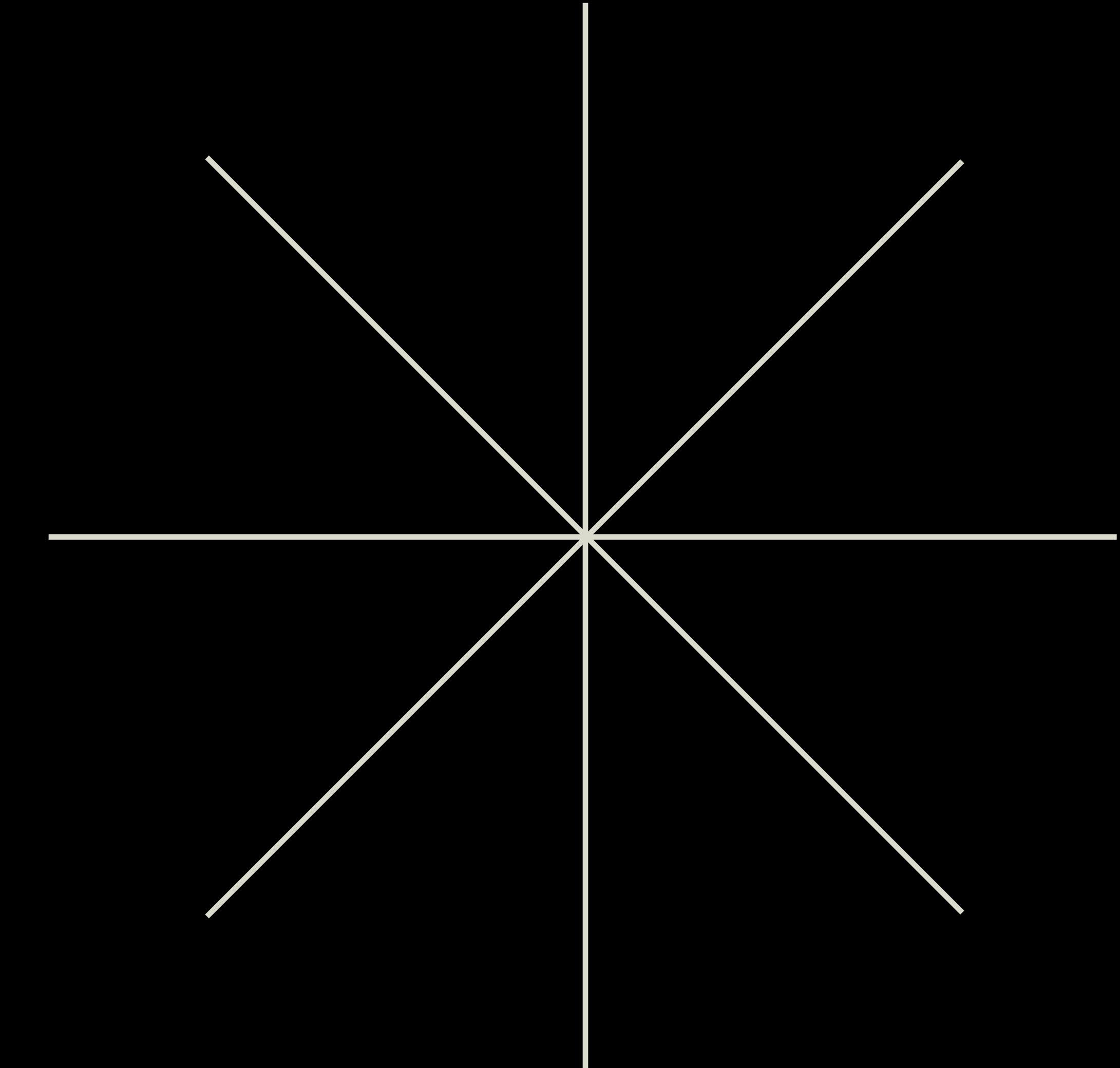


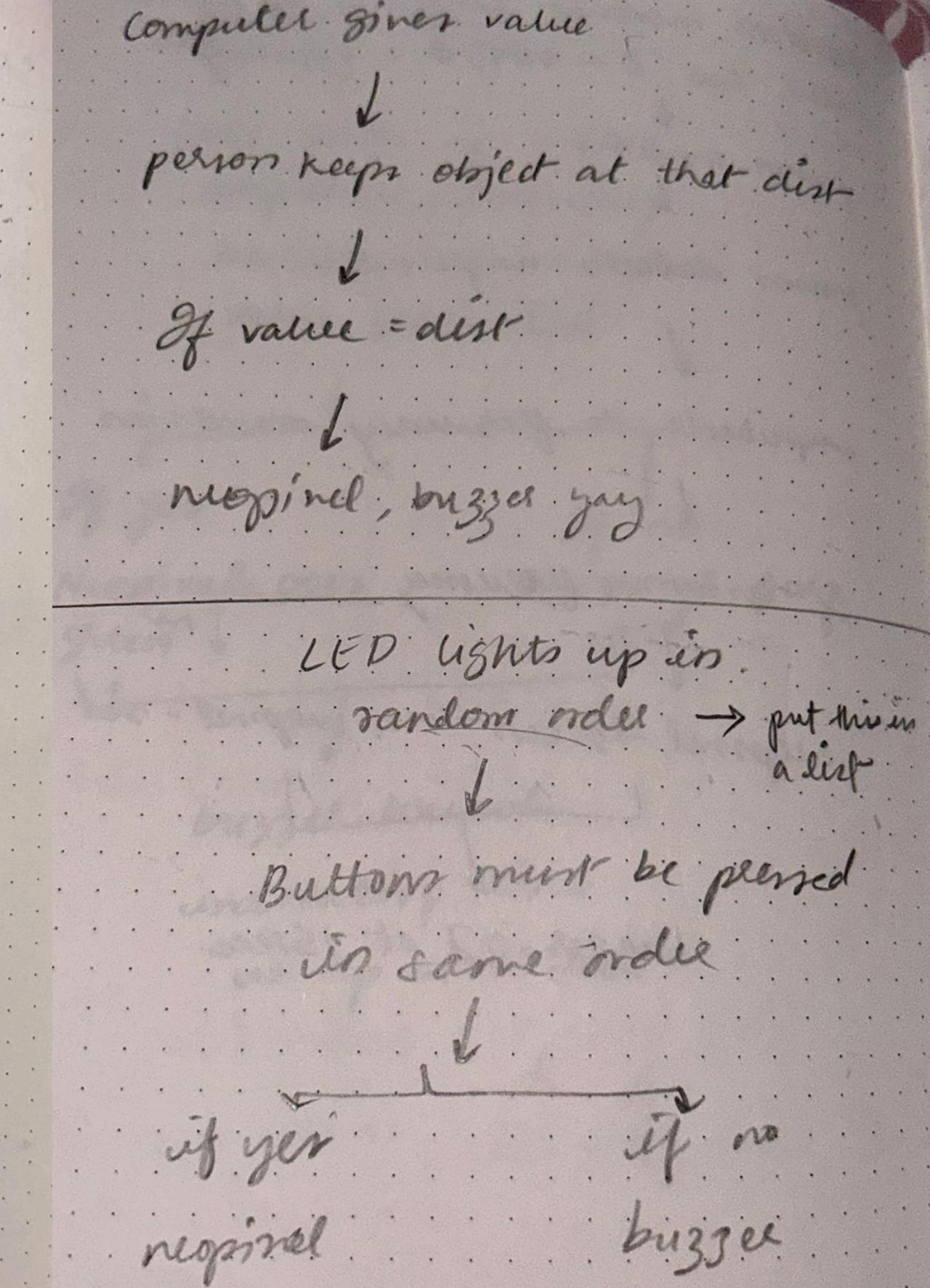
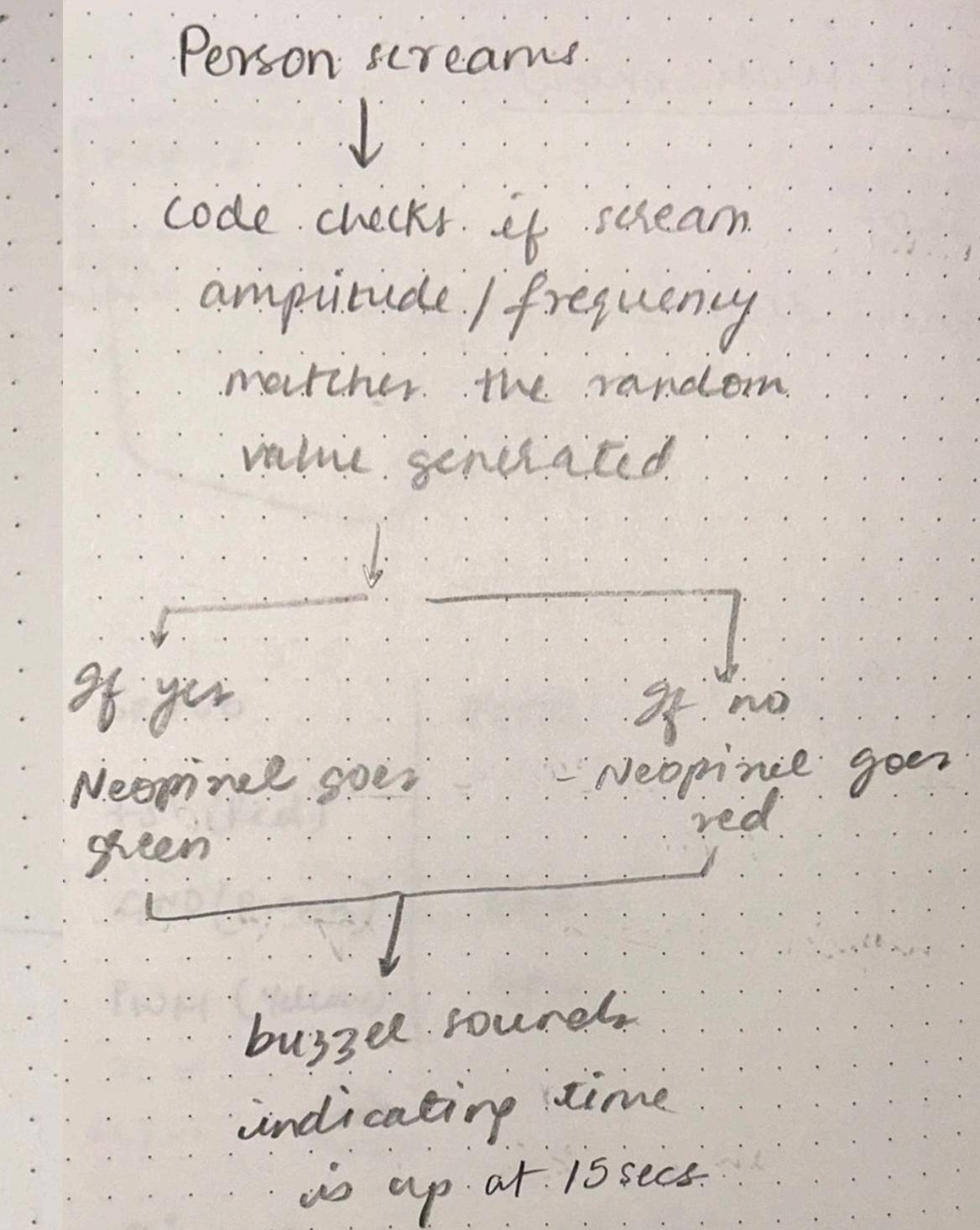
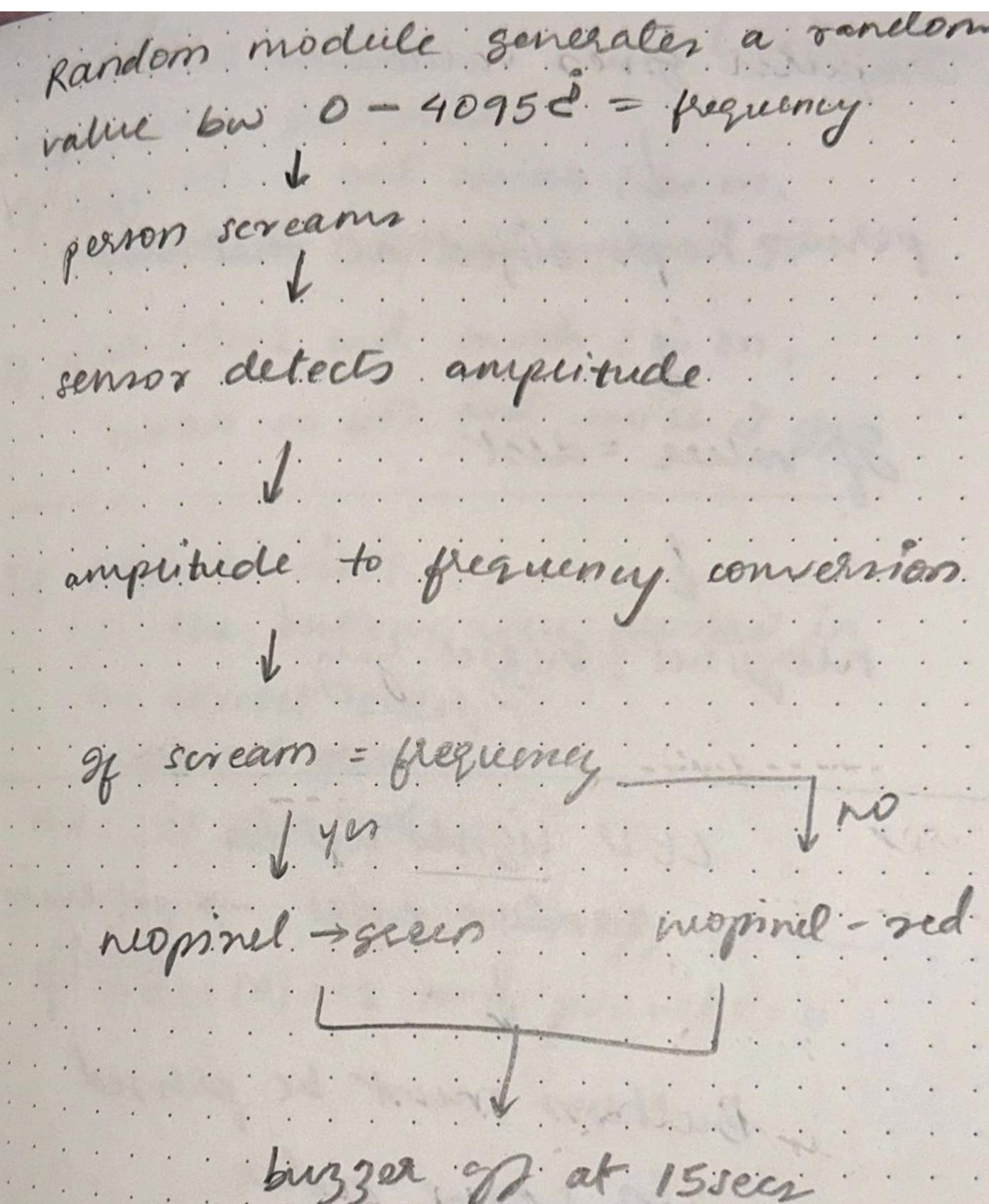
SUMMATIVE ASSESSMENT

OPEN DESIGN TECHNOLOGY



IDEATION TO PROTOTYPE

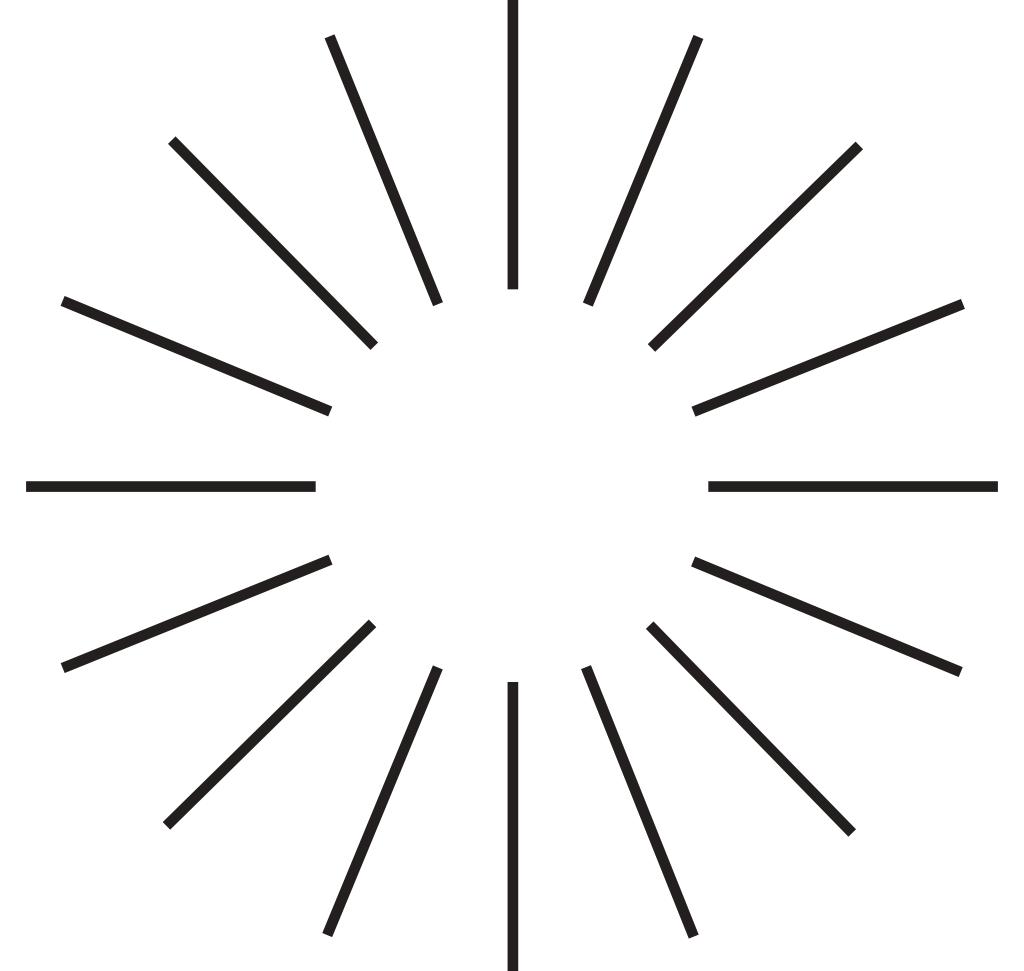
FLOWCHARTS



- user inserts value = [1,2,3]
- value order gets printed
- If order [0]=-1 and switch 1 is on, switch on led1 and switch it off
- If order [1]=-2 and switch 2 is on, switch on ud2 and switch it off
- If order == list1 i.e. the buttons were pressed in the correct order, neopixel green else, it goes red
- If order [0]=-2 and pr-val=0

THE JOURNEY

AT FIRST, WE WANTED TO MAKE A BASKETBALL GAME, THEN A FREQUENCY CHECKING GAME (THE MATH WAS TOO COMPLICATED) AND THEN FINALLY WE LANDED ON MAKING A **MEMORY GAME**.



THE DIFFICULTY IN MAKING THE FREQUENCY CHECKING CODE WAS THAT THE SOUND DETECTOR SENSOR MEASURES SOUND IN AMPLITUDE AND THEN TO CONVERT THAT TO FREQUENCY WE WOULD NEED AN ADC - ANALOG TO DIGITAL CONVERTOR WHICH WAS NOT PRESENT IN OUR SENSOR AND WE WERE NOT CONFIDENT ABOUT BEING ABLE TO USE IT WITH OUR ESP32.

WE KNEW WE WANTED TO DO MAKE A CODE THAT VALIDATES SOMETHING PROVES SOMETHING WRONG. HENCE, WE DECIDED TO MAKE A MEMORY GAME.

WITH THE HELP OF FLOWCHARTS WE EXAMINED WHAT WE NEEDED TO DO AND UNDERSTOOD THE FLOW OF OUR CODE.

THE INITIAL LOGIC

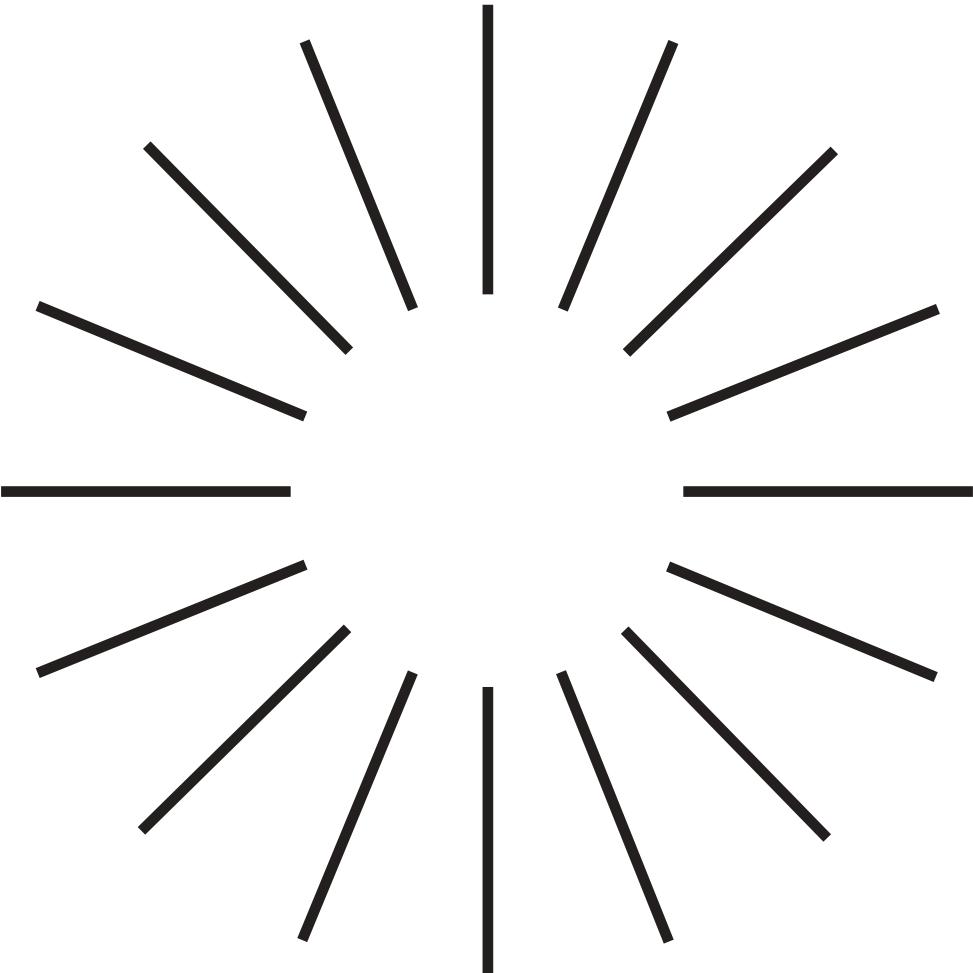
1. OUR LOGIC WAS THAT USING RANDOM MODULE, WE WOULD GENERATE A RANDOM SEQUENCE FOR THE LEDS TO LIGHT UP IN.



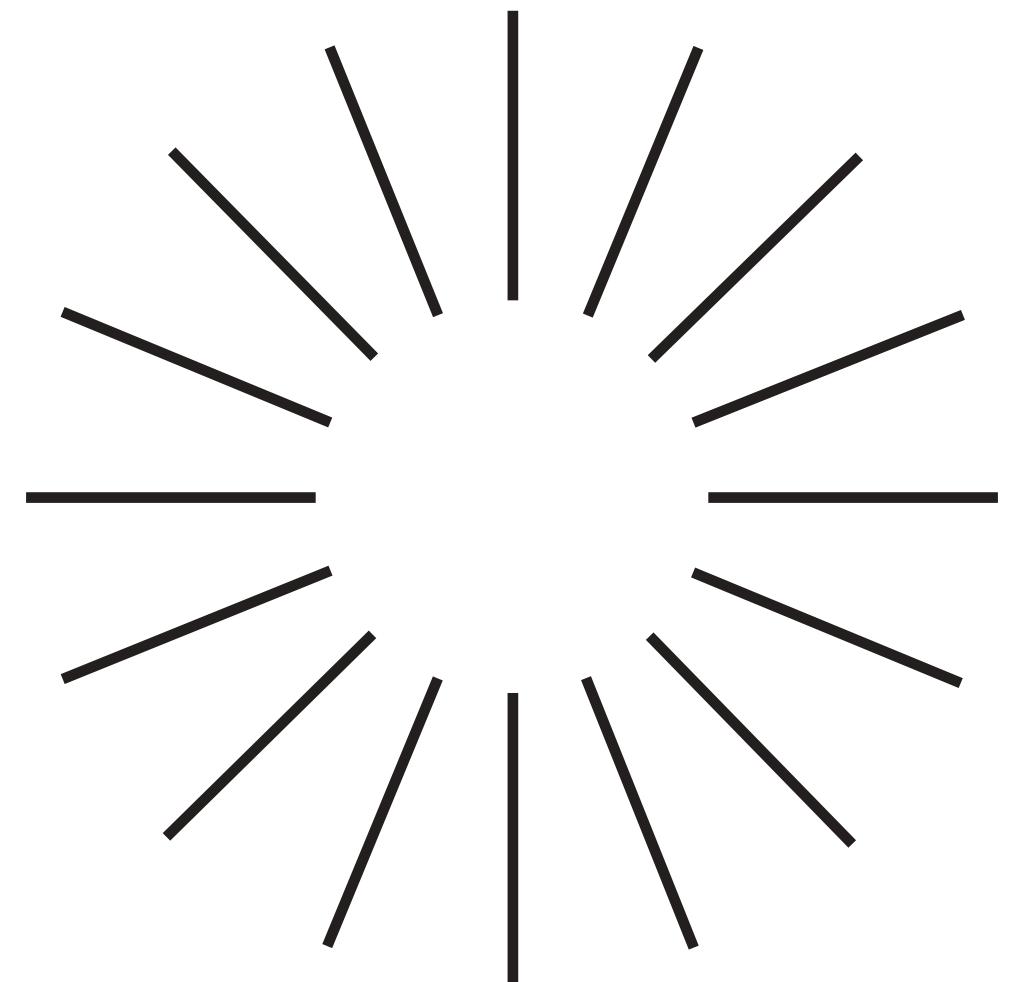
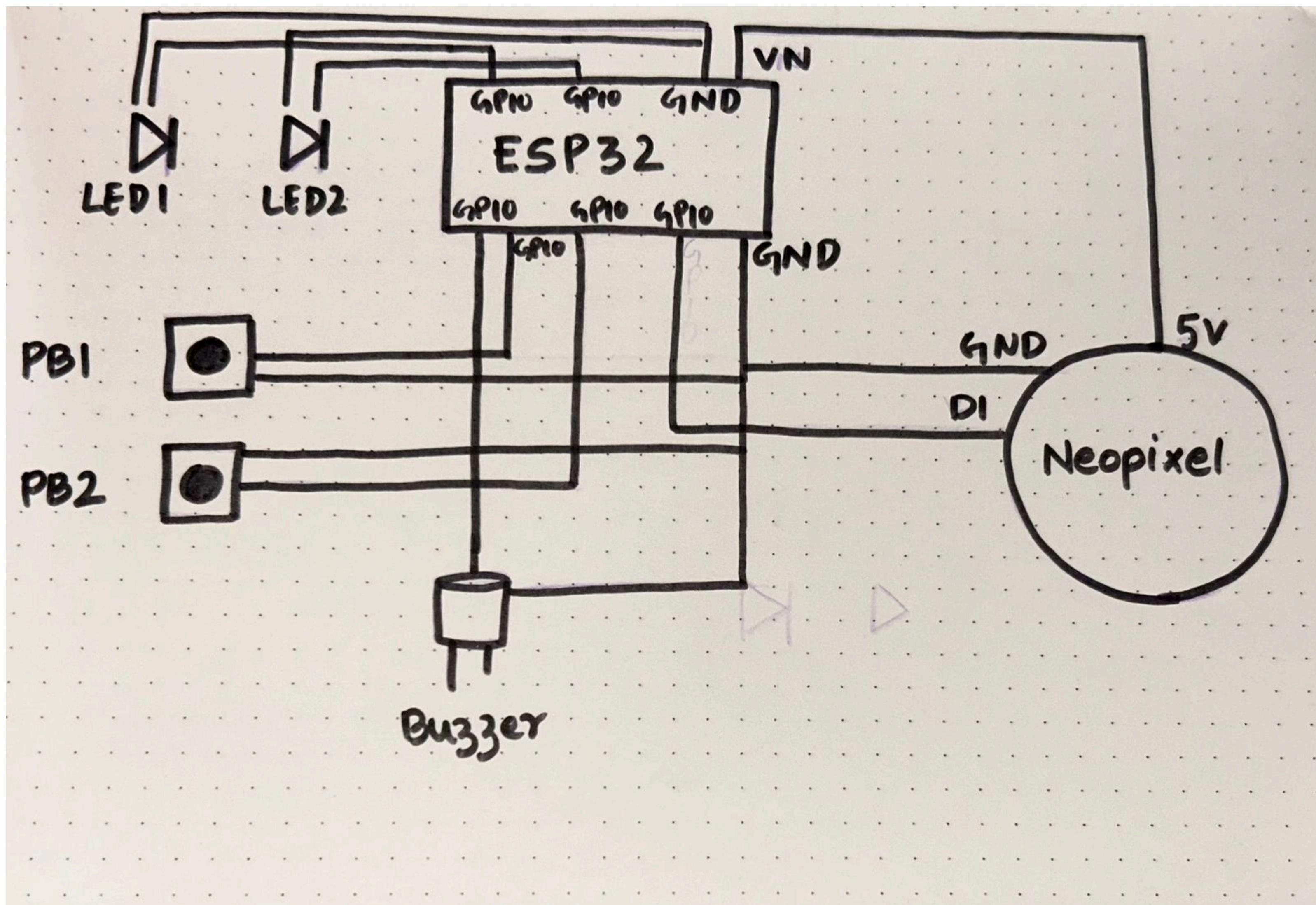
2. PUSH BUTTONS WOULD BE PRESSED IN THE SAME ORDER



3. NEOPIXEL LIGHTS UP AND BUZZER SOUNDS



CIRCUIT DIAGRAM



THE INITIAL CODE WRITTEN BY ME

```
1 from machine import Pin
2 import time
3 import random
4 astro = Pin(25,Pin.OUT) #led 1
5 astro1 = Pin(33,Pin.OUT) #led 2
6 pb= Pin(18,Pin.IN,Pin.PULL_UP) #pushbutton 1
7 p= Pin(32,Pin.IN,Pin.PULL_UP) #pushbutton 2
8 list1=[] #empty list
9 while True:
10     pb_val=pb.value() #initiating pushbutton 1
11     p_val=p.value() #initiating pushbutton 2
12     flow = [] #empty list
13     n = int(input("Enter a value:")) #user input to get sample space
14     order = [random.choice([1, 2, 3]) for i in range(n)]
15     #random order generation from the numbers 1,2,3
16     #print(order)
17     flow.append(order) #adding the random order to the list flow
18
19     #     print(flow)
20     #     Enter a value:3
21     # [[1, 2, 2]] = flow # this was the output
22     if flow[0]==1: #checking if the first digit in the order was 1
23         astro.on() #switch on led
24         time.sleep(0.1)
25         astro.off()
26         list1.append(1) #append the number 1 to another list called list1
27     if flow[1]==2: #checking if the second digit in the order was 2
28         astro1.on() #switch on led
29         time.sleep(0.1)
30         astro1.off()
31         list1.append(2) #append the number 2 to another list called list1
32
33     if flow[i]==list1[i]: #checking if both lists are equal
34         #neopixel goes green, ill write the code for this later i know it
35     else:
36         #neopixel goes red
37
38
39     # i tried this code neeche, tell me if itll work. i cant run it an check rn cuz
40     #i dont have the
41     #hardware for it but is logic and syntax correct?
42     if flow[0]!=1 and pb_val==0:
43         #neopixel goes red, buzzer makes sound
44
45     if flow[1]!=2 and p_val==0:
46         #neopixel goes red, buzzer makes sound
```

```
13     n = int(input("Enter a value:")) #user input to get sample space
14     order = [random.choice([1, 2, 3]) for i in range(n)]
15     #random order generation from the numbers 1,2,3
16     #print(order)
17     flow.append(order) #adding the random order to the list flow
18
19     #     print(flow)
20     #     Enter a value:3
21     # [[1, 2, 2]] = flow # this was the output
22     if flow[0]==1: #checking if the first digit in the order was 1
23         astro.on() #switch on led
24         time.sleep(0.1)
25         astro.off()
26         list1.append(1) #append the number 1 to another list called list1
27     if flow[1]==2: #checking if the second digit in the order was 2
28         astro1.on() #switch on led
29         time.sleep(0.1)
30         astro1.off()
31         list1.append(2) #append the number 2 to another list called list1
32
33     if flow[i]==list1[i]: #checking if both lists are equal
34         #neopixel goes green, ill write the code for this later i know it
35     else:
36         #neopixel goes red
37
38
39     # i tried this code neeche, tell me if itll work. i cant run it an check rn cuz
40     #i dont have the
41     #hardware for it but is logic and syntax correct?
42     if flow[0]!=1 and pb_val==0:
43         #neopixel goes red, buzzer makes sound
44
45     if flow[1]!=2 and p_val==0:
46         #neopixel goes red, buzzer makes sound
```

THE INITIAL CODE WRITTEN BY ME

```
68     time.sleep(0.3)
69     buzz.off()
70     if order==list1: #checking if both lists are equal
71         neo[0] = (0,255,0) #whatever it is for green
72         neo.write()#neopixel goes green, ill write the code for this later i know it
73         time.sleep(0.2)
74     else:
75         neo[0] = (255,0,0) #red
76         neo.write()#neopixel goes green, ill write the code for this later i know it
77         time.sleep(0.2)#neopixel goes red
78
79
80     #my doubt is, how do i make the else condition ki if button 1 was pressed when the number generated was 2 the neopixel goes red
81     #and the buzzer makes a sound. i tried this code neeche, tell me if itll work. i cant run it an check rn cuz i dont have the
82     #hardware for it but is logic and syntax correct?
83     if order==list1 and pb_val == 0:
84         #     print("wrong order")
85         #     neo[0]=(255,0,0) #red
86         #     neo.write()#neopixel goes green, ill write the code for this later i know it
87         #     time.sleep(0.2)
88         #     buzz.on()
89         #     time.sleep(0.1)
90         #     buzz.off()
91         #     #neopixel goes red, buzzer makes sound
92
93     if order == 1 and p_val == 0:
94         #     print("long order")
95         #     neo[0]=(255,0,0) #red
96         #     #neo.write()#neopixel goes green, ill write the code for this later i know it
97         #     time.sleep(0.2)
98         #     buzz.on()
99         #     time.sleep(0.1)
100        #    buzz.off() #neopixel goes red, buzzer makes sound
```

THE IMPROVED CODE WRITTEN BY ANSH

```
1 from machine import Pin
2 import time
3 import random
4 import neopixel
5 neo=neopixel.NeoPixel(Pin(21),16)
6 astro = Pin(25,Pin.OUT) #led 1
7 astro1 = Pin(33,Pin.OUT) #led 2
8 pb= Pin(18,Pin.IN,Pin.PULL_UP) #pushbutton 1
9 p= Pin(32,Pin.IN,Pin.PULL_UP) #pushbutton 2
10 buzz = Pin(22,Pin.OUT)
11 list1=[] #empty list
12 sequence = []
13 n = int(input("Enter a value:")) #user input to get sample space
14 for i in range(n):
15     order = [random.choice([1, 2]) for i in range(n)]
16     #random order generation from the numbers 1,2,3
17     print(order)
18     sequence.append(order)
19 while True:
20     pb_val=pb.value() #initiating pushbutton 1
21     p_val=p.value() #initiating pushbutton 2
22     for i in range(n):
23         if sequence[i] == 1:
24             astro.on()
25             time.sleep(0.4)
26             astro.off()
27             time.sleep(0.3)
28
29         if sequence[i] == 2:
30             astro1.on()
31             time.sleep(0.4)
32             astro1.off()
33             time.sleep(0.3)
34
35     # if order[i] ==1:
36     #     astro.on()
37     #     time.sleep(0.3)
38     #     astro.off()
39     #     time.sleep(0.3)
40
41     # if order[i] ==2:
42     #     astro1.on()
43     #     time.sleep(0.3)
44     #     astro1.off()
45     #     time.sleep(0.3)
46
47     if sequence==list1: #checking if the first digit in the order was 1
48         astro.on() #switch on led
49         time.sleep(0.3)
50         astro.off()
51         list1.append(1) #append the number 1 to another list called list1
52         print(list1)
53         print("button1 correct")
54     else:
55         print("wrong order")
56         buzz.on()
57         time.sleep(0.3)
58         buzz.off()
59     if sequence==list1: #checking if the second digit in the order was 2
60         astro1.on() #switch on led
61         time.sleep(0.3)
62         astro1.off()
63         list1.append(2) #append the number 2 to another list called list1
64         print("button2 correct")
65     else:
66         print("wrong order")
67         buzz.on()
```

THE FINAL CODE

```
from machine import Pin
import time
import random
import neopixel

pb = Pin(18, Pin.IN, Pin.PULL_UP)
p = Pin(32, Pin.IN, Pin.PULL_UP)
ir=Pin(26,Pin.In,Pin.PULL_UP)
neo=neopixel.NeoPixel(Pin(14),16)
astro = Pin(25, Pin.OUT)
astro1 = Pin(33, Pin.OUT)
buzz = Pin(22, Pin.OUT)

# led = []
# inputbutton = []
length = 3
ir_val=ir.value()
if ir_val==0:
    while True:

        led = []

        for i in range(length):
            led.append(random.choice([1,2]))

        print("sequence:", led)

        for i in range(length):
            if led[i] == 1:
                astro.on()
                time.sleep(0.4)
                astro.off()
                time.sleep(0.3)

            if led[i] == 2:
                astro1.on()
                time.sleep(0.4)
                astro1.off()
                time.sleep(0.3)

        inputbutton = []
        for i in range(length):
            while True:
                if pb.value() == 0:
                    inputbutton.append(1)
                    astro.on()
                    time.sleep(0.2)
                    astro.off()
                    time.sleep(0.2)
                    print("Pressed: 1")
                    time.sleep(0.3)
                    break
                if p.value() == 0:
                    inputbutton.append(2)
                    astro1.on()
                    time.sleep(0.2)
                    astro1.off()
                    time.sleep(0.2)
                    print("Pressed: 2")
                    time.sleep(0.3)
                    break

        if inputbutton == led:
            print("Correct")
            time.sleep(0.1)
            buzz.on()
            time.sleep(0.1)
            buzz.off()
            time.sleep(0.1)
            buzz.on()
            time.sleep(0.1)
            buzz.off()
            for i in range(0,16):
                neo[i]=(0,255,0)
                neo.write()
                time.sleep(0.1)
            for i in range(0,16):
                neo[i]=(0,0,0)
                neo.write()
                time.sleep(0.7)
                neo[i]=(0,0,0)
                neo.write()
                time.sleep(0.3)

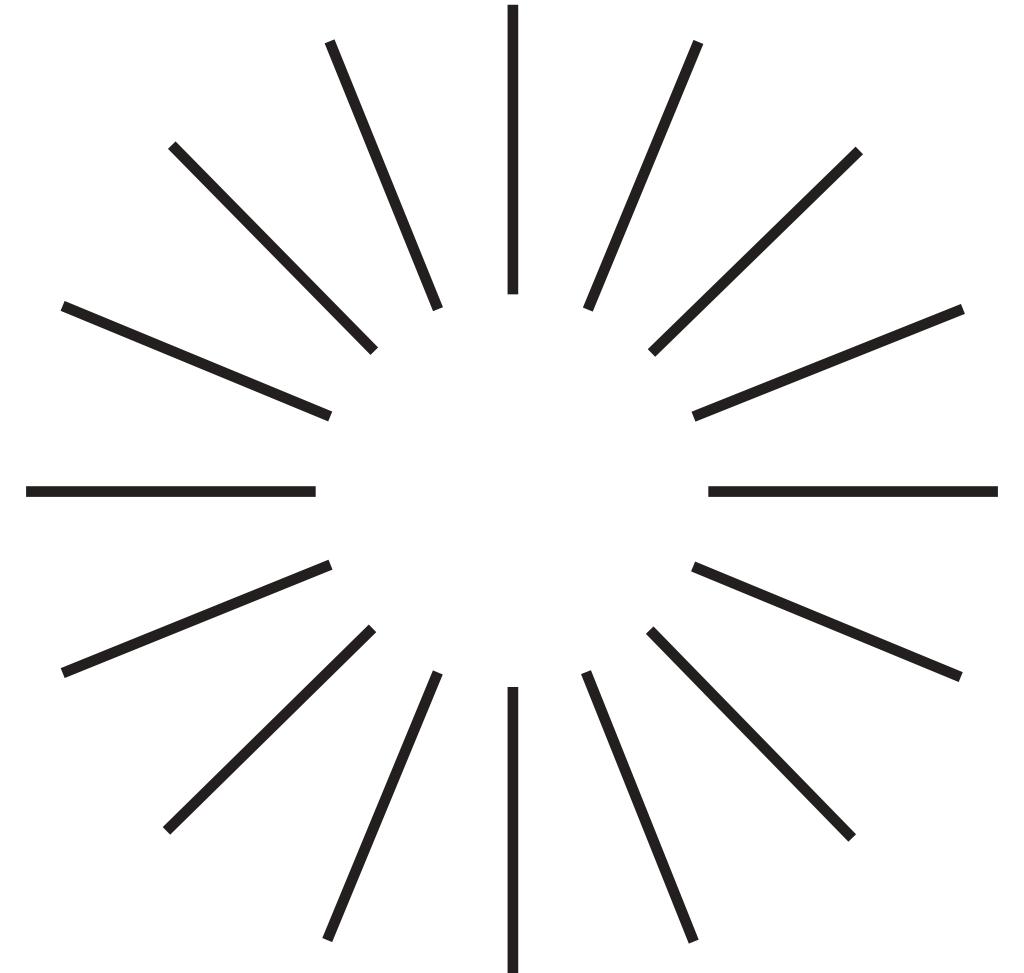
        else:
            print("Wrong")
            buzz.on()
            time.sleep(0.5)
            buzz.off()
            time.sleep(0.1)
            for i in range(0,16):
                neo[i]=(255,0,0)
                neo.write()
                time.sleep(0.7)
            for i in range(0,16):
                neo[i]=(0,0,0)
                neo.write()
                time.sleep(0.7)
                neo[i]=(0,0,0)
                neo.write()
                time.sleep(0.3)

        time.sleep(2)
```

LEARNINGS AND PAIN POINTS

PAIN POINTS :

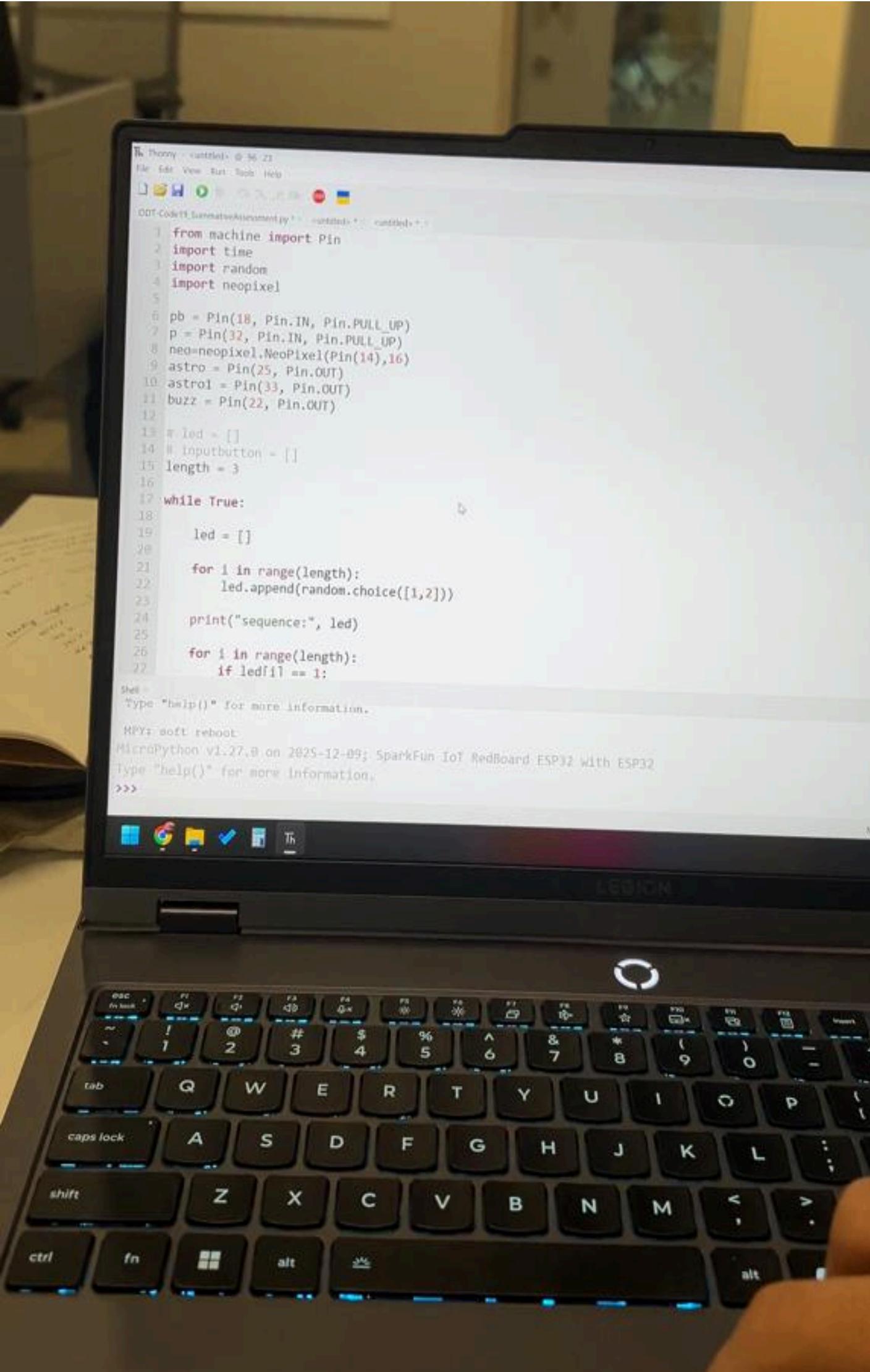
- OUR COMPONENTS STARTED MALFUNCTIONING AT DIFFERENT POINTS IN TIME.
- IN TERMS OF OUR CODE, WE/I HAD TO ITERATE AND UNDERSTAND HOW TO USE MULTIPLE LISTS TO COMPARE DIFFERENT SETS OF DATA FROM DIFFERENT COMPONENTS.
- I.E: IT WAS A LEARNING EXPERIENCE TO UNDERSTAND THAT WE NEEDED TWO DIFFERENT 'WHILE TRUE' LOOPS WITH TWO SEPARATE EMPTY LISTS TO COMPARE DATA FROM THE RANDOM LED SEQUENCE AND THE PUSH BUTTONS.



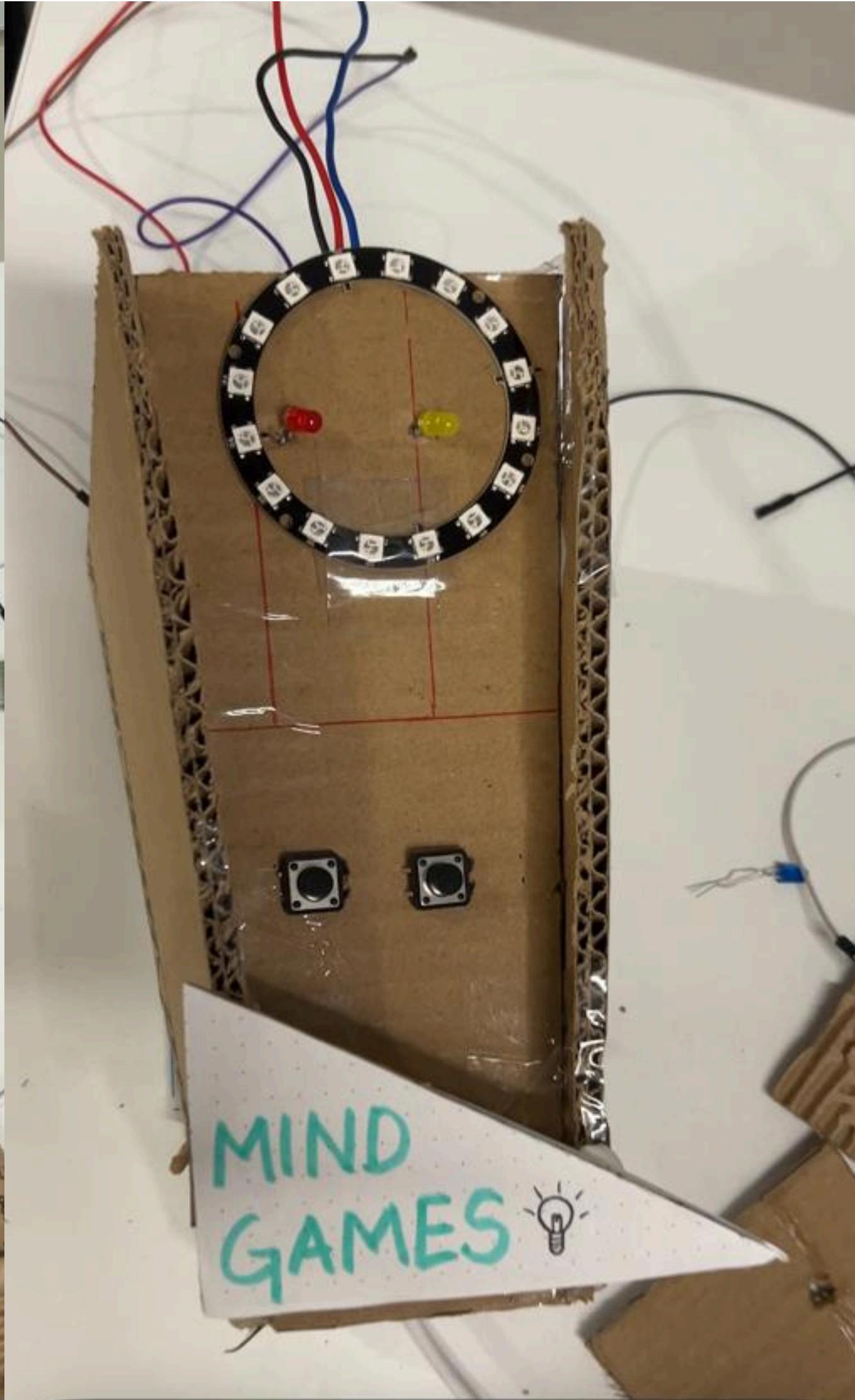
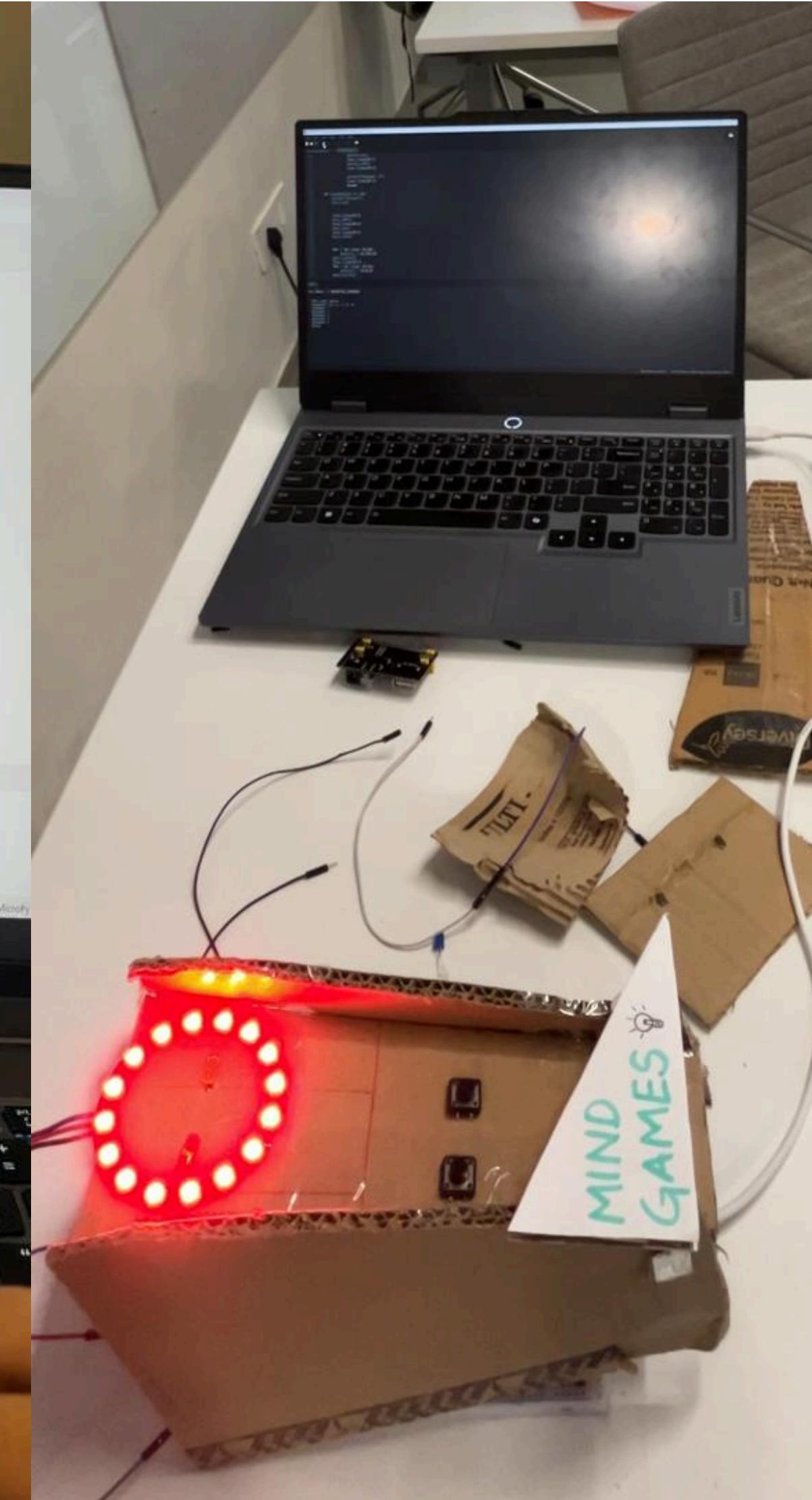
LEARNINGS :

- I LEARNT HOW TO COMBINE ALL THESE COMPONENTS
- I LEARNT HOW IMPORTANT "TIME.SLEEP()" IS BECAUSE THAT WAS THE REASON OUR BULBS WERE NOT LIGHTING UP MORE OFTEN THAN NOT.
- I ALSO LEARNT HOW TO POWER THE NEOPIXEL FROM OUR LAPTOPS BECAUSE OUR EXTERNAL POWER SOURCE STOPPED WORKING PROPERLY.

PROCESS DOCUMENTATION

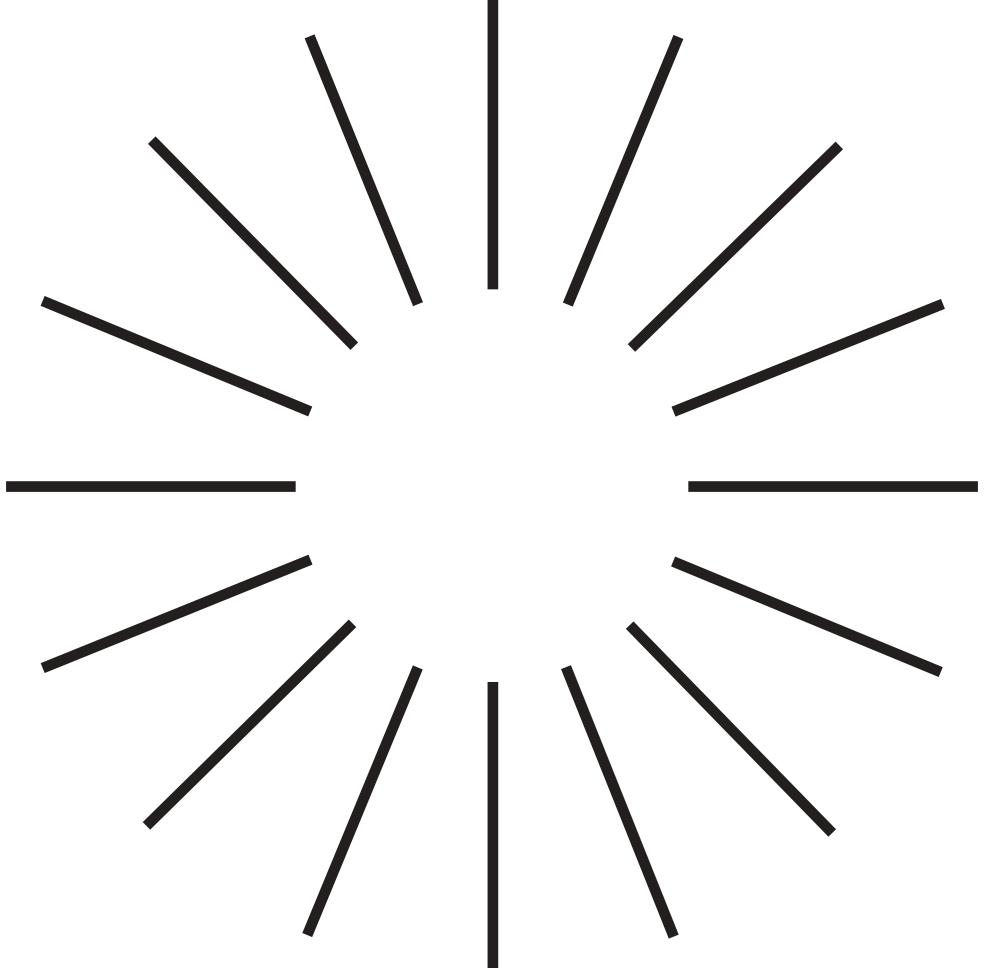


```
Thonny - controller, @ 36.23
File Edit View Run Tools Help
CDT-CodeIt! ScreenshotAssessment.py controller
1 from machine import Pin
2 import time
3 import random
4 import neopixel
5
6 pb = Pin(18, Pin.IN, Pin.PULL_UP)
7 p = Pin(32, Pin.IN, Pin.PULL_UP)
8 neo=neopixel.NeoPixel(Pin(14),16)
9 astro = Pin(25, Pin.OUT)
10 astro1 = Pin(33, Pin.OUT)
11 buzz = Pin(22, Pin.OUT)
12
13 # led = []
14 # inputbutton = []
15 length = 3
16
17 while True:
18     led = []
19     for i in range(length):
20         led.append(random.choice([1,2]))
21     print("sequence:", led)
22     for i in range(length):
23         if led[i] == 1:
24             Type "help()" for more information.
25             MPY: soft reboot
26 MicroPython v1.27.0 on 2025-12-09: SparkFun IoT RedBoard-ESP32 with ESP32
27 type "help()" for more information.
28 >>>
```



INDIVIDUAL CONTRIBUTION

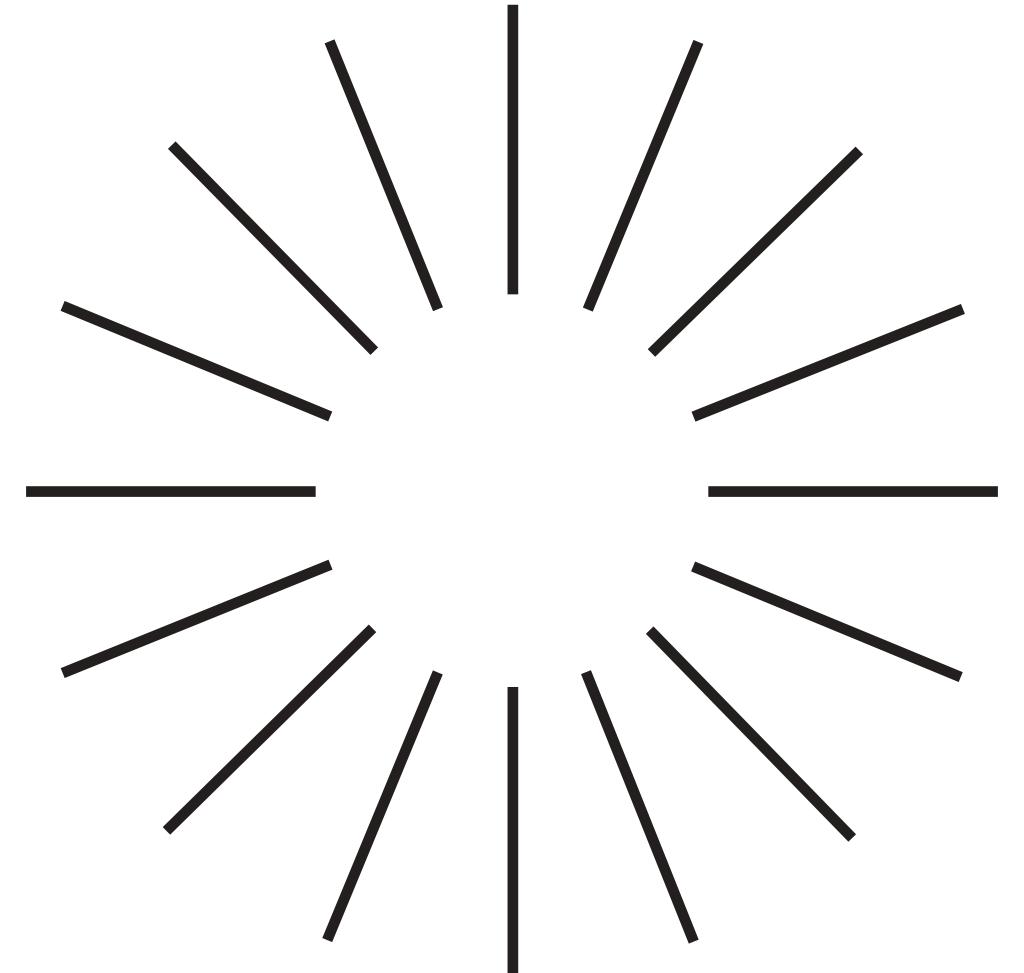
- LOGIC OF THE CODE
- CODING THE BASIC FRAMEWORK
- RANDOM MODULE INITIATION
- COMPARISON OF DIFFERENT DATA
USING TWO LISTS
- INITIATION OF LEDS
- INTEGRATING PUSH BUTTONS
- COMBINING CODES AND STREAMLINING
FLOW OF CODE



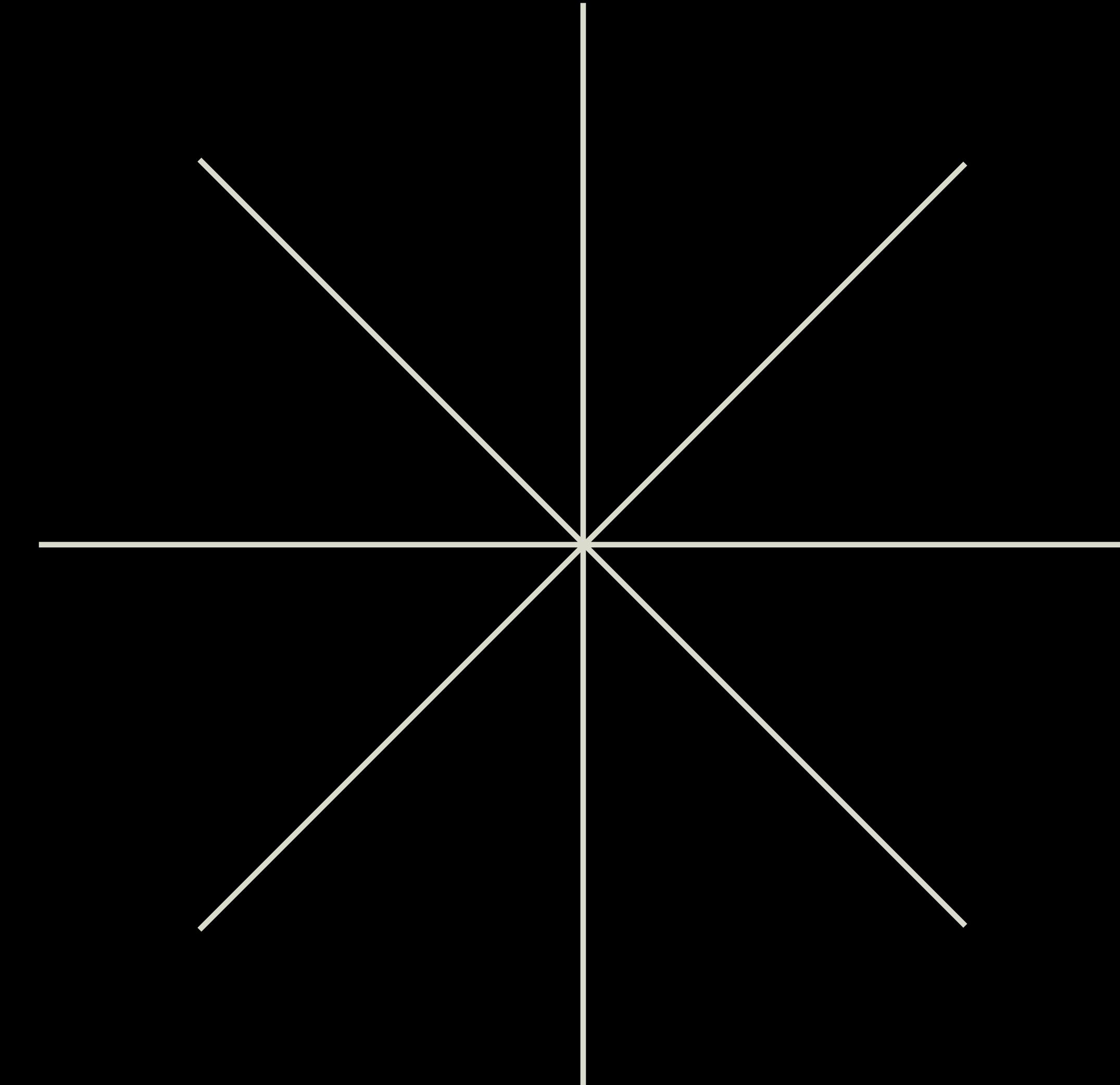
ADDITIONAL INFORMATION

UPON I REALLY LIKED DOING THIS ASSIGNMENT TODAY NOT ONLY BECAUSE IT WAS ABOUT MAKING A GAME BUT BECAUSE IT WAS SUCH A LEARNING EXPERIENCE AND COLLABORATIVE. I'M REALLY GRATEFUL FOR NAYAN, HIS TEACHING TECHNIQUES AND PATIENCE. IT HAS BEEN A GREAT LEARNING ENVIRONMENT.

I PARTICULARLY ENJOY OPEN DESIGN TECHNOLOGY BECAUSE IT CHALLENGES ME TO THINK DIFFERENTLY AND ALWAYS FIND AN EASIER WAY TO DO THINGS AND ALSO BECAUSE I GET TO DO IT WITH MY FRIEND.



SUMMATIVE ASSESSMENT



THANKYOU

AARUSHI PURI