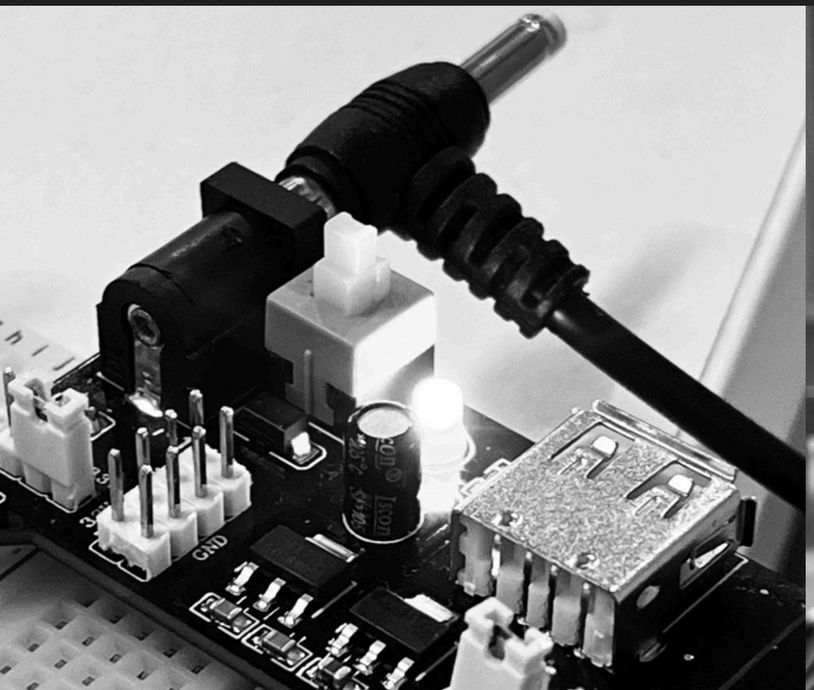
YOU SNOOZE YOU LOSE!

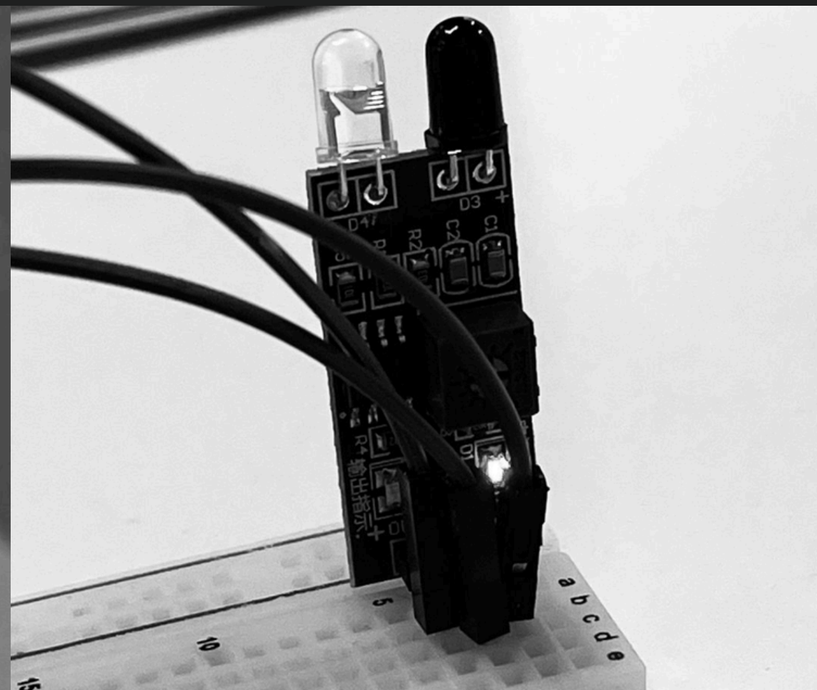
ODT WEEK 08 ASSESMENT - 1

Megha Anand
Siya Mittal

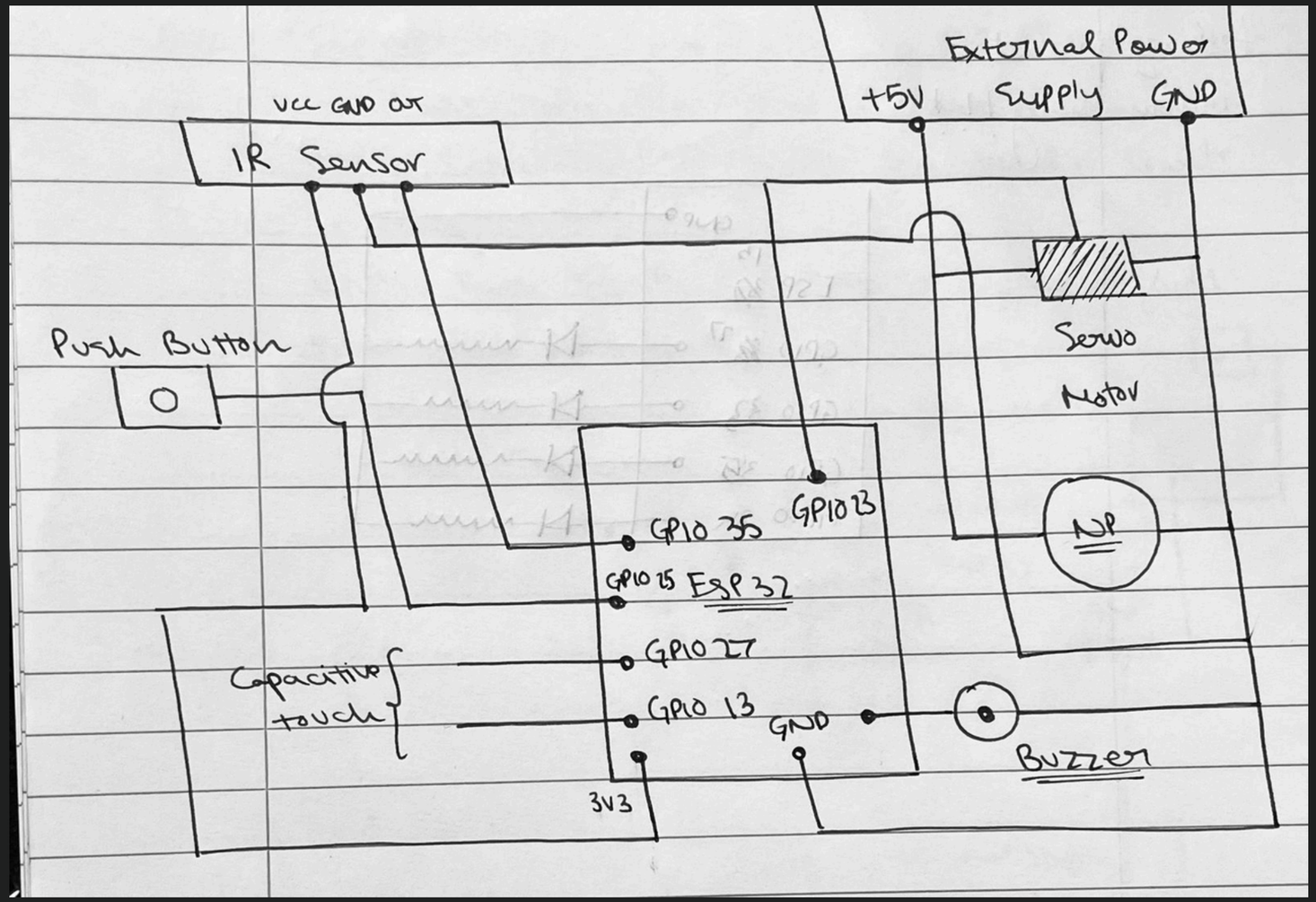
ABOUT OUR PROJECT

Inspiration: Our project is inspired by the classic catching falling sticks game, which tests how quickly someone can react to a sudden movement. We reimagined this idea as an interactive, timed challenge that not only measures reaction speed but also tests how well a person can stay focused over multiple rounds. By adding changing visual cues and sensor-based responses, we turned a simple reflex test into a game that challenges both quick reactions and sustained attention.

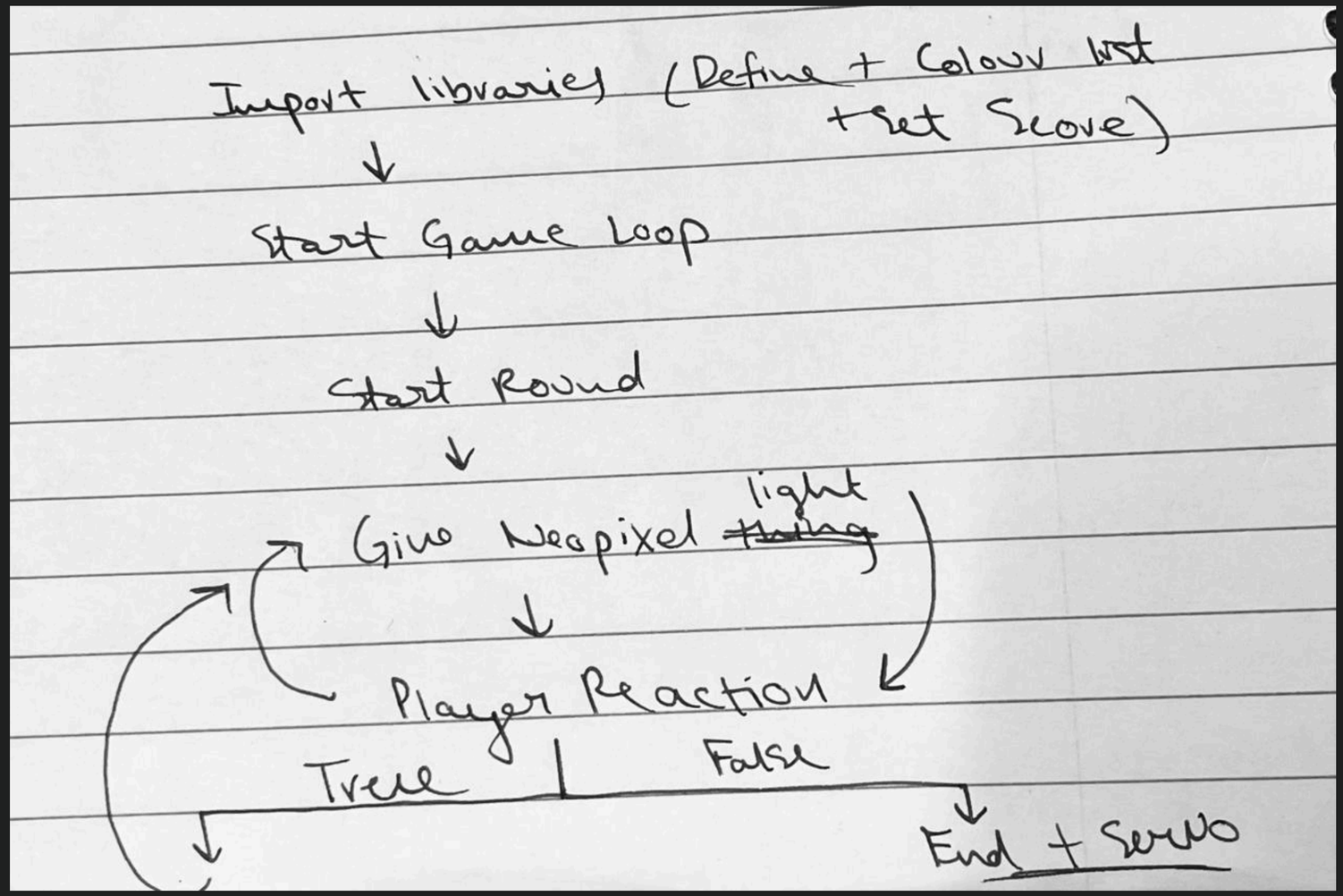




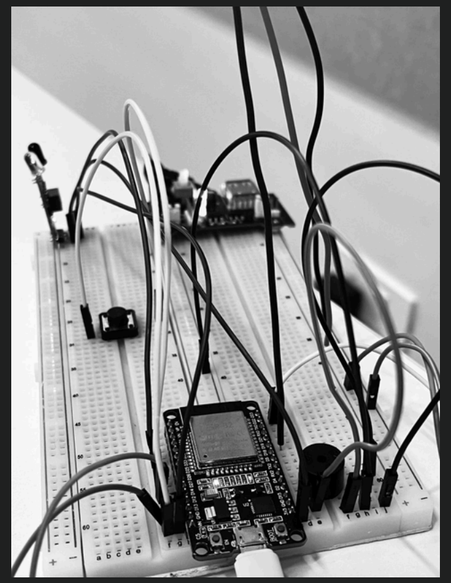
CIRCUIT DIAGRAM

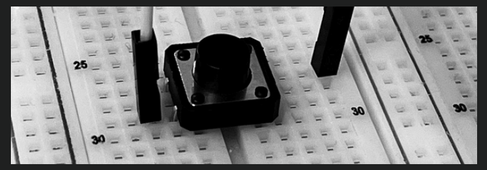


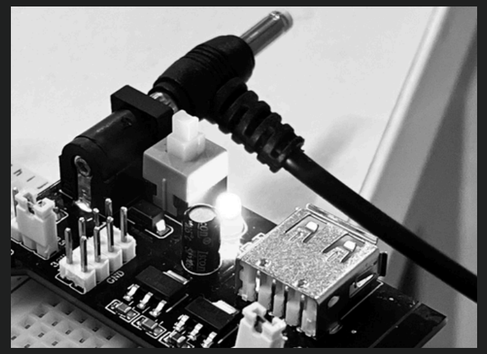
FLOW OF CODE

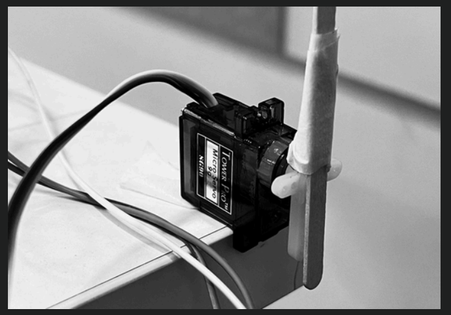

Import libraries (Define + Colour list + Set Score)
Start Game Loop
Start Round
Give Neopixel light
Player Reaction
True
False
End + Servo

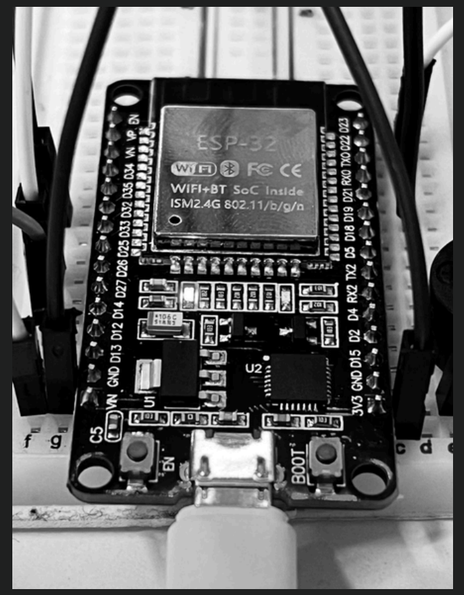
COMPONENTS USED

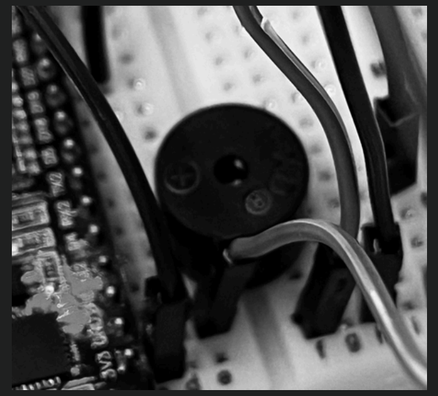


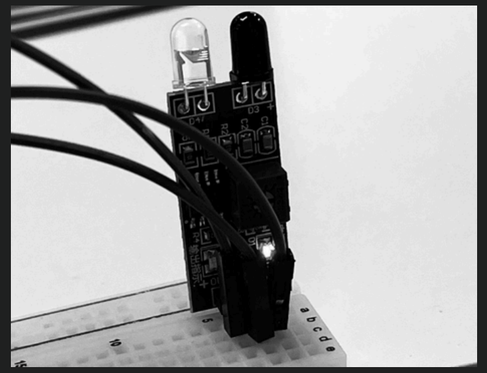












TECHNICAL FEASABILITY

Our initial concept was a straightforward interactive tool to measure task-performance speed. However, we pivoted toward a more gamified experience to increase engagement. We selected three distinct input modalities, prioritizing the reach and range of motion required for each interaction.

To ensure technical reliability, we conducted individual component testing before integrating them into the final circuit.

LOGIC

1. About the code : For the codes structure we used def() to call back all the pieces of code associated with the components in the circuit like servo motor, buzzer, neopixel, IR sensor etc. This allowed for easy repetition of the same piece of code.

2. The core game loop : The software follows the below mentioned flow :
    a. ***Initialization :*** We began by importing necessary libraries, defining variables for the components and initializing the colours and score metrics.
    b. ***The Game Loop :*** A while loop maintains the active state of the game, while for loops and if/elif/else conditional statements handle the logic of each round.
    c. ***Visual Cue & Interaction :*** The system selects a color from a predefined list to trigger the NeoPixel light.
    d. ***Time-Sensitive Capture :*** We used the time.ticks_ms() function to calculate the difference between the visual cue and the Player Reaction.
    e. ***Conditional Outcome :***
        i. ***True (Success) :*** If the reaction is valid, the loop iterates back to the next round.
        ii. ***False (Failure) :*** An incorrect or slow reaction triggers the "End" state, where a Servo motor provides a physical mechanical response (such as resetting a physical element or locking a latch).

PAIN POINTS AND LEARNINGS

1. Neopixel Malfunction
2. Touchpad Read Values
3. Code Loops Nesting
4. Confusion in which colour did what
5. Basic Syntax Errors

1. How to balance more than one element at the same time.
2. Application of time_ticks
3. Efficient code debugging
4. Code sometimes also gives false positives. Scope to improve that.

CONTRIBUTION

Siya Mittal

1. First Half of the code : Importing libraries, def() functions
2. Ideation : how to transform into interactive game
3. Circuit Making

Megha Anand

1. Second Half of the code : All the loops and conditional statements
2. Ideation : What all components we can use
3. Circuit diagram and making

THANK YOU