2024 Gizmo Porfolio
Group 32
Helen Shi
Nico Cheong

Interactive board game

Our project is a recreation of a scene in *Stranger Things* in the form of an interactive board game

- The main structure of this board game is a board, dice and scenes. Players can use the board and dice to play any game while triggering Stranger Things-themed scene animations.
- This animation is a classic plot in the first season of Stranger Things. Players can watch this plot like a drama by constantly triggering mechanisms.
- The plot is about a mother using lights to communicate with her son in another parallel world.
- When the user is ready to start, choose a chess piece and place from the starting point.
- Use the digital dice to roll a random number and move that number of steps.
- When you land on the desired square, the next part of the scene will play.
- The goal is to finish all 14 level of the plots.

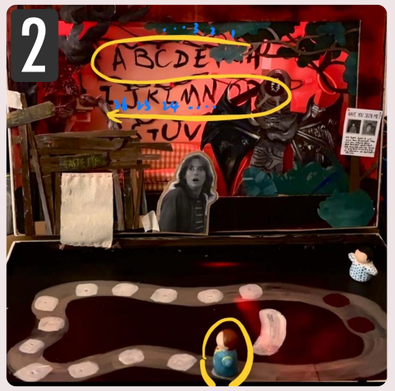
Resetting Point
Place the piece in this position at any time during the game and the game will start from the beginning.

Starting Spot
It belongs to level 0. When placed here, the initial animation will be played.

background Light
When the power is on and the switch is on, the backlight will stay on.

Interactive Demostration

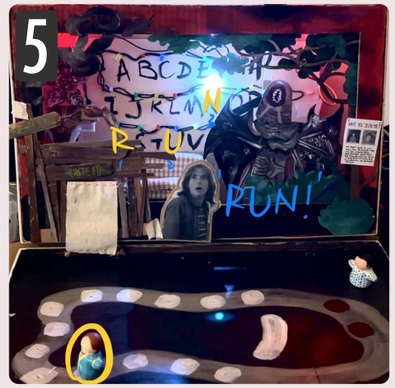
When the power is turned on and the switch is turned on, the two red LED lights of the background light will light up, giving feedback to the player: the game can start. At this time, the player can place the chess piece on the 'starting point', and the LEDs in the letter wall will light up in the order of: 'S-T-A-R-T', with the sound effect of 'ding', accompanied by the opening song.



After starting the game (level 0), the player (chess piece) enters the formal game area. When the chess piece falls on the grid on the chessboard, the first mechanism is triggered.
In the first level, the 26 LEDs on the letter wall light up in an S-shape from top to bottom, and flash repeatedly after reaching the vicinity of 'Castle Byers', and at the same time match the corresponding audio.

After the first level, the player will use the electronic dice provided with the game to randomly generate a number on the Led matrix screen. This number is the number of steps the player moves forward. When the player moves to the corresponding position, the mechanism of the second level will will be triggered.
In this level, the LED in ‘castle byers’ at the front of the scene lights up in conjunction with the audio.

The player moved forward through the numbers on the electronic dice and triggered the next level. "R-I-G-H-T-H-E-R-E" lights up in sequence and matches the sound effects in the audio.



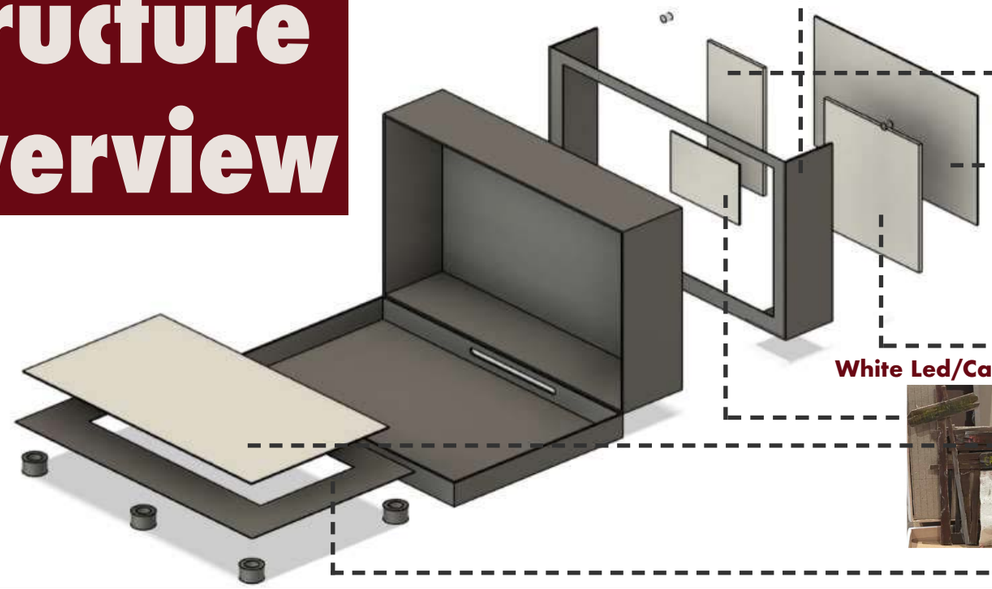
In the fifth level, the letters "R-U-N" light up in sequence and match the corresponding audio. (It implies that the protagonist escapes and the monster will appear in the next level)

In this level, all LED lights flash randomly, and the ‘bat’ flaps its wings and swings left and right.

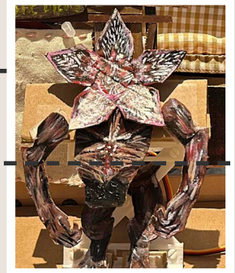
In this level, all LED lights flash randomly while the demogorgon moves up and down.

In the last level, the letter wall lights up in sequence "T-B-C", which means "to be continued" accompanied by the ending music

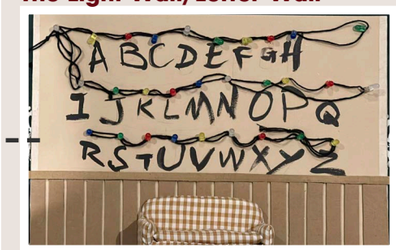
Structure Overview



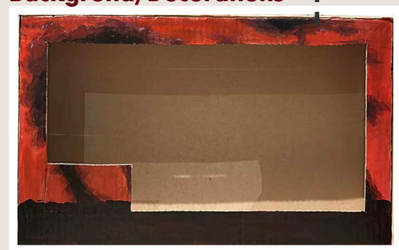
Demogorgon



The Light Wall/Letter Wall

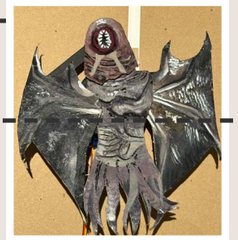


Backgrond/Decorations

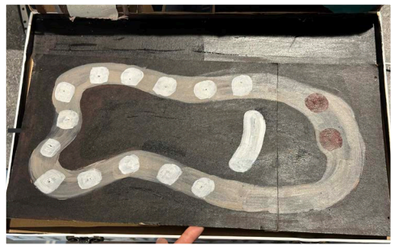


White Led/Castle Byers

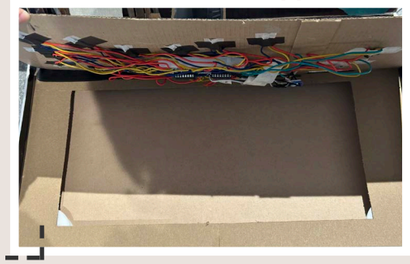
Demobat/Bat



Game Board/Grids



Board Support



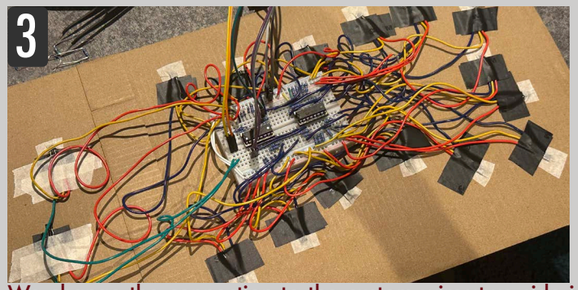
Production Process for Inputs

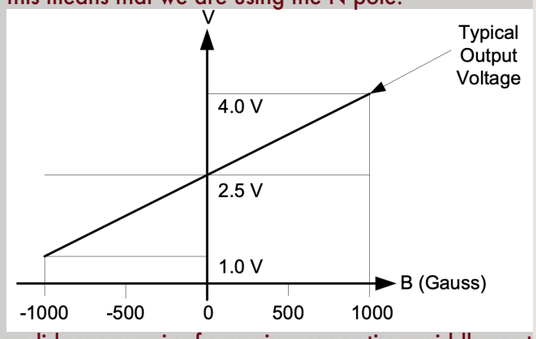
Game Board

At first we try to use Aluminium foil as conductor to cover all the GND and VCC pin of hall sensors which seems to be sustainable and convinient, but after we finished we find the contact between them is not very robust hence the impedance is always changing, which makes it very hard to get a stable value transfering to the parallel input of 74hc165 shift register.

The player pieces were moulded out of air-dry clay, with space at the bottom for magnets. The direction of the magnets should be justified.

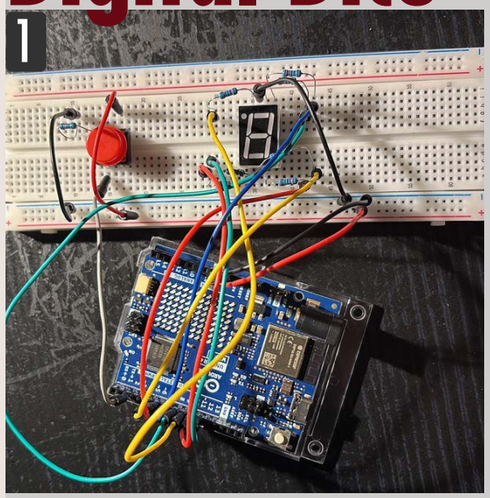
Our magnetic poles cause the voltage to increase from low to high (roughly from 1.8 V to 3.8 V), so the sensor contacts an area where the magnetic field strength changes from negative to positive values. According to the characteristics of the Hall sensor, this means that we are using the N pole.





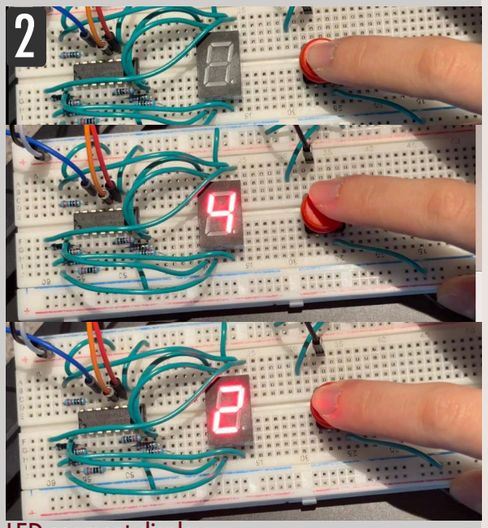
We change the connection to the custom wires:two side is solid copper wire for easier connection, middle part is standard copper wire, comfirming the flexiblity. After we soldering these wires to all GND and VCC wire for each hall sensor, we added a potential divider network to ensure register read logic LOW when there is no significant magnetic field. The shifter register is connected correctly according to the datasheet, and its clock pin and latch pin are shared with other part. Also, to make 2 stage cascade shift register, the closer register's serial input pin is connected to farther register' serial output. In this way there is only one serial output pin.

Digital Dice



When figuring out the circuit for the digital dice, tutorials on how to wire and program a 7-segment display, a button, and how to connect them both, were referenced and combined. In many occasions, troubleshooting was required: individual components had to be tested at the most basic level. For example, detecting whether the button was pressed.

However, even though it works, our arduino do not have enough space for 7 Pins, so we decided to use the same 74HC595 register for letter wall.



LED segment display:

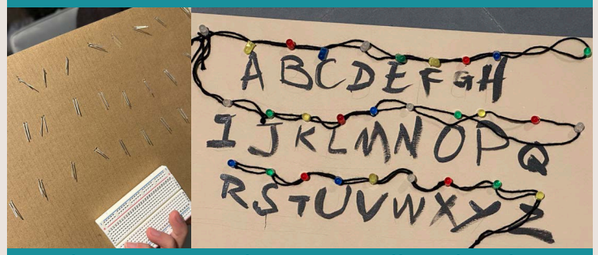
we use a 74HC595 to control 8 segments of LED. The code for number 1-6 is :

```
0b00000110,  // 1: b, c
0b01011011,  // 2: a, b, d, e, g
0b01001111,  // 3: a, b, c, d, g
0b01100110,  // 4: b, c, f, g
0b01101101,  // 5: a, c, d, f, g
0b01111101,  // 6: a, c, d, e, f, g
```

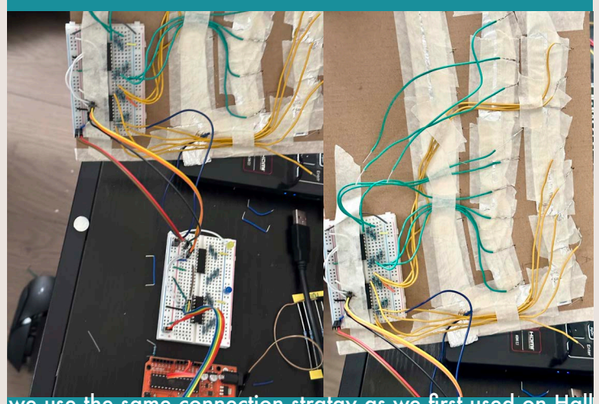
There is two pins for common, should only connected one of them to GND, otherwise the operation of LED will be affected according to our expriment.

Production Process for Outputs

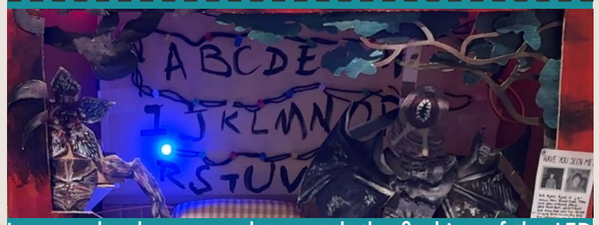
Letter Wall



In order to restore the Letter wall in the play, we considered using light strings, but most of the light strings cannot be directly programmed and need to be purchased additionally, so we chose to use existing LEDs, plug them into the cardboard and wire them behind the cardboard. Use yarns at the front to replace the "light string".



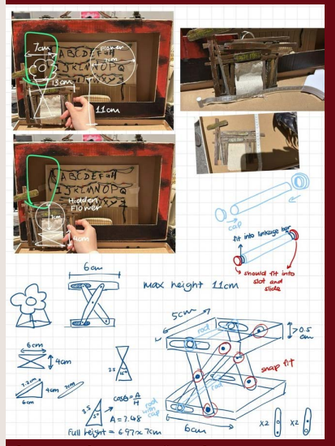
we use the same connection stratgy as we first used on Hall sensor: cover all the Cathode of LEDs with a folded kitichen foil. We make sure the thickness of foil and contact area between foil and LEDs cathode is big enough for large current pass through. Each anodes of LEDs are soldered to a 220 ohm resistor to limit the current, then soldered to the customized wires mentioned above for connection to breadboard. The 74HC595 shift register is connected correctly according to datasheet and its clock pin and latch pin are shared with other parts. Also, to make 4 stage cascade shift register, the closer register's serial output pin is connected to farther register' serial input. In this way there is only one serial output pin.

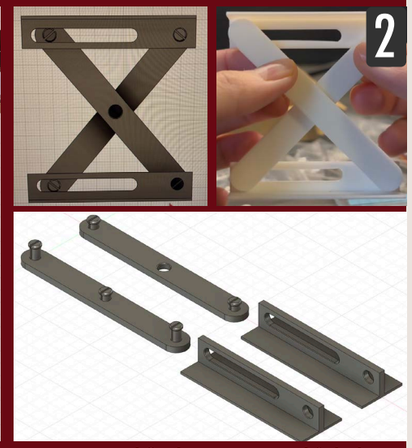


In many levels, we need to match the flashing of the LED with the audio in the play. We will record the video, and then adjust the delay through the time difference between the LED and the audio sound effect in the video.

DemoGorgon

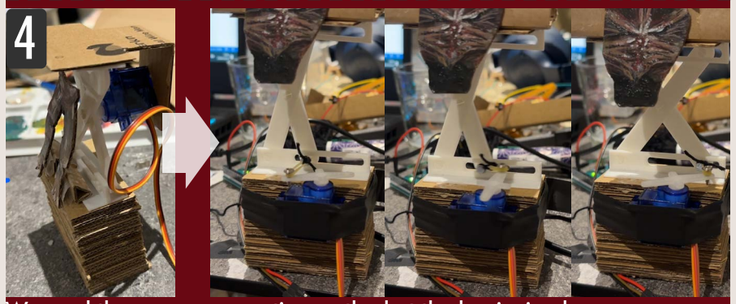
The original plan was to create a 5-petal flower based on this tutorial. However, we did not have access to all the required tools. A futile attempt was made with copper rods, plastic tubes and super glue: the moving parts often jammed up or broke apart.





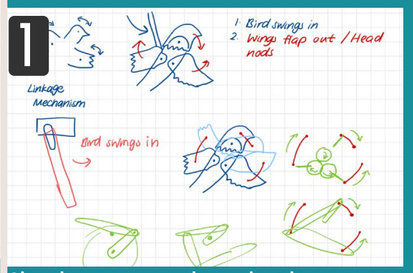
The mechanism for the Demogorgon was simplified to a scissor lift linkage, changing its movement from open-close to vertical translations only. Based on online images and the dimensions previously used for the Demobat pivots, a 3D model for the lift was designed and printed out.

The form of the Demogorgon was sketched out and painted in according to commonly found pictures online. The parts were attached to the scissor lift with supporting cardboard pieces.

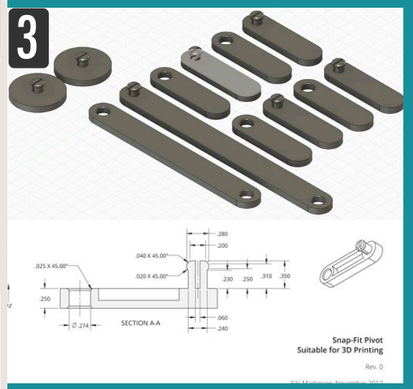


We used the wrong connection method at the beginning because we ignored the slight angular displacement caused by the movement of the connecting rod; therefore we chose to fix the servo on the base and use a small rubber band to drive the connecting rod to move left and right.

DemoBat

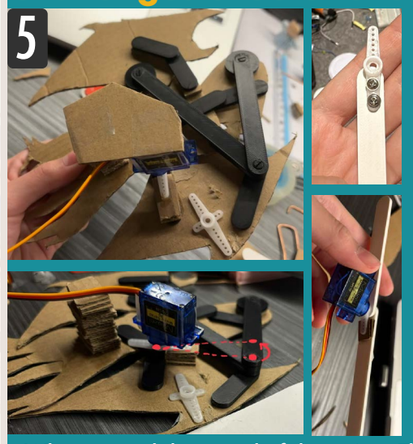


Sketches were made to decide on its desired movements. The wings would flap in opposite directions, emulating the flapping of real bird wings.

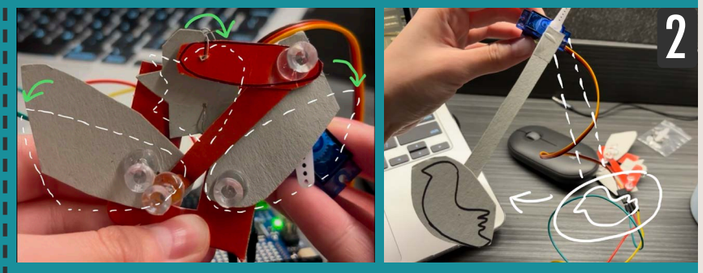


Fusion360 was used to model the pivot hinges. The design and proportions of the components were sourced from Eiki Martinson's website, but they were scaled and reshaped as needed.

Challenges & Solutions



- Optimize servo motor positioning. To ensure that the rotational motion drives the wing or head around its pivot, the servo is best placed above the pivot. Other placements result in unnecessary translational movement, complicating attachment. The second servo enables the Demobat to "swing" into the scene.
- When control the speed of the swing, The randomly generated LED function should be running at the same time, so the clock signal has been occupied. Since the code is run line by line, we call the randomly generated LED function in bat and borrow its delay to adjust the movement of the servos'speed



Lo-Fi prototypes were made out of packaging cardboard and thumbtacks to simulate the pivots and linkages. This acted as concept verification. Although it was simple to figure out the linkage between the head and the right wing, additional support was needed (ChatGPT) on using an "inverted configuration" at the left wing to allow for anti-clockwise movement. It involves placing the linkage before the pivot (left wing) instead of after the pivot (right wing).

Before permanently adhesing the printed linkages to the final cardboard pieces, they were attached to paper versions of the parts using tape. This was to confirm the placements of the pivots and ensure the expected motion. Scotch tape could not provide a stable connection between the printed linkages and the cardboard pieces, which is why the paper was needed.

Electric Circuit Design

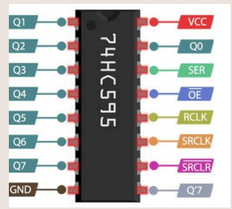
: ?
: ?
: VCC
: Ground
: ?
A0 : Pin number
: Input
: Output

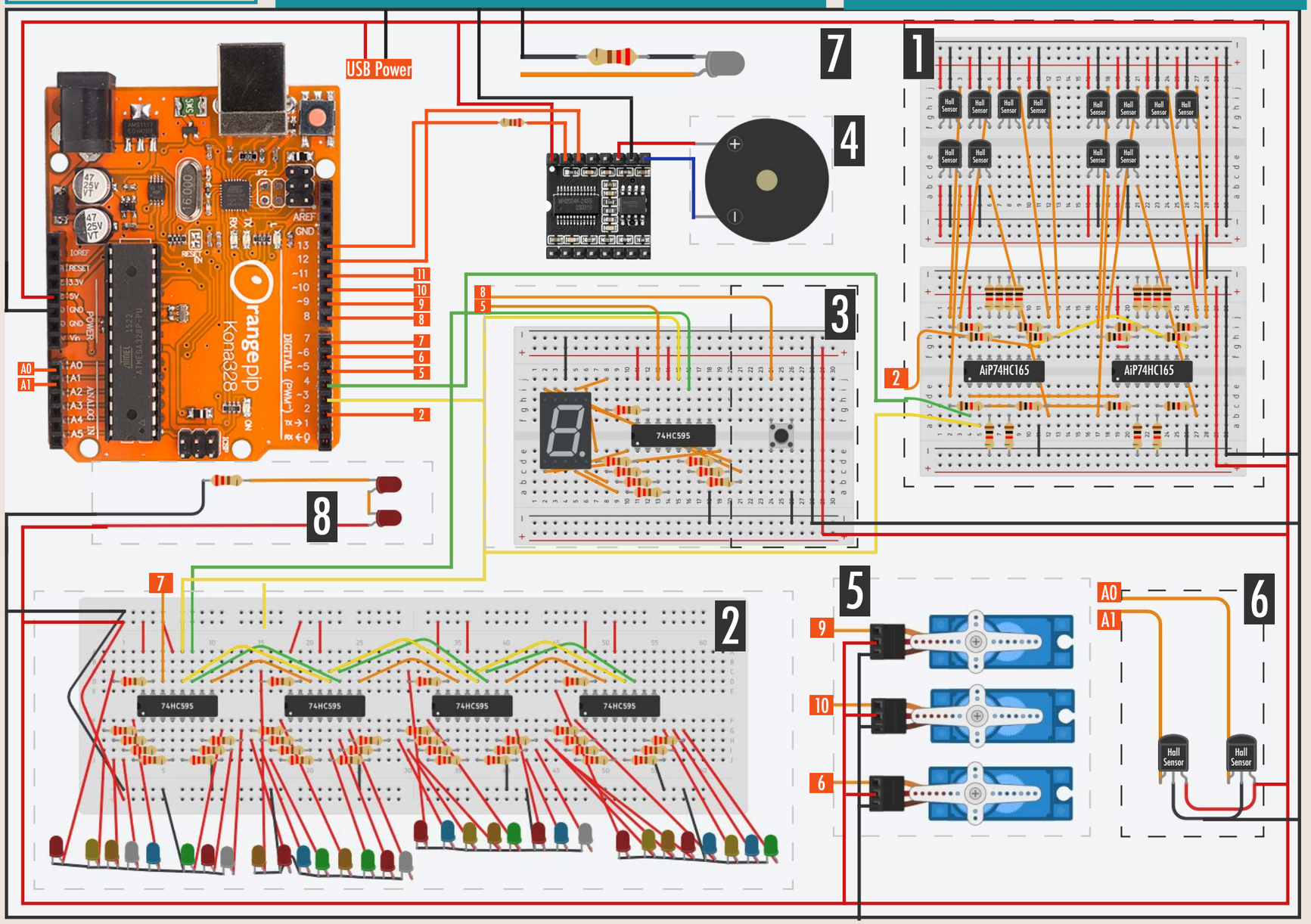
1 Game Board - Hall sensor circuits:

Detect if the player's piece is at certain specific grid.
We choose **49E linear hall sensor** & **Api74HC165 shift register** as a solution due to its low price and compact size, and we could get **status** of **12** hall sensors using only **3 GPIO**s. The 49E hall sensor would output a voltage that change with surrounding magnetic field. Each Api74HC165 connected with 6 hall sensors as parallel input and convert it to serial output that can use one GPIO to read. We make **2** shift register cascade so only need **1** data pin.

2 Light Wall CIrcuit

Individually control the status of the LED that corresponds to each letter from A to Z.
The Arduino send the status of LEDs one by one synchronized with clock signal and latched in shift register, once it received all the status it would update all the output in parallel according to the latch pin status.
We choose SN74HC595 shift register, 220-ohm resistor and normal LEDs as a solution due to the minimized GPIO usage (one data pin, **shared clock pin and latch pin** with other parts)





3 The Digital Dice

Push the button and get a random number (1-6) as steps.
We choose a small LED segment display, SN74HC595 shift register and a four-pin button as a solution due to the minimized GPIO usage and low price.
The button connects to a GPIO that is in INPUT_PULLUP mode so in default GPIO reads HIGH and once button is pressed GPIO would read LOW. The working logic of LED segment display is the same as LED light wall.

4 Speaker & MP3 Player Module

Play the specific script at each stage.We choose DFPlayer Mini MP3 module + 8Ohm 2W speaker as a solution as this cheap and compact has all functionality we need for playing sound and it has mature library that can directly be used.
This module contains a **decoder and amplifier**, can directly convert audio file into magnified analogue signal for speaker. It stores audio file using SD card and communicates with Arduino using **UART**.

Servo motor & Feedback Led

5.motor: control bat and Demogorgon by driven the mechanic structure behind them.
7.Red LED: This is background led that shine when power is on to give player feedback that the game is ready to start.

Power Management

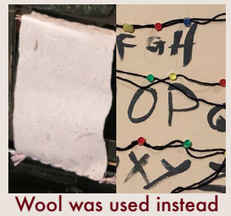
Use a external USB power module to supply 5V to all actuators and sensors, and make common GND from power source to Arduino. This avoid drawing large current through Arduino and burning it.
The maximum current during the playing is below 500mA.

Challenges & Solutions

- 1.49 E **linear hall sensor** output an **analogue** value, in default it would output voltage between 2V and 3V in a 5V supply. To stabilize the readings at shift register we add potential divider circuit to decrease the default voltage under 2V. Further solution would be adding circuit for converting analogue value to digital value, but the cost would be high. For reset and start grid, we directly connect the output of 49E to ADC of Arduino and set an threshold, which is another way of **converting** signal from **analogue to digital.**
- 2.In light wall, all LED are **common cathode and connect to GND**, it is hard to wire them so we use kitchen oven foil to make the path of GND behind the light wall, which is very practical and sustainable.
- 3.**sound player** communicate with Arduino using **UART**, but default UART pin 0 and 1 is taken up when connecting to computer. So we use software serial to set pin 12, 13 as RX and TX. Also, the module works at 3.3V but Arduino works at 5V, to avoid damage to mp3 module, we add a resistor between Arduino's TX and module's RX.

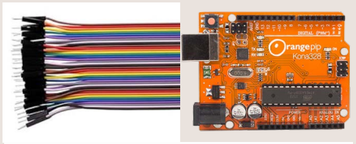
Bill of Materials

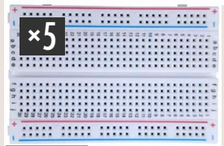
Non-electrical Components

	Gift Box	Cardboard	A4 Paper	Cotton wire	Paint	Clay
How is the material used	 The cardboard box was used to make the casing of the board game, with the deep side used for the scene and the shallow side used for the chessboard.	 Almost everything is made of cardboard, which can be easily punched and serves as a lightweight support structure without putting too much strain on the motor.	 A4 paper was used to make all parts that did not require any structural support, mainly foreground decoration to cover wires.	 Wool was used instead of wires to simulate light strings, as our 'light strings' were connected in series behind the 'wall'. and decorative cotton fabrics.	 Acrylic paint was used to color all the parts as it§ covers well, we were using scrap paper so there was a lot to cover.	 Used to make chess pieces, because the friction on the surface after caly dries can easily catch the magnets stuck at the bottom.
Source of material	 Boxes previously given as gifts to family members.	 Used express carton (previously contained a bicycle).	 Used/Printed white A4 Paper.	 I like to do crafts so I have a lot of leftover yarn.	 Bought from Amazon.	 The remaining clay from 1st year design work.
Cost	£0	£0	£0	£0	£8	£0

Electrical Components

Basic Components

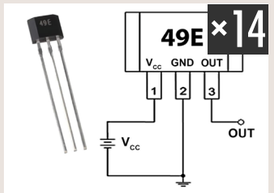




×5 Jump wires, Breadboard, Dupont wires, Soldering kit, Multimeter.

All the basic electronic components are bought previously for last year's project, and the Soldering kit are perchased in July for practicing.

Input



S49E TO-92 3 Pins Linear Hall Effect Sensor

£5.69

https://amzn.eu/d/fYNyCN5

8-Bit Parallel-Load Shift Registers

Borrowed from EE Lab

×1 Micro Momentary Tact Button

From Lab kit

Output

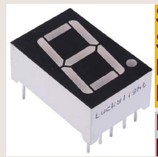
Micro Servo Motor

Leftover from last year's project.

SN74HC595N 8-Bit Counter Shift Registers

£6.99

https://amzn.eu/d/2TniKK2



Seven Segment LED Display

From Lab kit

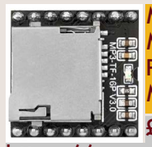
light-emitting diodes

From previous Lab kit

Metal Shell Round Internal Magnet Speaker 2W 8 Ohm

£2

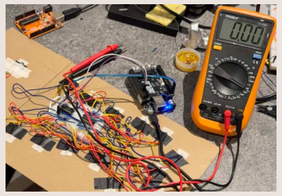
https://amzn.eu/d/a6hliDl



Mini MP3 Player Module

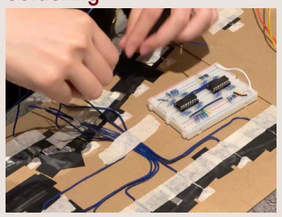
£6.45

https://amzn.eu/d/b8l4Jxq



Testing using Multimeter

Soldering



Wiring

Main Code Logic

Initialization[*Setup()*]
Sets up the initial state of pins, registers, servos, and the MP3 player.

Main Loop[*loop()*]
Contains checks and operations that are continuously evaluated.

Check Button Press[*CheckButton()*]
Checks if the dice roll button is pressed and triggers a dice roll.

Display Number[*displayNumber()*]
Displays a number on the LED segment display.

Check Special Positions[*checkSpecialPositions()*]
Checks for the reset and start positions of the game piece.

Check Position[*checkPosition()*]
Checks the current stable position of the game piece on the board.

Play Script[*playScript()*]
Plays specific audio tracks and controls LEDs and motors based on the game's current checkpoint.

Update LEDs[*updateLEDs()*]
Updates the state of the LEDs using shift registers.

Servo[*batswing(); demogorgon();*]
Control sevros and LEDs for specific stage

Player control pieces jumps over grids one by one, during this the check position don't call play script as there is no stable reading at hall sensor, once the piece is placed on certain grid and remain stable for about 1s, the check position in the main loop will call playscript(), with the current game counter as a input parameter.
In playscript(), program go through a case check, find the match case and carry out corresponding code for actuators.
The game plot will go on once the piece is placed on a new position.
To reset game, player put piece on reset grid and the game counter will return to zero.