Concordia University, Computation Arts

ASSIGNMENT THIS

CART 360, Fall 2018

Students: Roberto Gutierrez & Mélina Lopez-Racine

GitHub repository:

https://github.com/mlopezracine/CART360/tree/master/ASSIGNMENTS/ASSIGNMENT_THIS

Table of Content

Part One: Sensor Research and Prelim Work

1. SENS-37 TCS230/TCS3200 COLOUR RECOGNITION SENSOR

   Adapted to Arduino, the sensor communicates directly with the microcontroller and is connected through pins (both digital and analog outputs). Each pin of the sensor controls their component needed to detect the proper colour; The pins S0 and S1 control the frequency, S2 and S3 control the RGB filter, while OE is the analog output. The sensor needs about 2.7 V to 5.5 V of supply voltage, and have an absolute maximum of 6 V. Over the absolute would cause permanent damage to the sensor. The calibration is essential to detect colors properly; Although an external library is not required to program the sensor, one may find it easier to calibrate the device with. It is important to note the colour recognition sensor is not precise, as it has many factors (light and distance) that might influence the result. The closer the object is to the sensor, the better the result is.

   All information on the sensor was found on its Date Sheet (https://www.pobot.org/IMG/pdf/tcs230_datasheet.pdf), and on Abra Electronics's website (https://abra-electronics.com/sensors/sensors-light-imaging-en/sens-37-tcs230-tcs3200-color-recognition-sensor-board-for-arduino.html). Additional information was found on an online tutorial created by Dejan, "Arduino Color Sensing Tutorial – TCS230/ TCS3200 Color Sensor", posted on How To Mechatronics in May 2016. The tutorial included a schematic of the sensor connected to the Arduino, as well as source codes.

2. RANGING HC-SR04 ULTRASONIC SENSOR

   This sensor uses sonar frequency, a high-frequency sound of 40k Hz, to calculate the distance between an object and itself, to an extend of 2 cm to 500 cm (the resolution is 1 cm). The object reflects the signal, which is then received by sensor's transmitter, which will calculate the time between transmission to determine the distance. Therefore, the sensor does not require light, or is affected by sunlight, to function. Adapted to Arduino, the sensor may be connected directly to the devise, and needs 5V of supply voltage to function properly. The sensor is composed of four pins, VCC, Trig (Trigger, which is the analog input), Echo (Echo, which is the analog output), and GND.

   All information on the sensor was found on Abra Electronics's website (https://abra-electronics.com/sensors/sensors-proximity-en/hc-sr04-ultrasonic-sensor-ranging-modulehc-sr04.html). Additional information was found on an online tutorial created by Dejan, "Ultrasonic Sensor HC-SR04 and Arduino Tutorial", posted on How To Mechatronics in July 2015. The tutorial included a schematic of the sensor connected to the Arduino, as well as source codes.

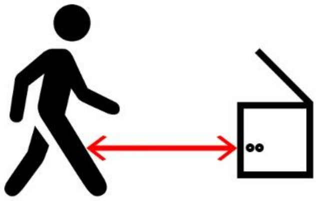
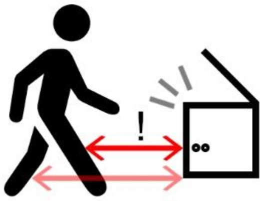
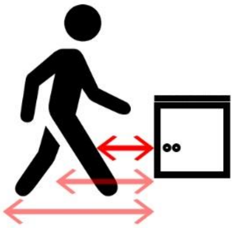
3. TMP37FT9Z TEMPERATURE SENSOR

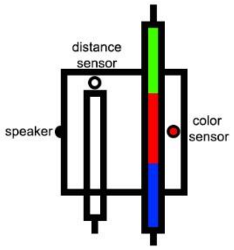
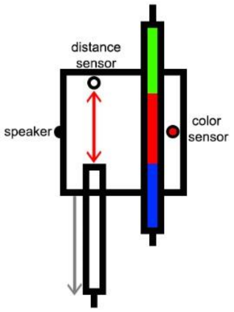
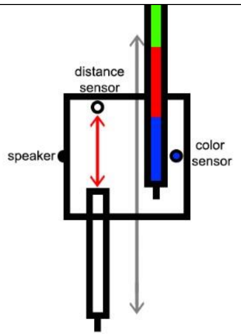
   This sensor give voltage proportionally linear to Celsius (centigrade) temperature. The accuracy is ±2 ºC, have a linearity of ±0.5 ºC, and provides a nominal value of 500mV output at 25 ºC. The operating temperature is specified to between -40 ºC to +125ºC and is operational to +150 ºC. Although, it is intended to be working applications over the range of 5 ºC to 100 ºC. The analog is compatible with Arduino, it requires a supply voltage of 2.7 V to 5.5 V to function properly, and a charging current of 0.05 mA. The sensor is composed of three pins, a voltage input, an analog voltage out, and a GND. At an absolute maximal capacity, the sensor will peak its temperature in 20 seconds to 40 seconds. Depending on the circuit and needs, it would require resistors to limit its input, as if the sensor function over above the absolute maximum ratings, it may cause permanent damage to the sensor as well as to the Arduino.

   All information on the sensor was found on its Data Sheet, and on Abra Electronic's website (https://abra-electronics.com/sensors/sensors-temperature-en/tmp37ft9z-temperature-sensor.html).

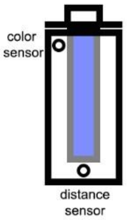
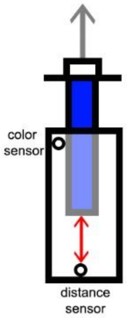
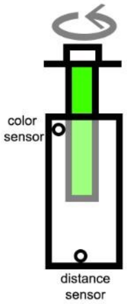
We ended choosing the RANGING HC-SR04 ULTRASONIC SENSOR (referred as the "distance sensor" in our document), as we thought it would be interesting to force the user to use their body to activate the sensor. Through different iterations, we decided to add the SENS-37 TCS230/TCS3200 COLOUR RECOGNITION SENSOR (referred as the "colour sensor" in our document) to our concept to obtain more complex interactions, thus obtaining a more meaningful experience.

Storyboard:

STORYBOARD: Iteration 1		
Scene 1	Scene 2	Scene 3
		
The two dotes represent the distance sensor. It calculates the distance between the box and the user	When the user gets too close to the box, a speaker (not pictured) activates and makes noise.	The user gets even closer to the box. It triggers a servo sensor (not pictured) and close the lid.

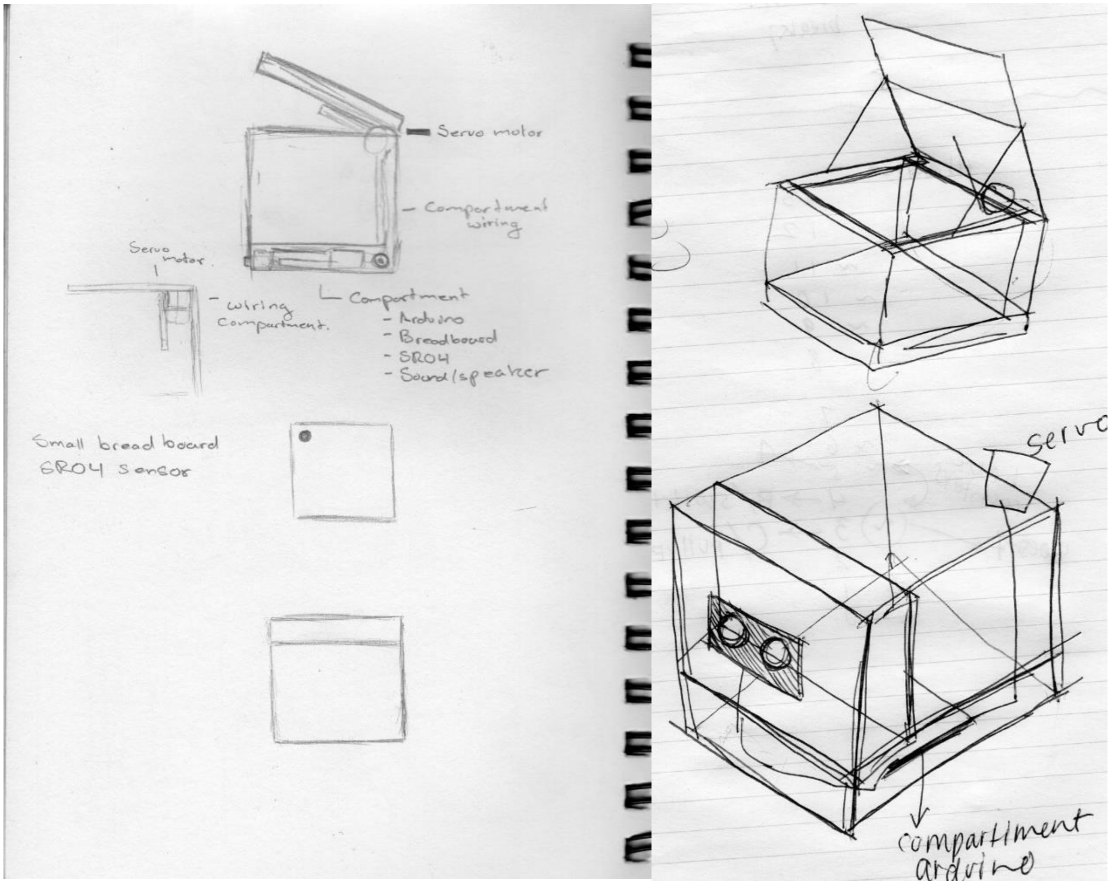
STORYBOARD: Iteration 2		
Scene 1	Scene 2	Scene 3
		
In a container, two handlers are place in front of their respective sensors, a distance sensor and a colour sensor.	The distance sensor calculates the distance between the handler and itself. It affects the speaker, and alters the noise.	The colour sensor detect the change of colour from its handler. It affects the speaker, and alters the pitch.

STORYBOARD: Iteration 3		
Scene 1	Scene 2	Scene 3

		
In a container, a unique handler is place on top of a distance sensor. A colour sensor is placed on the side. A speaker (not pictured) is inside the box.	When the user moves the handler up and down, the distance sensor calculates the difference between the handler and itself, altering the noise of the speaker (not pictured).	The handler have different colours on each side of its surface. When the user rotates the handler, the colour sensor detects the change of colour, and alter the pitch of the speaker.

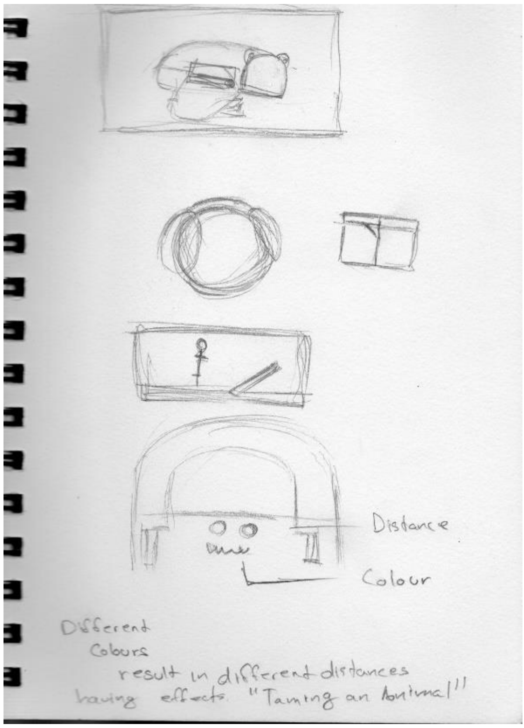
Part Two: Ideation and Progress

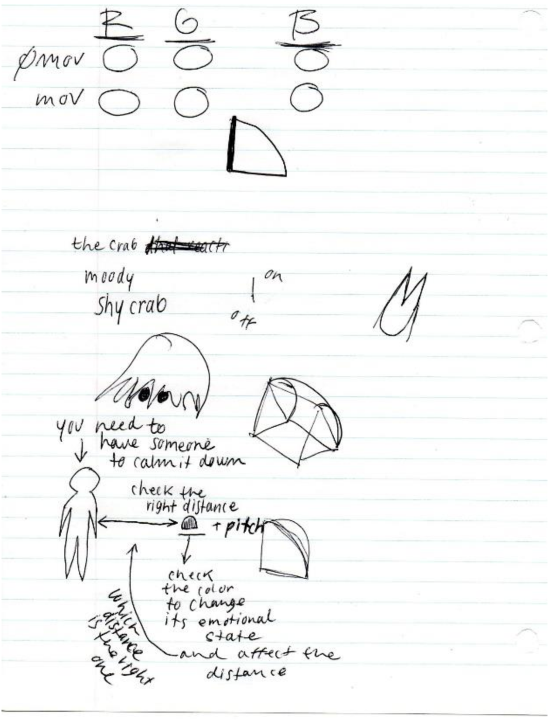
The project went through many changes over the course of our brainstorming, and even mid development. It had been a while since any of us had done a lot of programming, so it was a great joy, but also obstacle, figuring out what we may be able to make while enjoying the process. Due to a bit of wariness in regard to our perceived skill level in the subject, we started off with a fairly simple idea, that really didn’t put what we thought would be fun to do at the forefront. Our original idea centered around a box that would make noise and close itself if a person were to get too close to it, mimicking a sort of scared or mischievous animal, or guard.



After presenting this idea though we were encouraged to try something more complex, as this was too simple. During recommendations as to what to do we were shown “Electronic Musical Instruments from the Physical World,” Though our focus didn’t immediately turn to making an instrument, the idea was at the back of our minds throughout.

We first decided to focus further on the concept of an animal or creature, as well as a recommendation of having certain conditions for things to happen in which people interact with the object. We were also encouraged to explore our focus on games from out of class for inspiration.

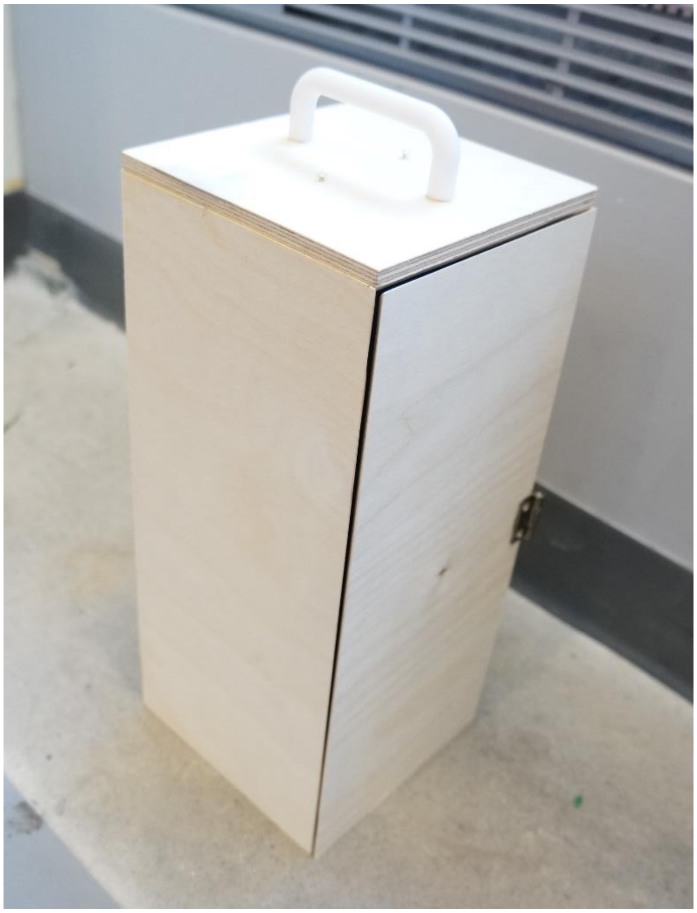

Distance
Colour
Different Colours result in different distances having effects. "Taming an Animal!!"


R G B
∅mov
mov
the crab ~~that watch~~
moody
on
off
Shy crab
you need to have someone to calm it down
check the right distance
+ pitch
check the color to change its emotional state and affect the distance
which distance is the right one

We explored the idea of, and settled on, using a colour sensor along with the distance one to simulate the concept of character space and condition relative to it. We thought it'd be interesting if the creature were to act differently depending on what distance a person stood from it, as well as where the creature was. The idea developed further into a game in which two people would need to cooperate: One person standing in front of the creature at a distance that keeps it calm, while the other moves it from one are to another, in hopes of calming it down for good, as opposed to temporarily. Each area would have a colour that would change how the creature would act according to the Person's distance. To better understand how the sensors worked, (and ultimately what the code would be for them), we mainly used howtomechatronics.com appropriating the code by Dejan Nedelkovski on both the Colour and Distance Sensors.

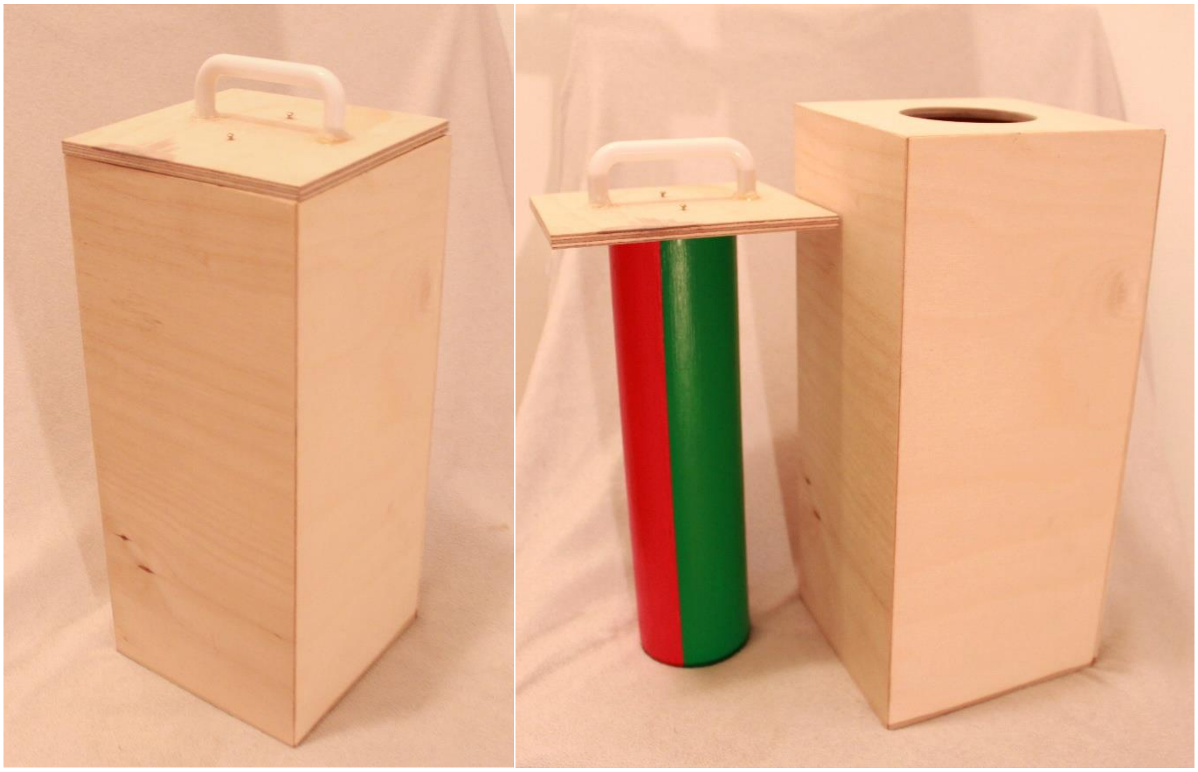
(Distance: https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/ and Colour: https://howtomechatronics.com/tutorials/arduino/arduino-color-sensing-tutorial-tcs230-tcs3200-color-sensor/)

During the coding phase, along with the code from howtomechatronics.com, we implemented some of the materials we learnt in class so far, along with techniques we slightly recalled from working on unity code and other game engines. Though we weren't sure if it would work exactly the same as in unity or as planned, we were happy when it seemed to produce results, and are eager to see how else we can combine our previous experiences with this one. This led to a further change in the project. When we tested we liked how the tone would switch depending on the distance and colour, and thought it was interesting to play around with it. First, we thought it may be interesting to have the notes correspond to people passing by, or dancing with the object, like a game. Eventually though we remembered the physical instruments from the physical world, and used that as inspiration and reference to create our own instrument that people could play around with to make music, the notes played being dependant on the colour of the pieces as well as their distance as they are manipulated by the users.

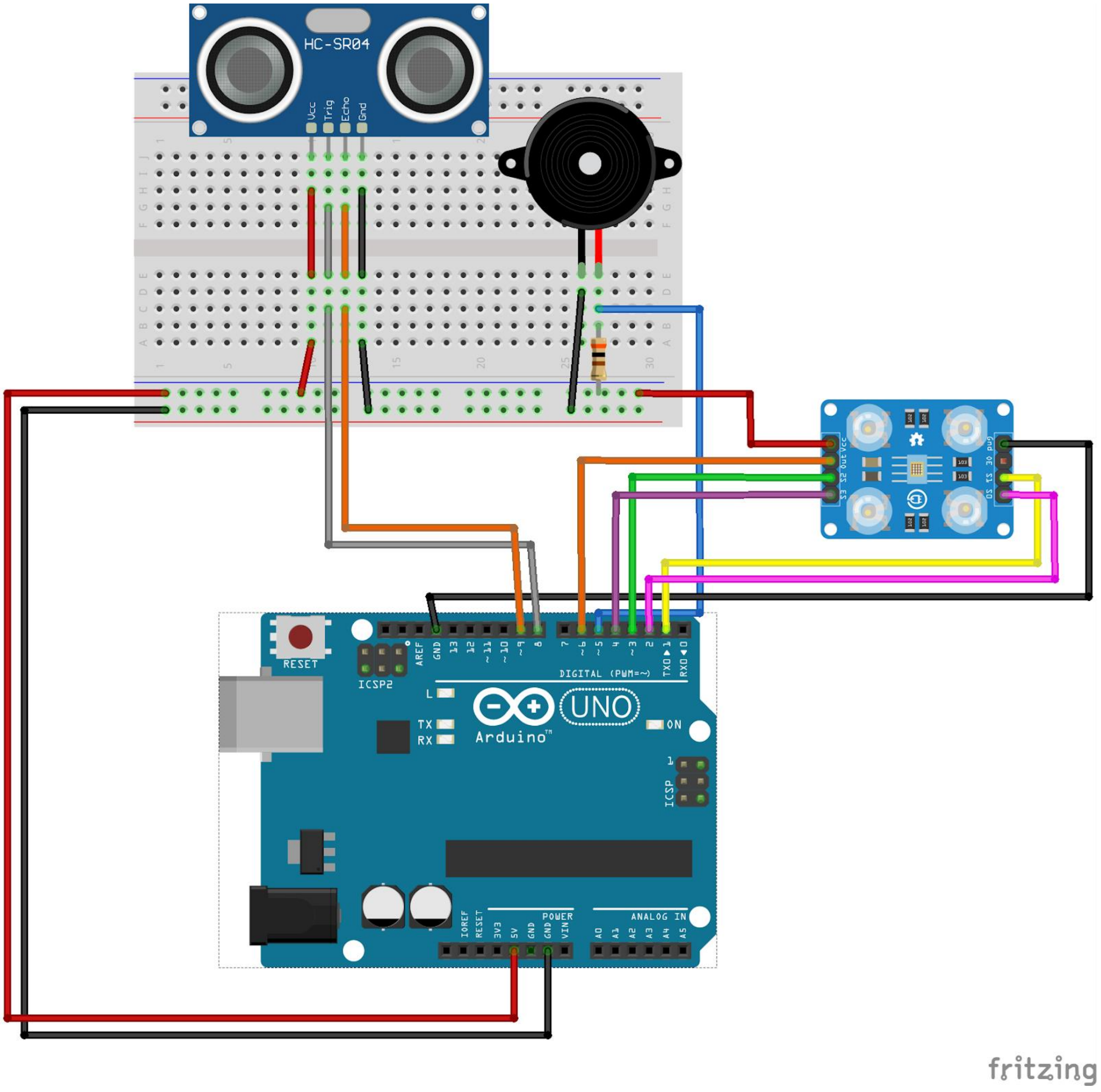


A for was finally settled on for the object, which harkened back to the original box idea, while also making use of the space to affect both sensors as needed to make the piece work. The protrusion from the box serves to affect both the distance and colour sensors, allowing for fun interaction and playfulness with the item, by all types of audiences.

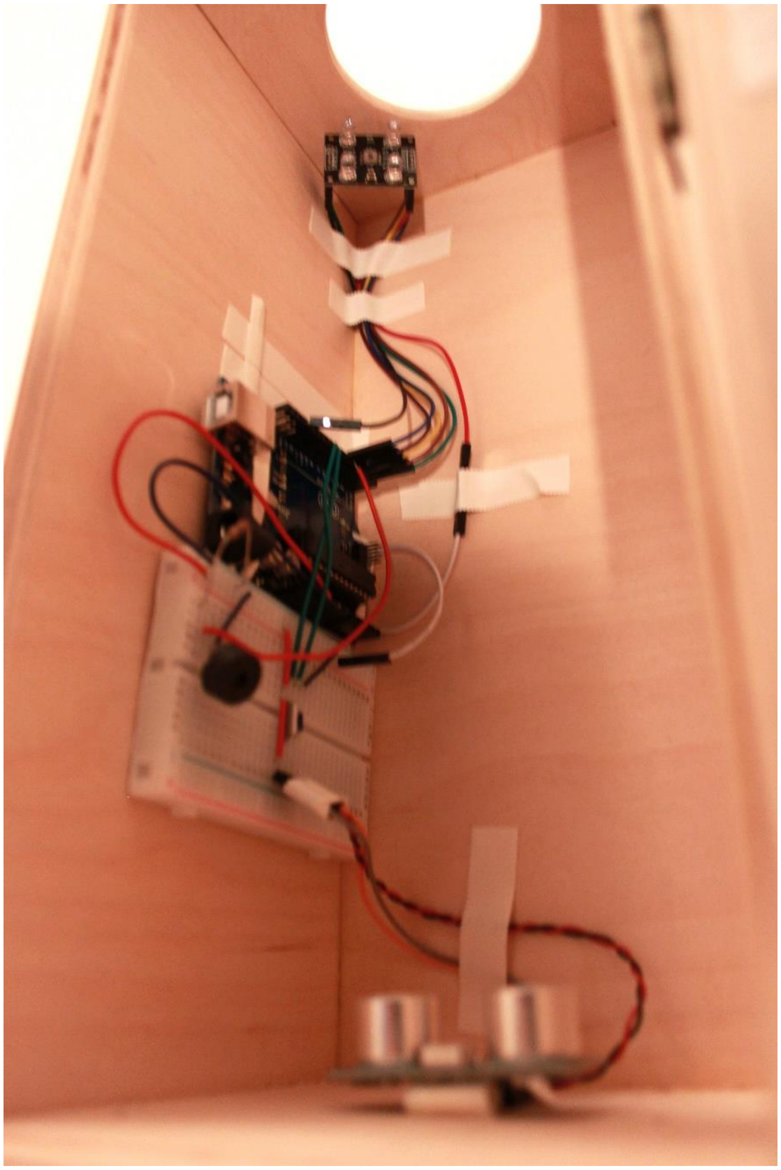
The way it was eventually built also takes into account human behaviour in order to illicit certain emotions and actions from the audience. For example: The implementation of a handle alludes top the already present thought in people that it is something that people would want to pull on, up and down or back and forth, and so it feeds into our use of a distance sensor for certain aspects. The circular form of the pillar also feeds into this as people like to turn things like knobs and handles naturally and so they are encouraged to play with it like that as well. The association of colour and sound was also an aspect we became interested in during the process. Certain colours are associated with different emotions in psychology, and similarly in music different tones and notes are associated with different emotions.

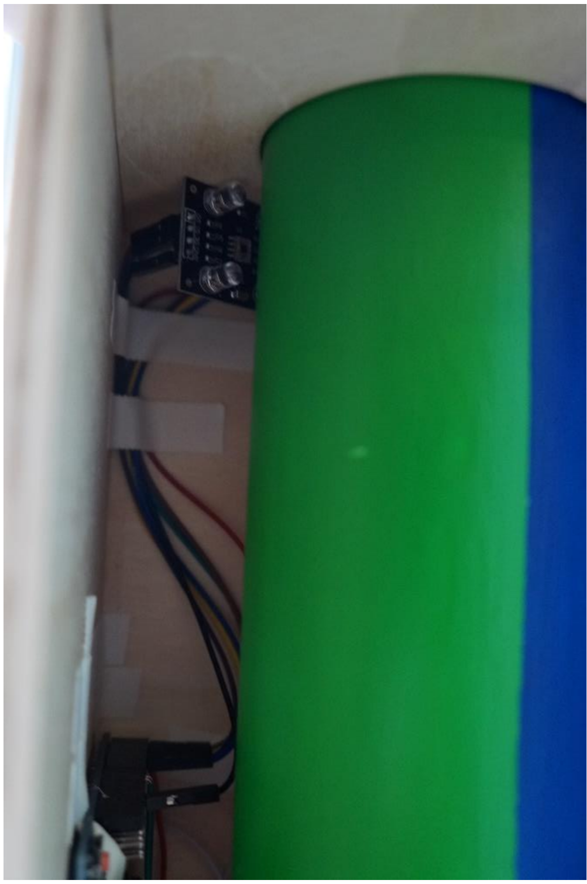


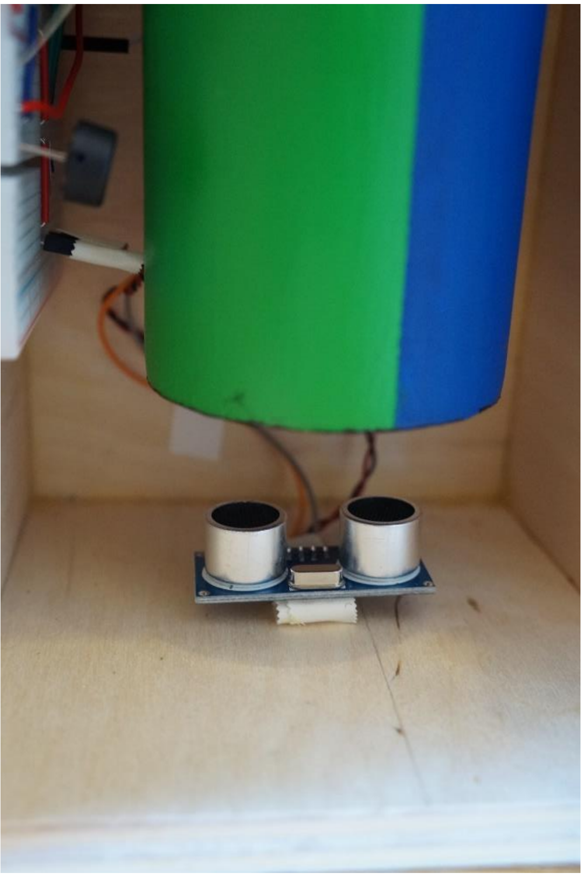
Schematic (Fritzing):



Circuit:







Bibliography

Nedelkovski, D. (2016, May 16). Arduino Color Sensing Tutorial - TCS230 TCS3200 Color Sensor. Retrieved October 10, 2018, from https://howtomechatronics.com/tutorials/arduino/arduino-color-sensing-tutorial-tcs230-tcs3200-color-sensor/

Nedelkovski, D. (2015, July 26). Ultrasonic Sensor HC-SR04 and Arduino Tutorial. Retrieved October 10, 2018, from https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/