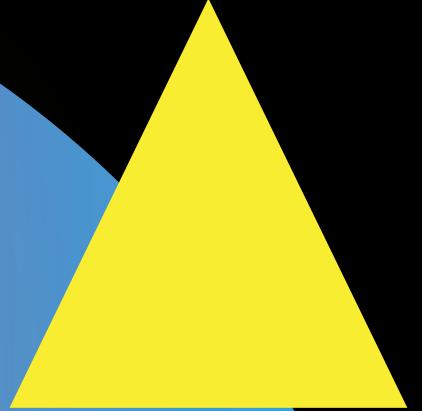


tic

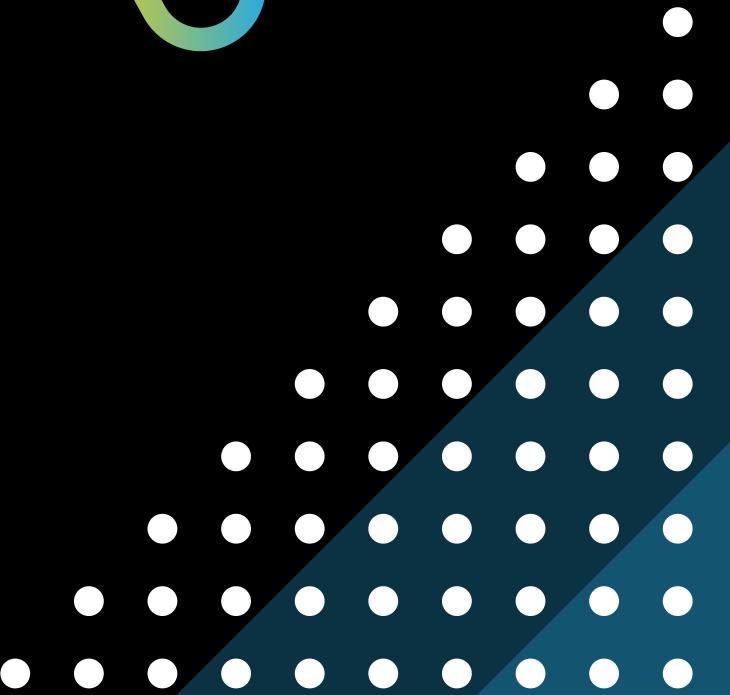
tac

toe

SHREYA AND DURVA

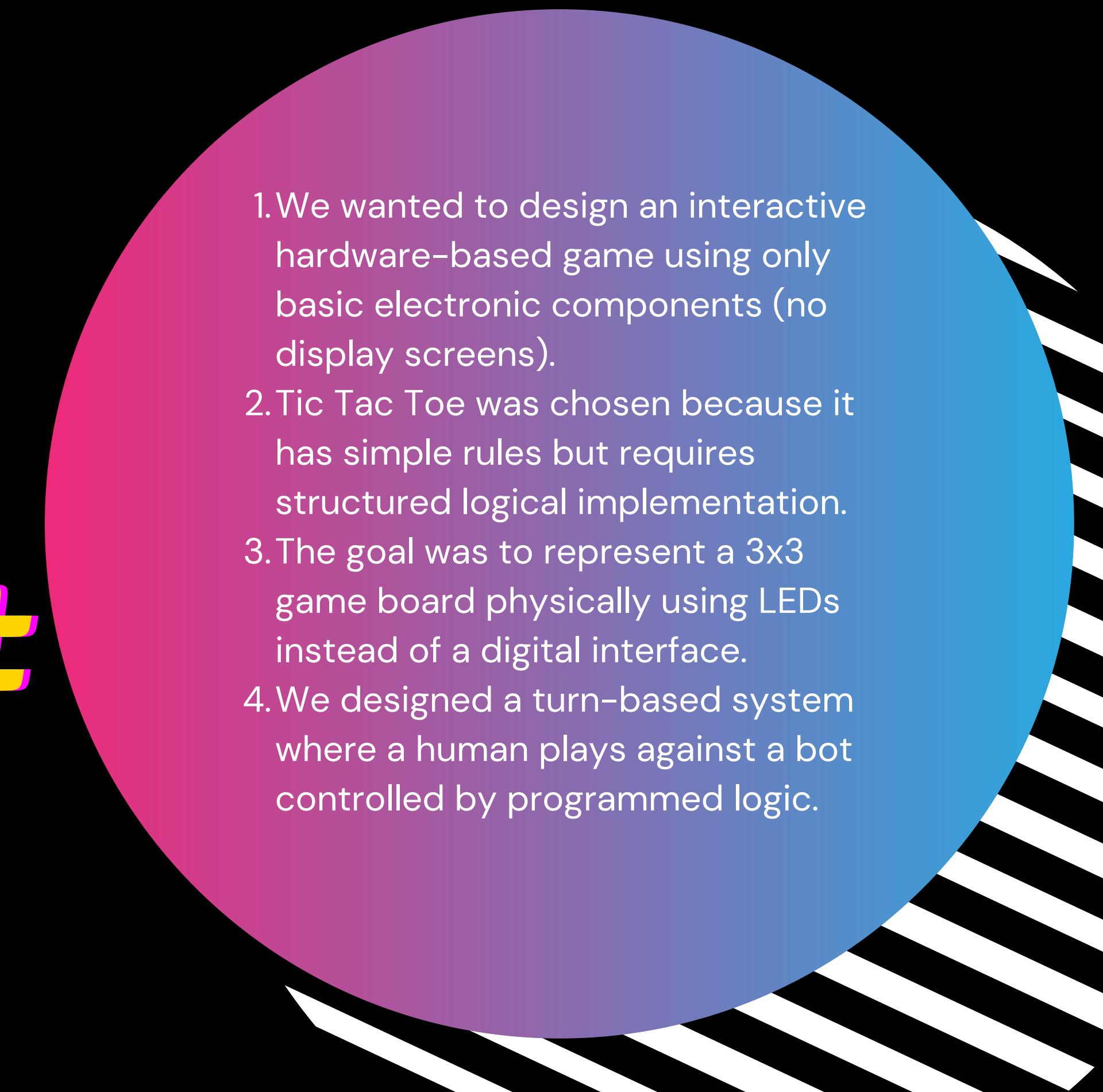


LETS **NOT** PLAY TTT





jidea development



1. We wanted to design an interactive hardware-based game using only basic electronic components (no display screens).
2. Tic Tac Toe was chosen because it has simple rules but requires structured logical implementation.
3. The goal was to represent a 3x3 game board physically using LEDs instead of a digital interface.
4. We designed a turn-based system where a human plays against a bot controlled by programmed logic.



logic

1. The game board was stored in a list (ttt) where 0 = empty, 1 = player, and 2 = bot.
2. One button cycles through the 9 positions, while another confirms the player's move.
3. The code prevents selecting already occupied boxes using conditional checks.
4. After the player move, the bot selects a random empty box and updates the board.
5. Win conditions are evaluated by checking predefined index combinations representing rows, columns, and diagonals.

Components

- 1.ESP32 Microcontroller – Used as the main processing unit to control logic and GPIO operations.
- 2.18 LEDs (9 Green + 9 Red) – Represented player and bot moves on the 3x3 board.
- 3.Push Buttons – Used for navigating between boxes and confirming selections.
- 4.NeoPixel LED Strip – Provided visual feedback for win/loss animations.
- 5.Breadboard + Power System – Used for circuit connections and stable voltage supply.

reflection

learnings

1. Learned how to design a turn-based embedded system using state-based logic.
2. Understood how to map software data structures to physical hardware outputs (LED grid).
3. Improved debugging skills by fixing infinite loops and logical errors.
4. Gained experience handling button debouncing in hardware systems.
5. Learned how to structure code efficiently for readability and scalability.

pain points

1. Making sure all connections were secure and all...and managing the huge number of jumper wires
2. Initial win animation logic caused the system to freeze due to infinite loops.
3. Managing LED updates without clearing previous states was challenging.
4. Ensuring the bot selected only empty positions required careful condition checks.
5. Resetting the board properly after a win required restructuring the control flow.

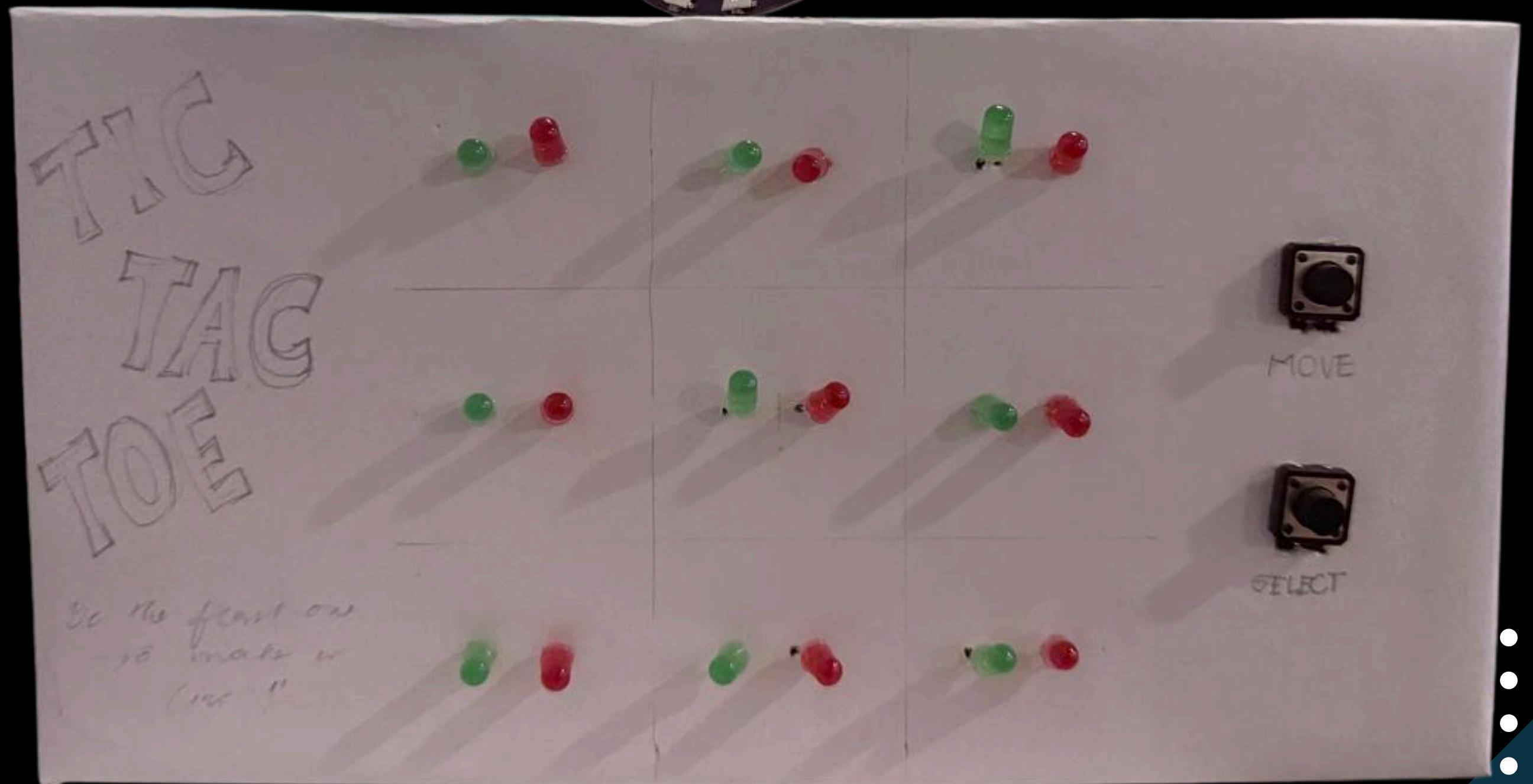
contribution

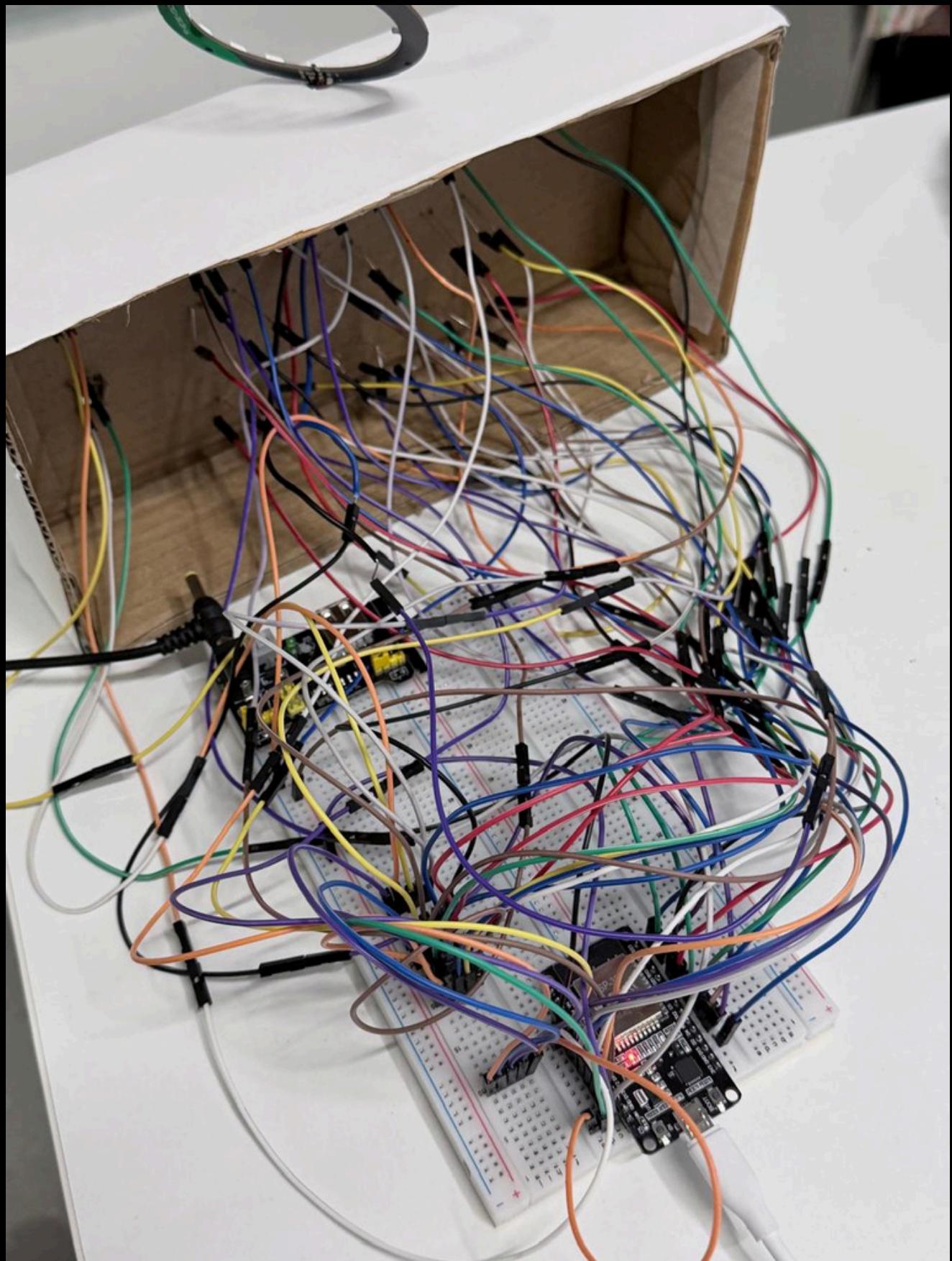
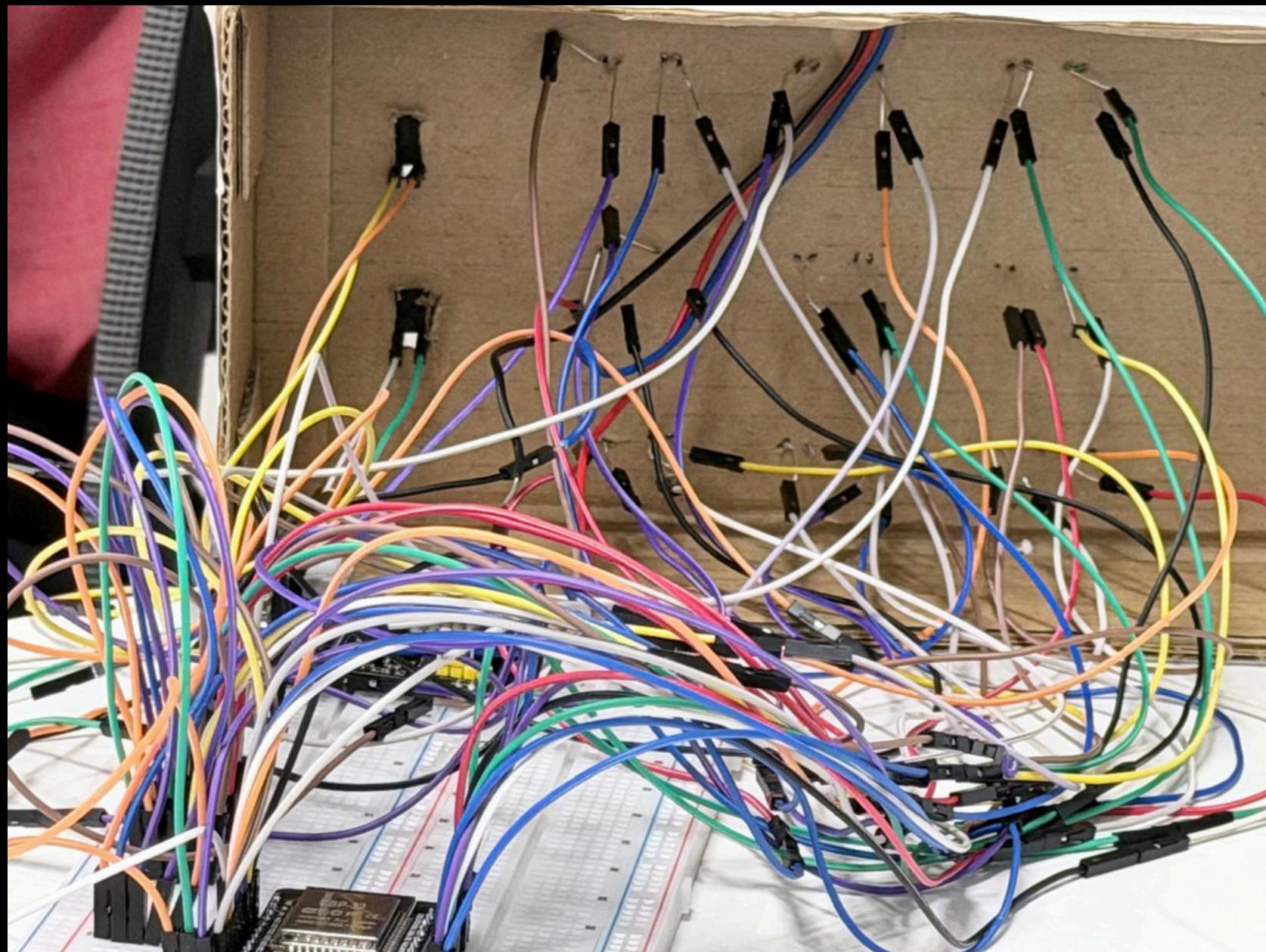
shreya

1. Ideation
2. Coding – selecting and moving to different grids, winning refined code
3. Aesthetics (building with cardboard)
4. Circuit building

durva

1. Ideation
2. Coding – assigning values and neopixels and old winning code, random.
3. Aesthetics (building with cardboard)
4. Circuit building







*thanks
for not
playing!*