BARATTH.H

BHUMIKA.V

DRAG RACE

SUMMATIVE ASSESSMENT 1

THE IDEA

As passionate Formula 1 fans, cars and racing are constantly on our minds. So even after exploring multiple concepts and brainstorming various directions, we naturally gravitated toward the idea we were always inclined to pursue.

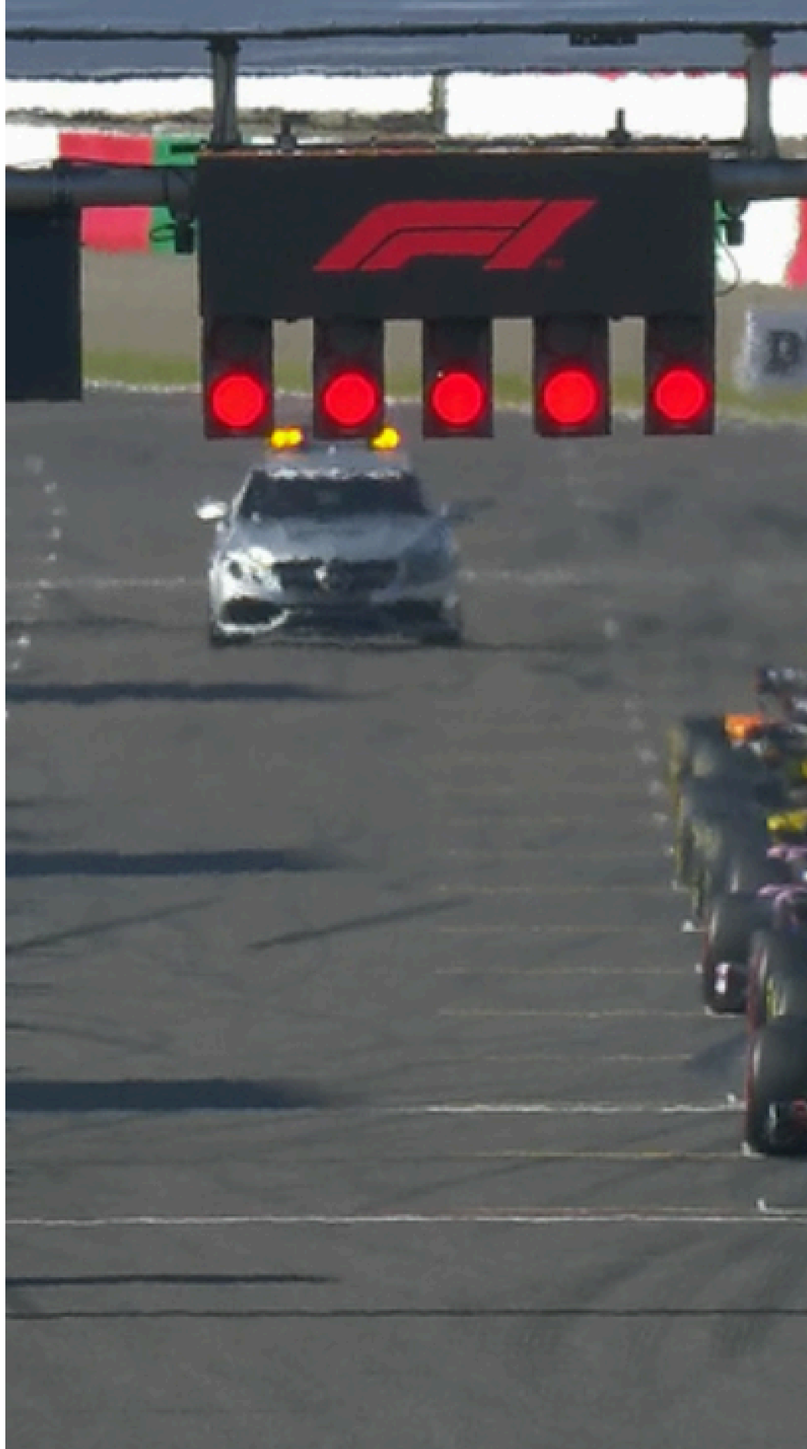
WHAT IS IT?

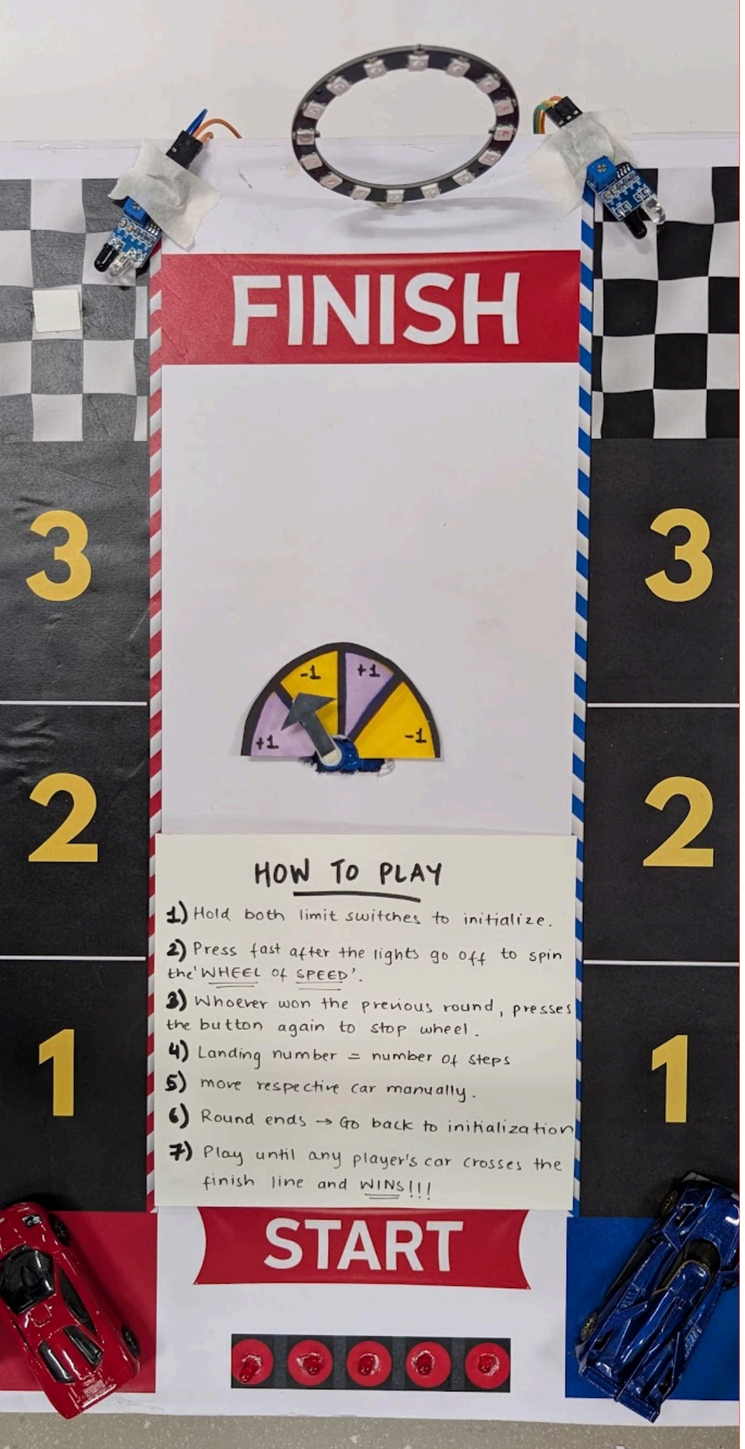
A reflex- and observation-based speed test that propels your car forward in a 1v1 battle against your rival.

In classic ***"lights out and away we go"*** fashion, both players compete in a quick speed challenge. The winner earns the chance to spin the **Wheel of Speed (i.e the servo motor)**.

Once the wheel is spinning, the winning player must stop it at the right moment. Wherever the wheel lands determines how many steps their car moves forward — *for example, +1 step, +2 steps, or even -1 step.*

The first player to reach the finish line wins the race.





THE FLOW

To visually indicate the round winner, we integrated NeoPixel lights, assigning each player a specific color. The NeoPixels flash the respective player's color as immediate feedback, reinforcing the competitive aspect of the game.

For movement mechanics, we used a servo motor to power the "Wheel of Speed." The angular position of the servo determined the number of steps a player could move forward (e.g., +1, +2, -1). This translated rotational position into discrete game outcomes.

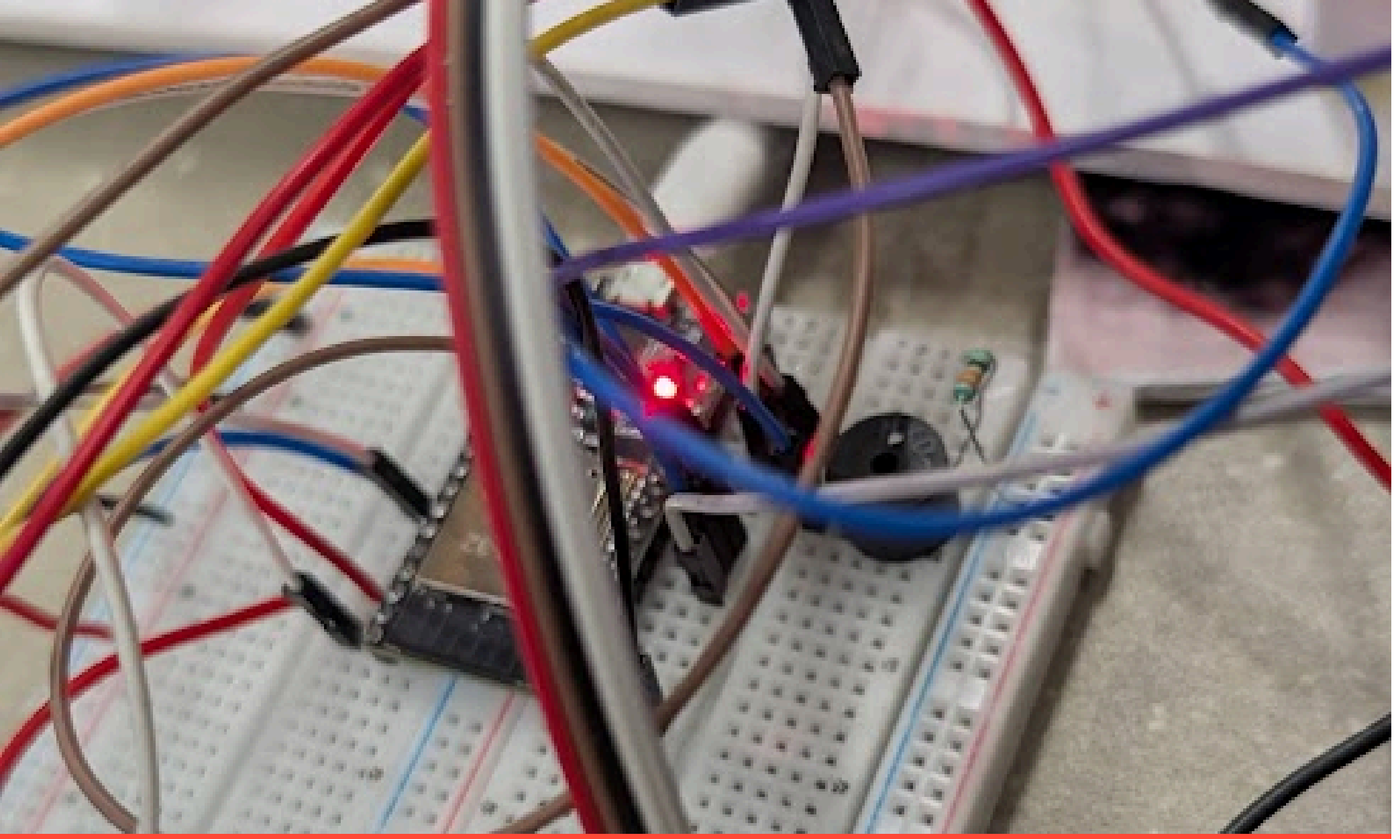
Players then manually moved their cars from position 1 through 3 based on the wheel result. The entire sequence — speed test, winner detection, wheel spin, movement — was enclosed within a while loop, allowing the process to repeat continuously.

The game resets to the speed test after each round and continues looping until one player reaches the finish line. When the winning car reaches the final position and crosses the finish line, an IR sensor detects its presence. and triggers the NeoPixel lights to display the winner's color as a final celebratory output.

LOGIC

- Conditional statements (if, elif, else) to compare reaction times and determine the round winner.
- Servo angle mapping to convert motor position into step values.
- Controlling neopixel lights to visually indicate round wins and the final winner
- A master while loop to maintain continuous gameplay until a win condition is met.
- Using break statements strategically to exit specific loops and reset the game flow when necessary

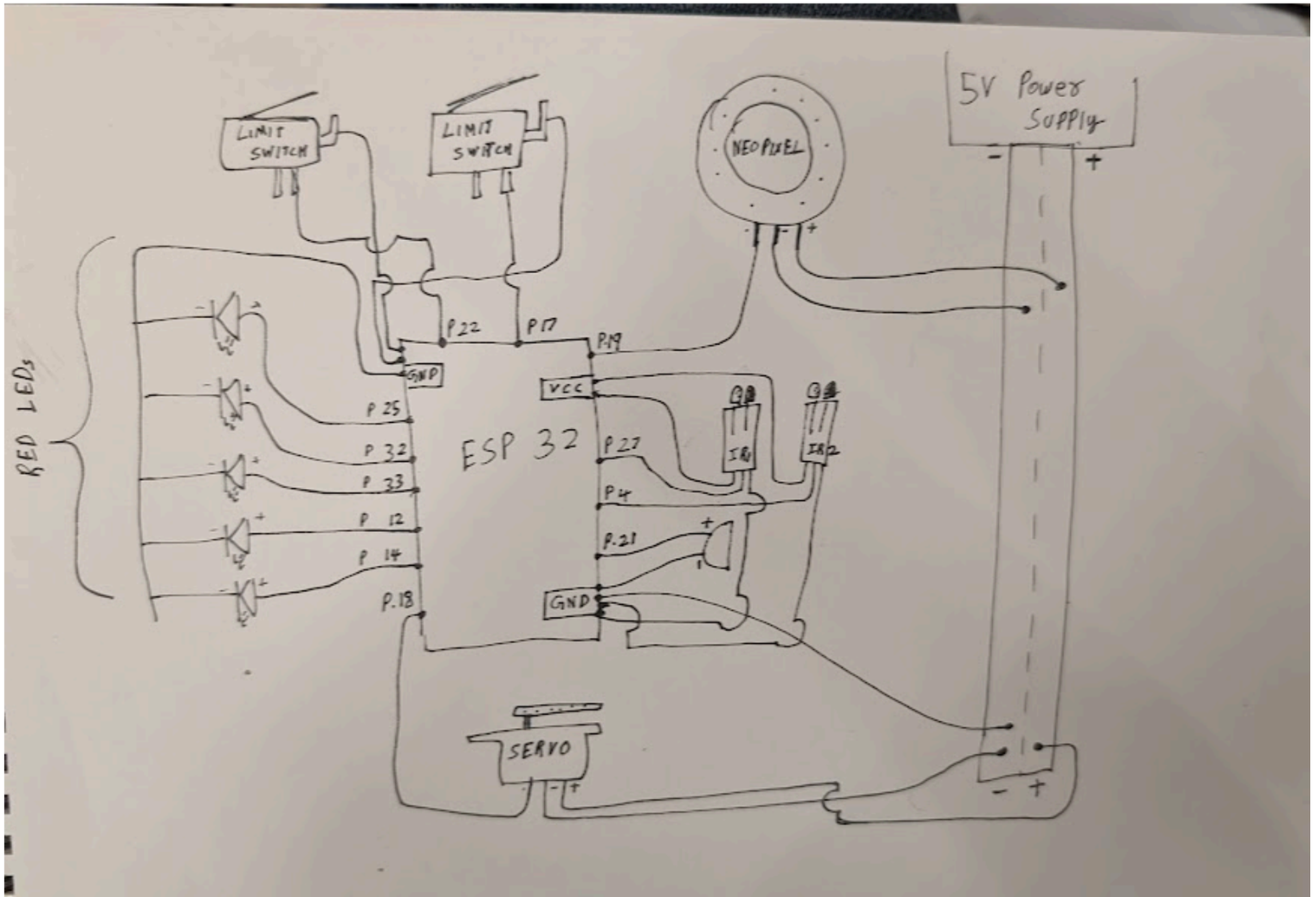
```
            break
        time.sleep(0.01)

if turn == 1:                    #if p1 wins
    print("TURN 1")
    chance = p1

elif turn == 2:                  #if p2 wins
    print("TURN 2")
    chance = p2

while p1.value() == 0 or p2.value() == 0:
    time.sleep(0.05)

while True:                     #servo starts here
    for a in anglez:
        dice.duty(a)
        time.sleep(0.4)
        if chance.value() == 0:   #to break the flow of for loop
            break
    if chance.value() == 0:     #to break the flow of while loop
            break


while True:
    val1 = ir1.value()      #reading ir sensor values
    val2 = ir2.value()

    if val1 == 0:
```



PAIN POINTS

- Synchronizing reaction inputs fairly between two players and making sure the codes don’t overlap on each other.
- Managing the power supply and GPIO connections so that components like the IR sensor, servo, and NeoPixels didn’t interfere with each other.
- Debugging wiring issues between LEDs, NeoPixels, and switches.
- Managing multiple loops withing each other was challenging, as indentation errors and breaking/resetting loops often caused unexpected behavior.
- Faulty wires/some components.

CIRCUIT DIAGRAM



CODING

BHUMIKA

- Initial importing of modules+ Red light starting system with lights going out at random time.
- If P1 is faster..... elif P2 is faster... : Determining which player goes first and displays respective player's colour on neopixel.
- Reading IR sensor values to initiate buzzer action and Neopixel flashing winning player colours.

BARATTH

- False start penalty code.
- Assigning variable "Turn": to designate turn to winning player
- Assigning variable "chance": so that other player's limit switch actions is disabled
- Servo motor angle in a list
- Breaking the flow of while loops strategically.

TEAMWORK

- Game Design and Ideation
- Circuit building
- Aesthetic visuals
- Circuit Diagram