

# “DON’T TOUCH



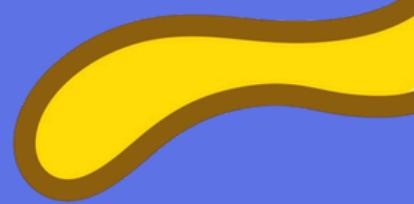
# MY TAIL!”

ODT SUMMATIVE 1

Avantika & Charvi

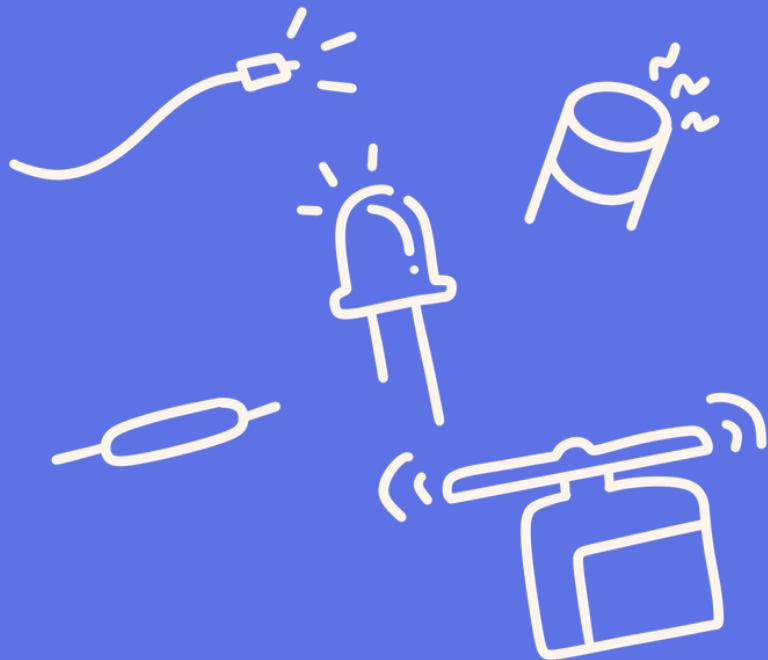
# WHAT LED US TO THIS IDEA?

We first decided to do something related to an animal, and we thought of a cat and its different explosive mannerisms and gestures. We started with one idea (head turning when we touch its tail) and then expanded it to include it reacting to a fish in its paw

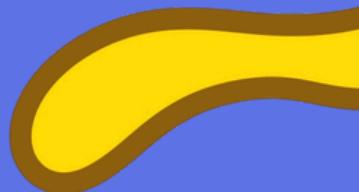




# HOW DID WE DECIDE OUR COMPONENTS?



We went over each of the input and output devices we had and decided to go with two inputs (capacitive touch and reed switch) and three outputs (buzzer, servo motor and LED)



# HOW DID WE DECIDE TECHNICAL FEASIBILITY?

We were convinced it was fairly easy to execute, as it was just a matter of input and output. However, we ran into trouble with our else statement...



# PAIN POINT

Our code read the first condition and immediately went to the else statement to execute it if we weren't holding the wire, which meant it took time to execute

```
while True:  
    c_value=c.read()  
    print(c_value)  
    time.sleep(0.2)  
    r_val=r.value()  
    print(r_val)  
    time.sleep(0.2)  
    if c_value <200:  
        for i in range(900,1400,1): #buzzer  
            b.freq(i)  
            b.duty(600)  
            time.sleep(0.001)  
        for i in range(1400,1100,-1):  
            b.freq(i)  
            b.duty(512)  
            time.sleep(0.001)  
        for i in range(1100,1600,1):  
            b.freq(i)  
            b.duty(600)  
            time.sleep(0.001)  
        for i in range(1600,1200,-1):  
            b.freq(i)  
            b.duty(512)  
            time.sleep(0.001)  
        for i in range (0,120,1): #motor  
            s.duty(i)  
            time.sleep(0.001)  
        for j in range (120, 0, -1):  
            s.duty(j)  
            time.sleep(0.001)  
            l1.duty(512) #led  
            l2.duty(512)  
  
    else:  
        b.duty(0)  
        for i in range(0,256,1):  
            l1.duty(i)  
            l2.duty(i)  
            l3.duty(i)  
            print("l2")  
            time.sleep(0.012)  
  
        for i in range (256,-1,-1):  
            l1.duty(i)  
            l2.duty(i)  
            l3.duty(i)  
            print("l1")  
            time.sleep(0.012)  
        time.sleep(0.01)|
```

Additionally, our cat's head was turning back way too far and it looked kind of freaky

```
time.sleep(0.001)
for i in range (0,120,1): #motor
    s.duty(i)
    time.sleep(0.001)
for j in range (120, 0, -1):
    s.duty(j)
    time.sleep(0.001)
l1.duty(512) #led
l2.duty(512)
l3.duty(512)
time.sleep(0.02)
l1.duty(0)
l2.duty(0)
l3.duty(0)
time.sleep(0.001)
print('1')
```

Last but not least,  
we both wrote our  
code separately  
and combining  
them proved to be  
difficult



# AND OUR LEARNINGS

We simply reduced the time of the else statement and the code instantly became quicker to respond

```
else:  
    b.duty(0)  
    for i in range(0,256,1):  
        l1.duty(i)  
        l2.duty(i)  
        l3.duty(i)  
        print("l2")  
        time.sleep(0.012)  
  
    for i in range (256,-1,-1):  
        l1.duty(i)  
        l2.duty(i)  
        l3.duty(i)  
        print("l1")  
        time.sleep(0.012)  
    time.sleep(0.01)
```

```
else:  
    b.duty(0)  
    for i in range(0,256,1):  
        l1.duty(i)  
        l2.duty(i)  
        l3.duty(i)  
        print("l2")  
        time.sleep(0.002)  
  
    for i in range (256,-1,-1):  
        l1.duty(i)  
        l2.duty(i)  
        l3.duty(i)  
        print("l1")  
        time.sleep(0.002)  
    time.sleep(0.01)
```



And we changed the value to 80 so that the head does not turn all the way back

```
    b.freq(1)
    b.duty(512)
    time.sleep(0.001)
    for i in range (0,80,1): #motor
        s.duty(i)
        time.sleep(0.001)
    for j in range (80, 0, -1):
        s.duty(j)
        time.sleep(0.001)
        l1.duty(512) #led
        l2.duty(512)
        l3.duty(512)
        time.sleep(0.02)
        l1.duty(0)
        l2.duty(0)
        l3.duty(0)
        time.sleep(0.001)
        print('1')
```



# CONTRIBUTIONS

## ACTIVITY [CONCEPT]:

### Features

#### 1. Face turns back when the cat's tail is touched

Input: Capacitative touch (tail)

Output:

- a) Stepper motor (head) moves back fast and forward again slowly
- b) Buzzer (optional) (diff frequency) (meow)
- c) LED heart blinks faster, lights slightly brighter

#### 3. FISH

Input: Reed switch + magnet (fish)

Output:

- a) Buzzer
- b) LED heart blinks faster, light very bright

Figure out a way to ensure LED is on breadboard|

Avantika: no input, Stepper motor + LED

Charvi: reed switch magnet + buzzer + buzzer + LED



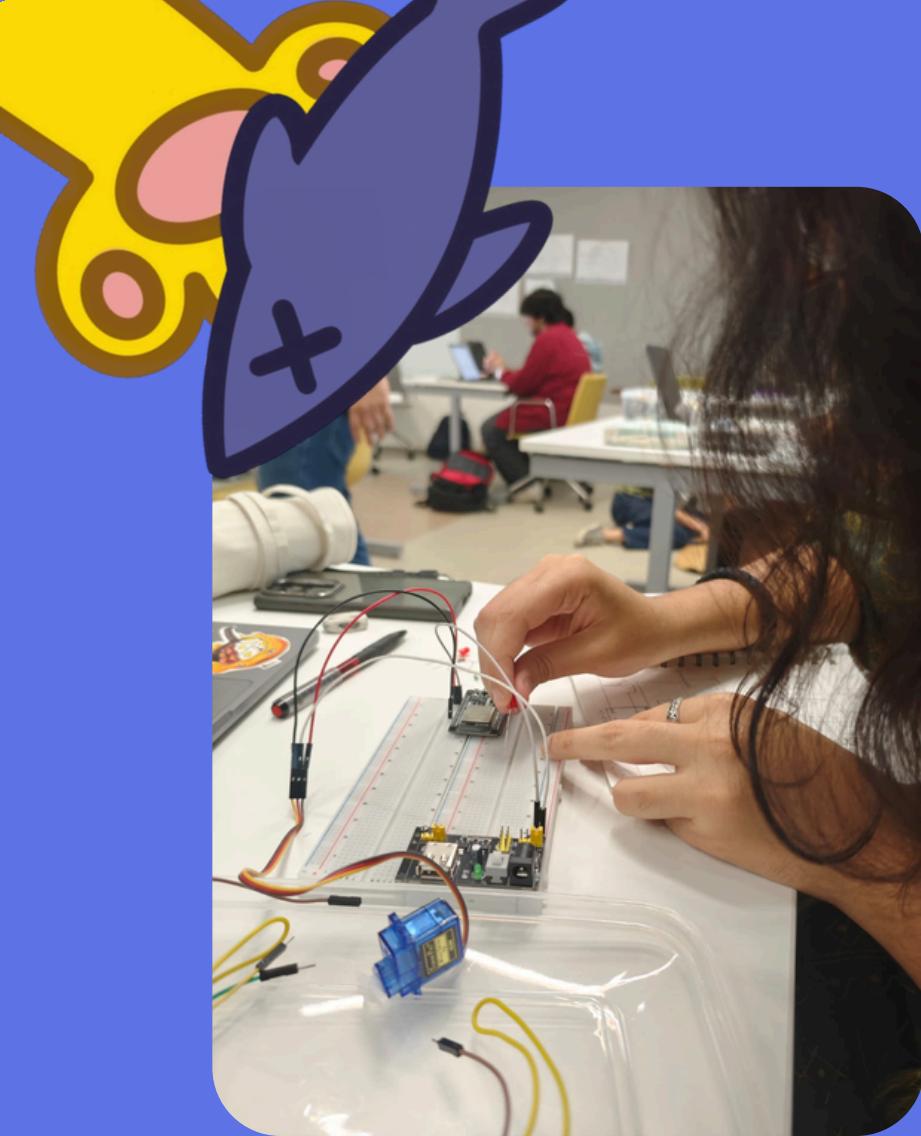
We divided the work  
among ourselves  
accordingly



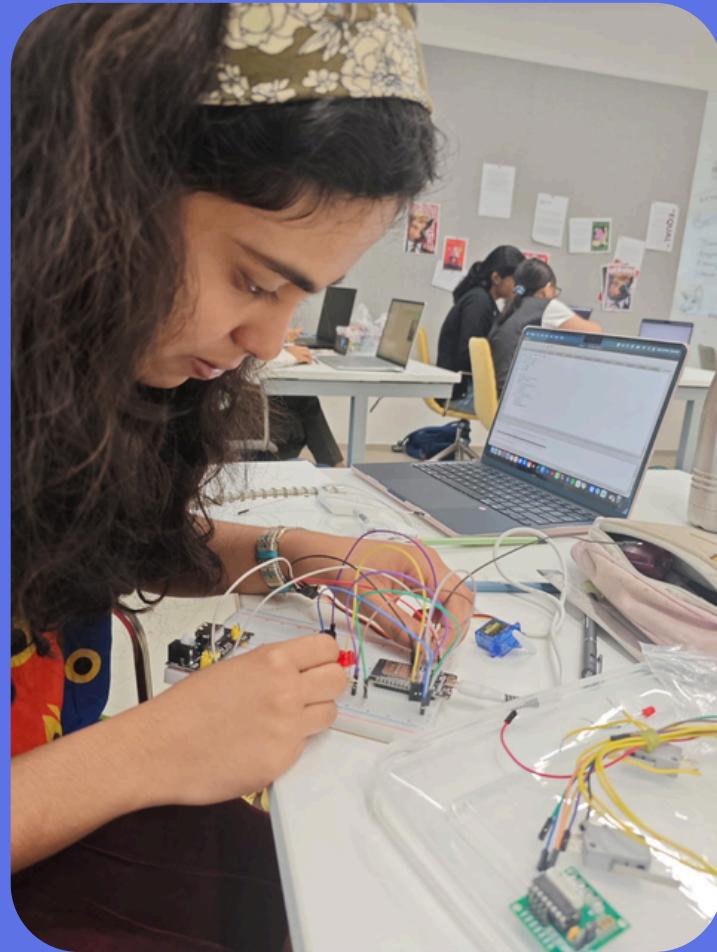
Charvi



Avantika

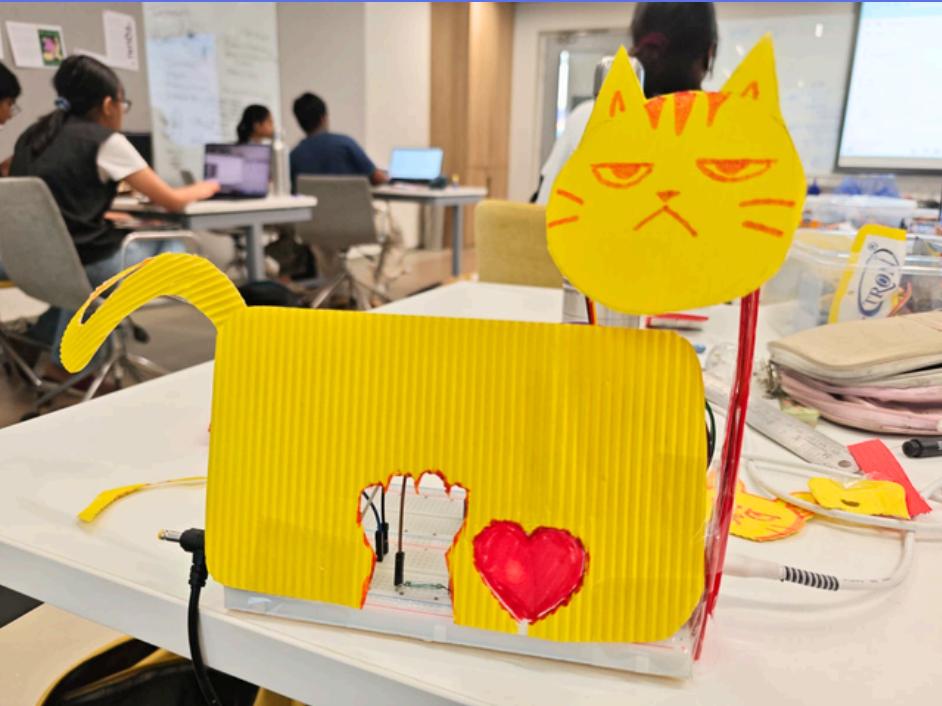


Charvi: connected the  
LED and motor



Avantika: capacitance  
and buzzer

# IF WE HAD MORE TIME...



- We would have added more features to our cat, making use of even more inputs like our sensors
- We had wanted to make the paw extend further to hold the fish, but the placement of the reed switch restricted it

# THANKYOU!



# LINK TO VIDEO:

<https://youtu.be/hdfygokOcUo>

