t noteOn = {0x09, 0x90 | channel, pitch, velocity};
  MIDIUSB.sendMIDI(noteOn);
}

void noteOff(byte channel, byte pitch, byte velocity) {
  MIDIEventPacket_t noteOff = {0x08, 0x80 | channel, pitch, velocity};
  MIDIUSB.sendMIDI(noteOff);
}
```

Figure 12. Arduino Code for CV control

```
int vcpi = 11;



void setup() {
  // put your setup code here, to run once:


pinMode (vcpi,OUTPUT);
pinMode (A0, INPUT);
Serial.begin(9600);


}

void loop() {
  // put your main code here, to run repeatedly:
 int sensorValue = analogRead(A0);
 int VC = map(sensorValue, 0, 1024, 1, 54);
Serial.println(VC);
analogWrite(vcpi,VC);
  delay(1);


}
```

Figure 13. Arduino Code for LFO control

```
int frequency = 2; //Set frequency in Hertz
// frequency * 2


void setup()


{

pinMode(11, OUTPUT);
pinMode(A0, INPUT);


}


void loop()
{
  double sensorValue = analogRead(A0);
  double sensorMap1 = map(sensorValue, 0, 1024, 100, 3000);
  double delayTime = sensorMap1;

 digitalWrite(11, HIGH);

 delay(100);



 digitalWrite(11, LOW);
 delay(delayTime);




}
```

Figure 14. Using the Ring of Fortune during a performance at Radio City, Valencia — Back

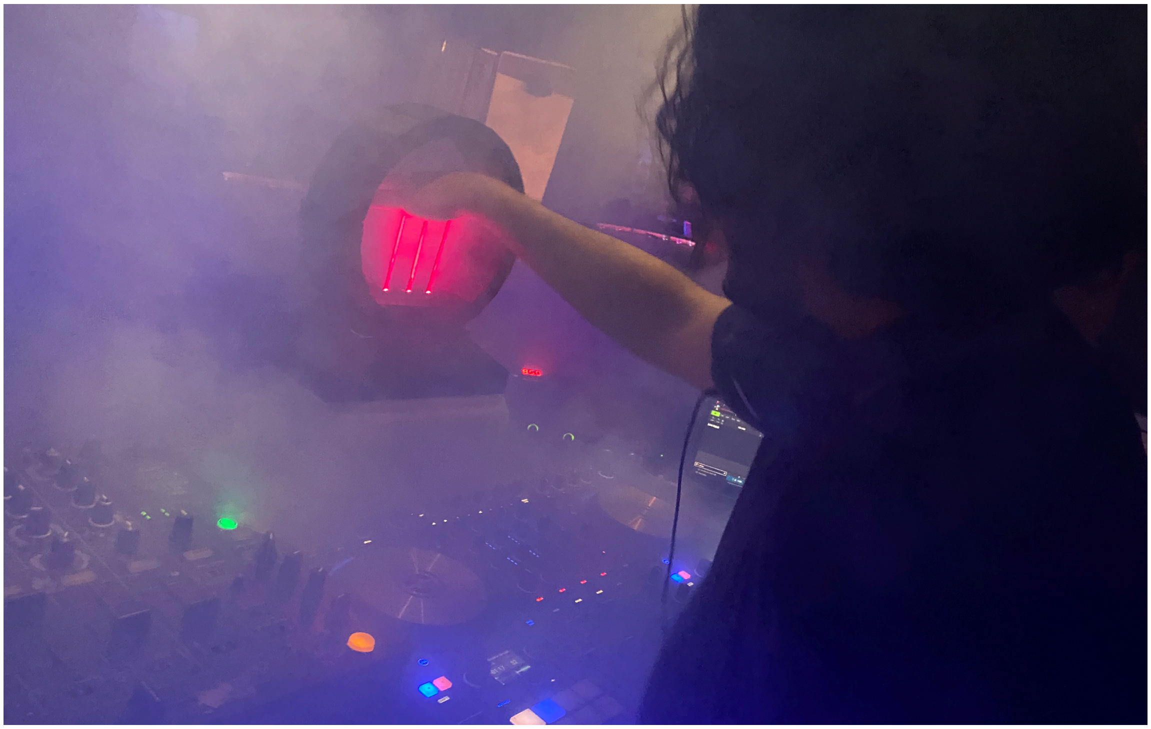


Figure 15. Using the Ring of Fortune during a performance at Radio City, Valencia — Front

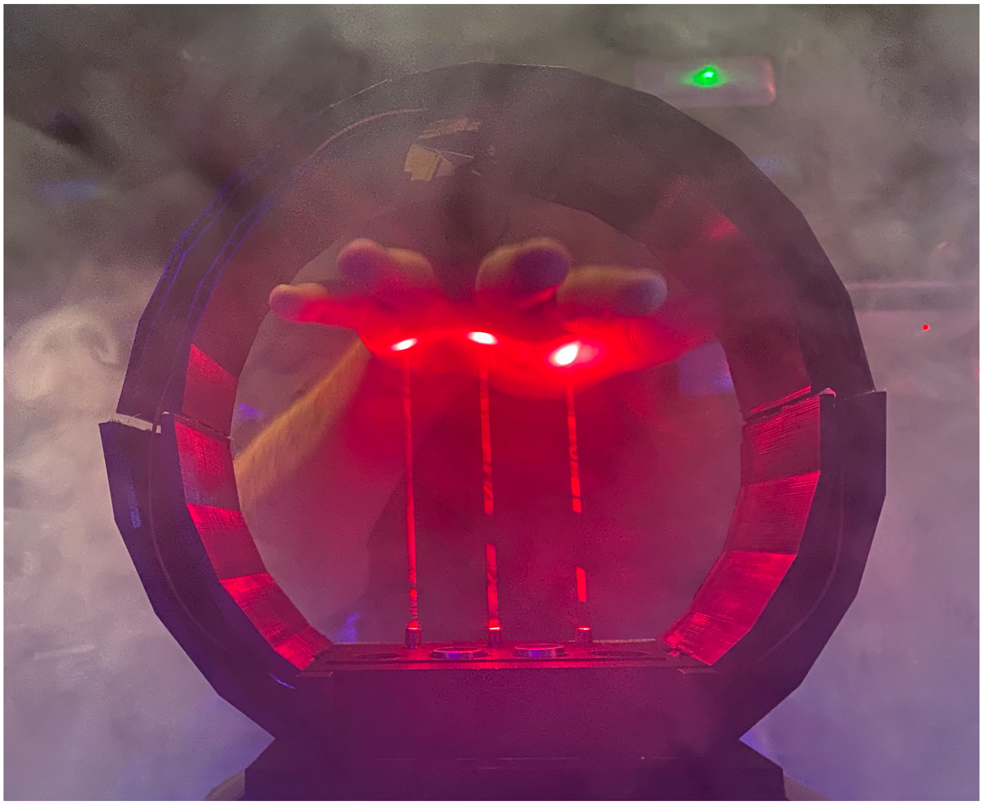


Figure 16. Complete image of the Mystery Box

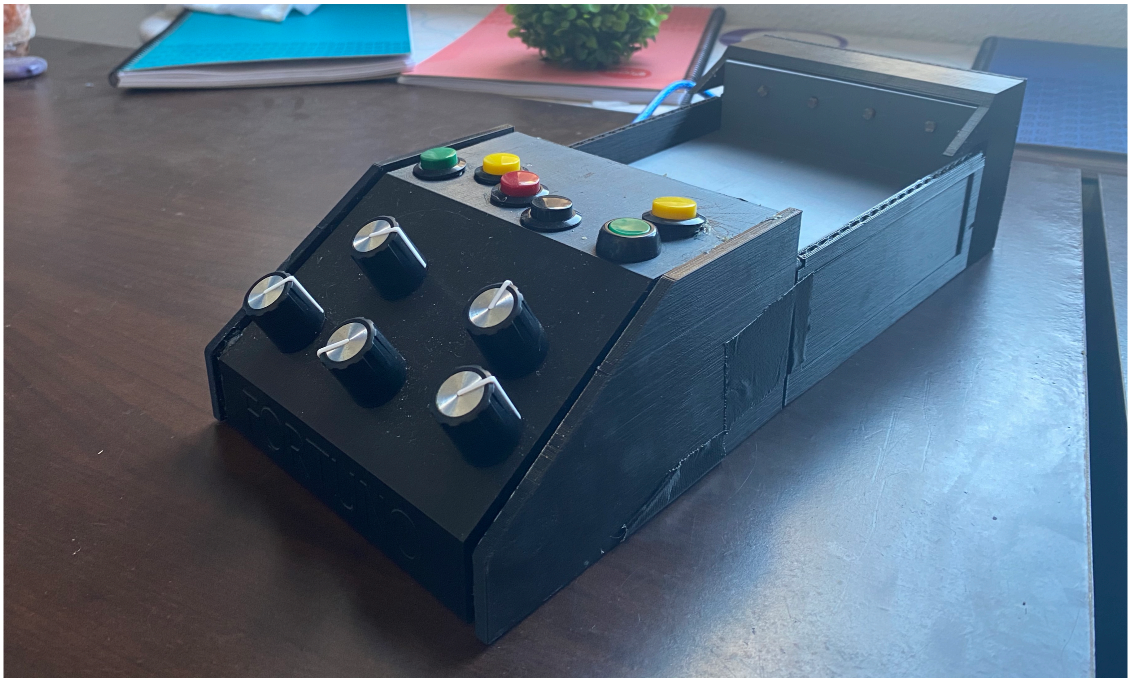
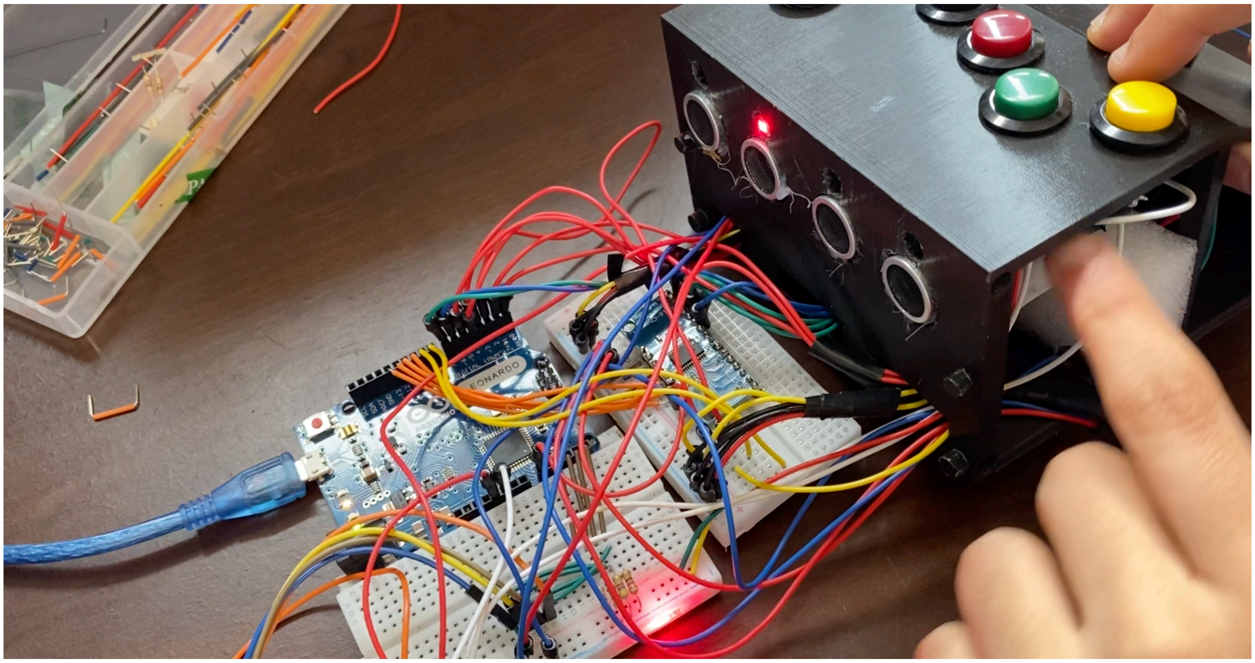


Figure 17. The inside of the Mystery Box



Bibliography

Ben Bohmer: Anjunadeep. “Ben Böhmer - Breathing.” Anjunadeep, 2019. https://anjunadeep.com/ben-bohmer/.

Brechja , Boris. Boris Brejcha, 2018. https://www.borisbrejcha.de/.

Deahl, Dani. “Eric Prydz Is Going to DJ inside a Giant Glowing Sphere - Here's How It Was Made.” The Verge. The Verge, July 9, 2019. https://www.theverge.com/2019/7/9/20683461/dj-eric-prydz-epic-6-0-holosphere-tomor-rowland.

Gomez, Juan. “Sound and Technology.” Portfolio, 2020. https://totumag.wixsite.com/portfolio/copy-of-circuit-design

Math Make Noise: Noise, Make. “MATHS.” Make Noise Co., 2012. https://www.makenoisemusic.com/modules/maths.

Noise, Make. “Pressure Points.” Make Noise Co., 2012. https://www.makenoisemusic.com/modules/pressure-points.

Resident Advisor. “Stephan Bodzin.” Resident Advisor, June 18, 2021. https://www.residentadvisor.net/DJ/stephanbodzin/biography.

VA. “Biography.” JeanMichel Jarre, 2020. https://jeanmicheljarre.com/biography.

V.A. “Stephan Bodzin's PO10 LiveController.” Stephan Bodzin's PO10 LiveController | Equipboard®. Accessed December 20, 2020. https://equipboard.com/pros/stephan-bodzin/po10-livecontroller.