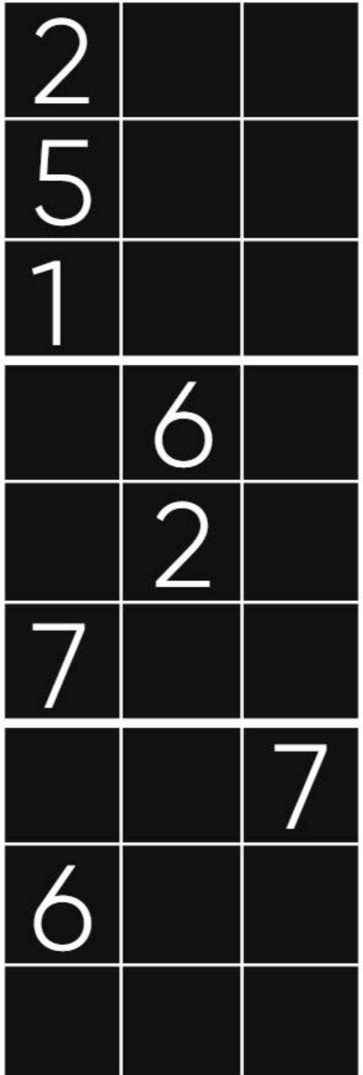
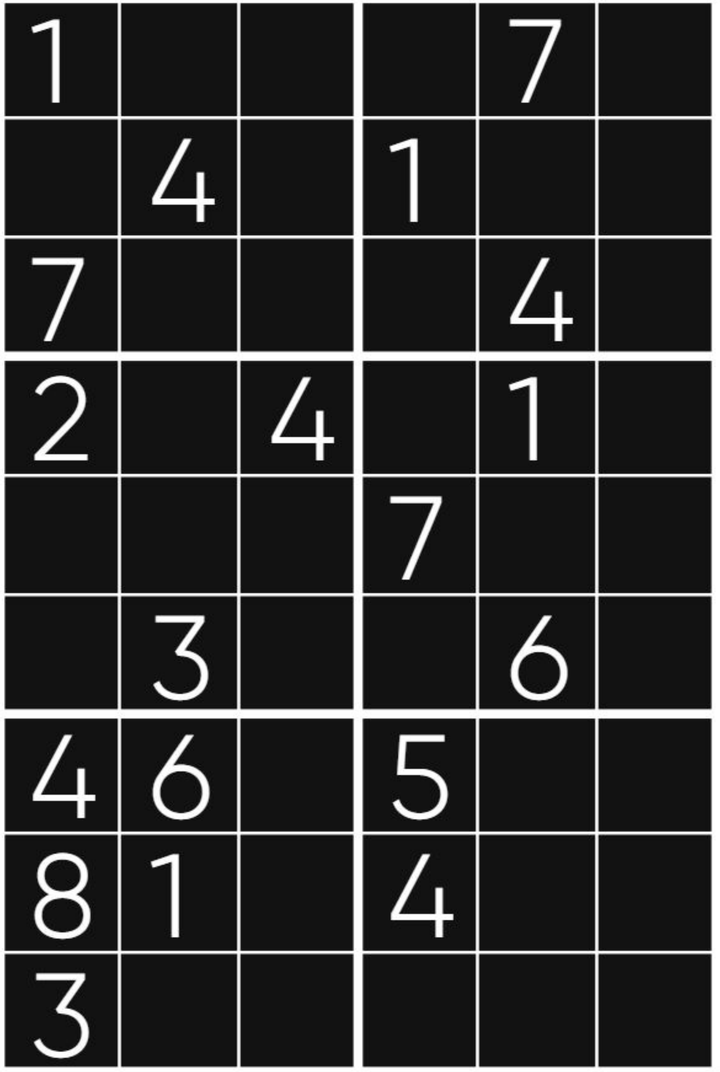


F3: 9x9

Winstone Yang
Michael Lee
Moises Gavarrete



Use Case

9x9 aims to merge the pros of paper sudoku and online solvers without any of the cons - bringing together accessibility of the former with the dynamic interactivity of the latter

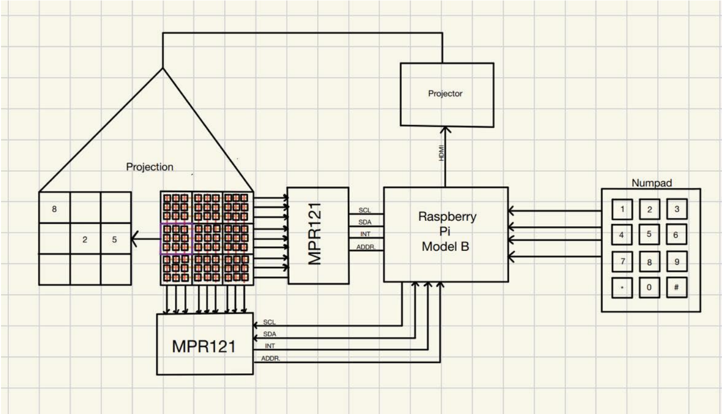
Use Case	**Solution**
Traditional Sudoku solvers passively spit out answers as opposed to helping solvers learn	Our system promotes learning with an interactive hint system and undo features that guides users through problem-solving
Online solvers and learning systems require internet access	Our self-contained, portable device works offline, making it accessible anywhere
Apps & online learners can be difficult for elderly or non-tech-savvy users	The projection-based interface mimics traditional Sudoku, making it intuitive and easy to use
Paper Sudoku lacks customization and interactivity	Our system enables real-time updates, undo functionality, and dynamic puzzle generation
Touchscreens are costly for large-scale use	We use low-cost materials like MPR121 ICs and copper foil tape for an affordable alternative

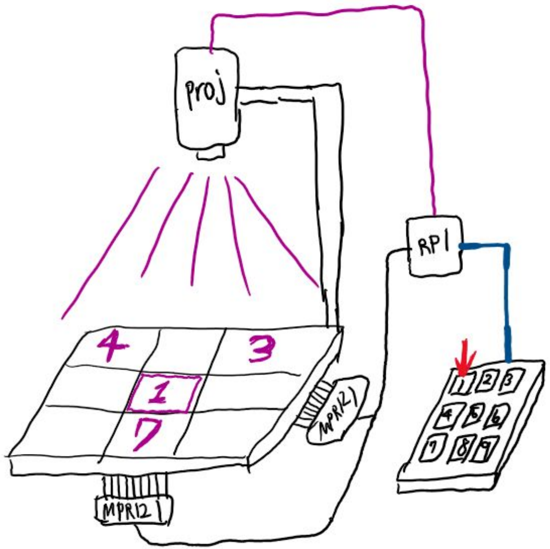
Use Case Requirements & Quantitative Design Req

Use Case Requirements	**Quantitative Metrics**	**Reasonings**
Accurate Touch Input Detection	≤1mm error margin	The 1mm error margin is based on standard capacitive touch accuracy seen in consumer electronics, preventing accidental touches
Real-Time Board Updates	≤25ms latency from touch to projection update	Mimics the responsiveness of traditional Sudoku, 25ms since it's the average of low latency touchscreen display (10-40ms)
Projection Alignment with Capacitive Grid	≤1mm deviation between projected grid and touch-sensitive areas	A 1mm deviation is small enough to maintain usability without noticeable offset caused by projection misalignment
Portability & Self-Sufficiency	Fully functional without internet; weight ≤ 1.3kg	This makes the system lightweight and easy to transport, 1.3kg is chosen as it aligns with the lower end of standard laptops
Projection Readability	Brightness of ≥ 200 lumens with adjustable contrast	200 lumens is chosen based on short-throw projectors at 1-2 meters, ensuring readability without excessive glare even in regularly lit rooms

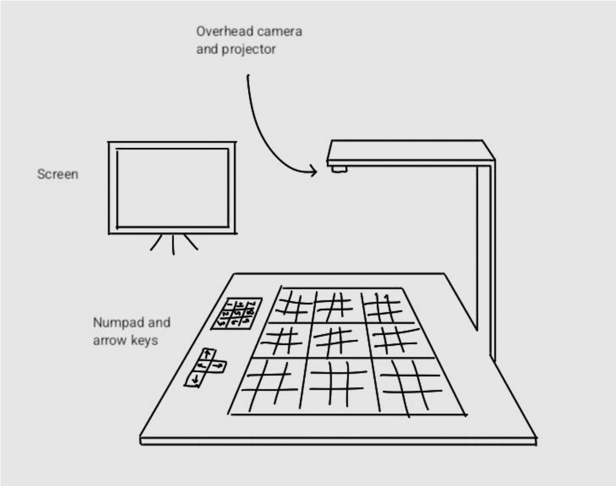
Solution Approach + Implications

- **Projection-Based Sudoku Board:** A Raspberry Pi-driven projector displays the Sudoku grid onto a surface
- **Capacitive Touch Sensing:** Users directly select cells instead of navigating with buttons
- **Real-Time Updates & Undo Feature:** The projected board dynamically updates within 10ms latency, ensuring smooth interaction
- **Standalone & Offline Functionality:** Unlike online Sudoku apps, our system is fully self-contained, does not require internet, and is portable for classrooms, libraries, and recreational use.
- **Cost-Effective Alternative to Touchscreens:** By using copper wiring and capacitive sensors, we eliminate the need for expensive touchscreens, making the system more affordable and scalable.

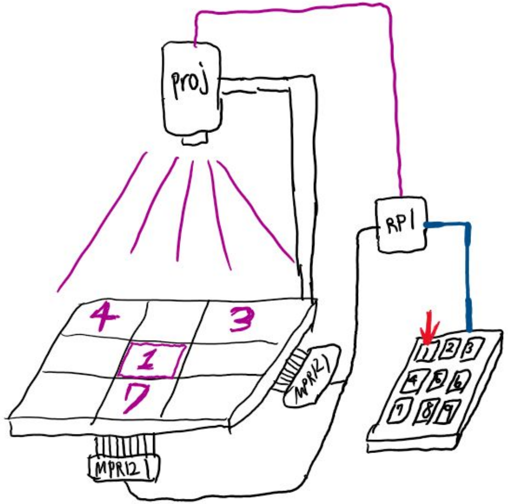




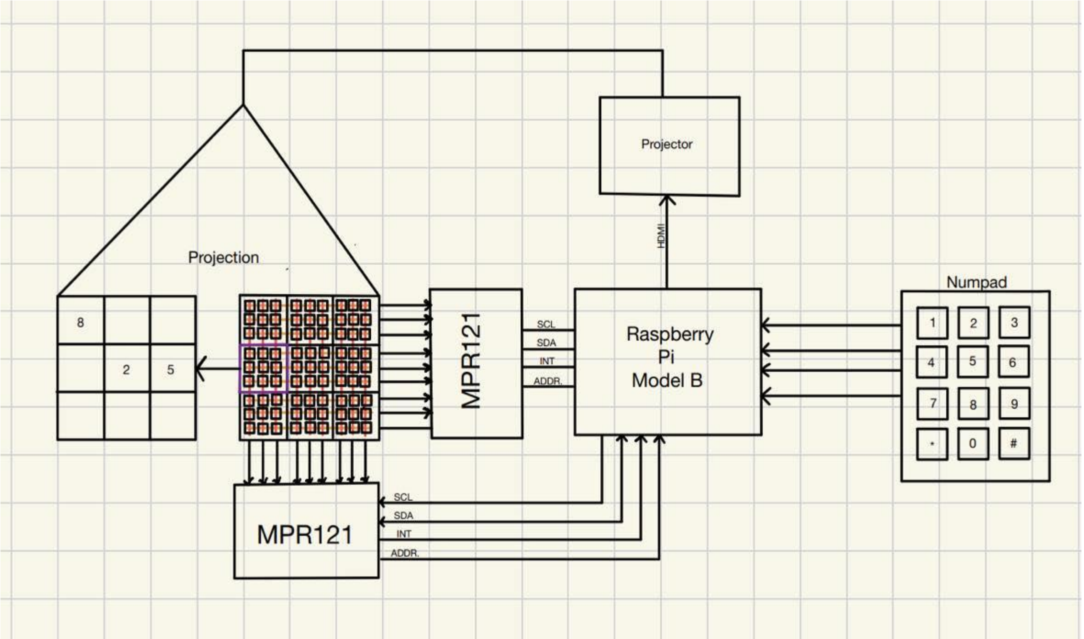
Solution Approach - Changes Made

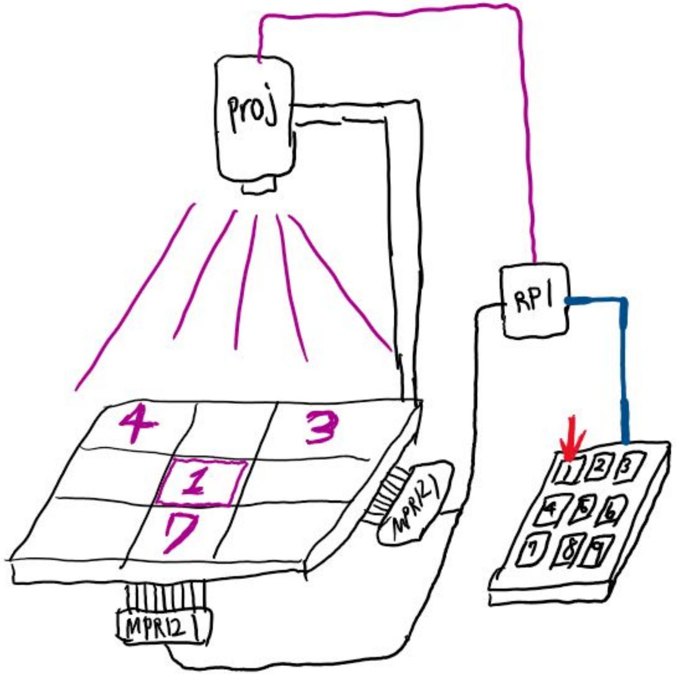


1. **Removed OpenCV and Camera-Based Detection**
   a. System originally used computer vision (CV) to scan and recognize a Sudoku board from paper to image
   b. Replaced with loading a predefined Sudoku file, eliminating the need for camera-based board detection and reducing software complexity
2. **Replaced Button Navigation with Capacitive Touch Sensing**
   a. Initial design required arrow key navigation to move through 81 cells
   b. We introduced capacitive touch sensing allowing users to directly select cells
3. **Projected Board Onto Copper Wire Grid for Seamless Interaction**
   a. The board is now projected directly onto a surface with embedded copper wiring, integrating both display and input in the same physical space
   b. Allows for natural touch-based interaction similar to traditional Sudoku but with real-time digital updates
4. **Removed FPGA from the Plan**
   a. Decided to streamline development by relying solely on a Raspberry Pi, which offers enough processing power for real-time updates

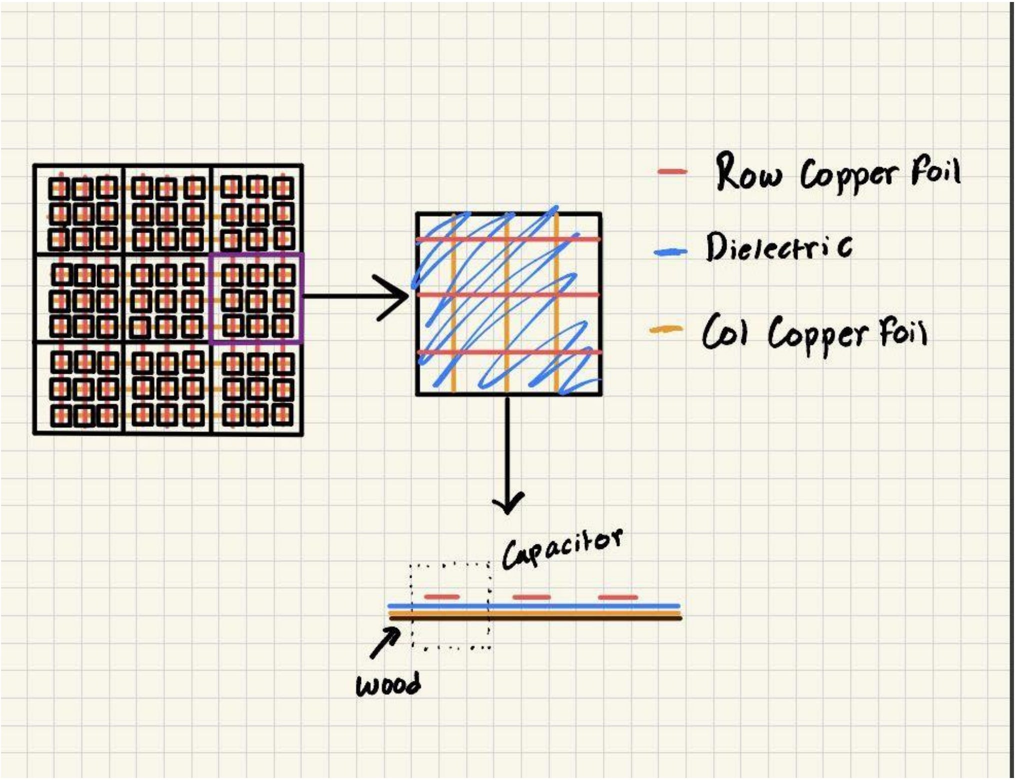


Block Diagram for Overall Design





Block Diagram - Copper Wiring



Grid-Based Touch Sensing: The 9×9 grid consists of copper foil strips separated by a dielectric layer, creating 81 capacitive touch points

Capacitance Detection Mechanism: When a user touches an intersection, their finger alters the capacitance, which is detected and processed by the MPR121 IC

Direct Cell Selection: This system eliminates button navigation, allowing users to tap directly on the projected board for a more intuitive and seamless Sudoku experience

Implementation Plan (buy vs. download/reuse)

Component	Develop	Buy	Download/Reuse
Capacitive Touch Grid Processing	Custom firmware & input handling for MPR121	MPR121 ICs, copper tape, and wiring	N/A
Projection System	Software to align projection with touch grid	Vankyo Burger 101 projector	Pygame for rendering
Sudoku Board Logic & State Management	Custom game logic, undo feature, and hint system	N/A	N/A
Processing Unit	Custom GPIO handling & sensor input mapping	Raspberry Pi 4 Model B	Raspberry Pi GPIO & sensor libraries
User Interface Rendering	Dynamic board updates & interactive hint system	N/A	Pygame (for UI and real-time updates)

Testing, Verification, Metrics + Plan for Failure

Use case requirement + metric	Testing method	Potential failure	Impact	Mitigation strategy
Accurate touch input detection (<= 1mm error margin)	Test with 100+ touch inputs, measure detected touch vs. intended position using a calibration grid	Touch input has high error rates (>1mm deviation)	Users struggle to select the correct cell	Recalibrate capacitive sensor hardware, change sensitivity of the capacitive grid
Real time board updates (<= 25ms latency from touch to projection update)	Run timed experiment to measure delay from touch input event to visible projection update	Board updates exceed 25ms latency, or update is inconsistent/ happens non-chronologically	Gameplay is ruined or slow and unresponsive	Reduce unnecessary computations, optimize the code, switch library from Pygame if needed

Projection Alignment with Capacitive Grid (<= 1mm deviation)	Compare registered touch coordinates with projected board position and calculate deviation	Projection misalignment (> 1mm deviation)	Users tap the wrong cells due to misalignment, disrupting gameplay	Change projector positioning, or even implement correction algorithms in software for alignment
Portability & Self-Sufficiency (Fully functional offline, weight <= 1.3kg)	Weigh the assembled device	Device exceeds 1.3kg weight	Reduced portability, making it harder to carry or mount in different locations	Optimize hardware layout, use lighter casing materials, remove unnecessary components
Projection Readability (>= 200 lumens with adjustable contrast)	Test projection visibility in different lighting conditions (dim, normal indoor lighting, and bright environments)	Projected board is unreadable in bright rooms	Users struggle to see the board in well-lit environments	Increase projector brightness, adjust contrast dynamically based on light conditions, explore alternative projection surfaces for better visibility

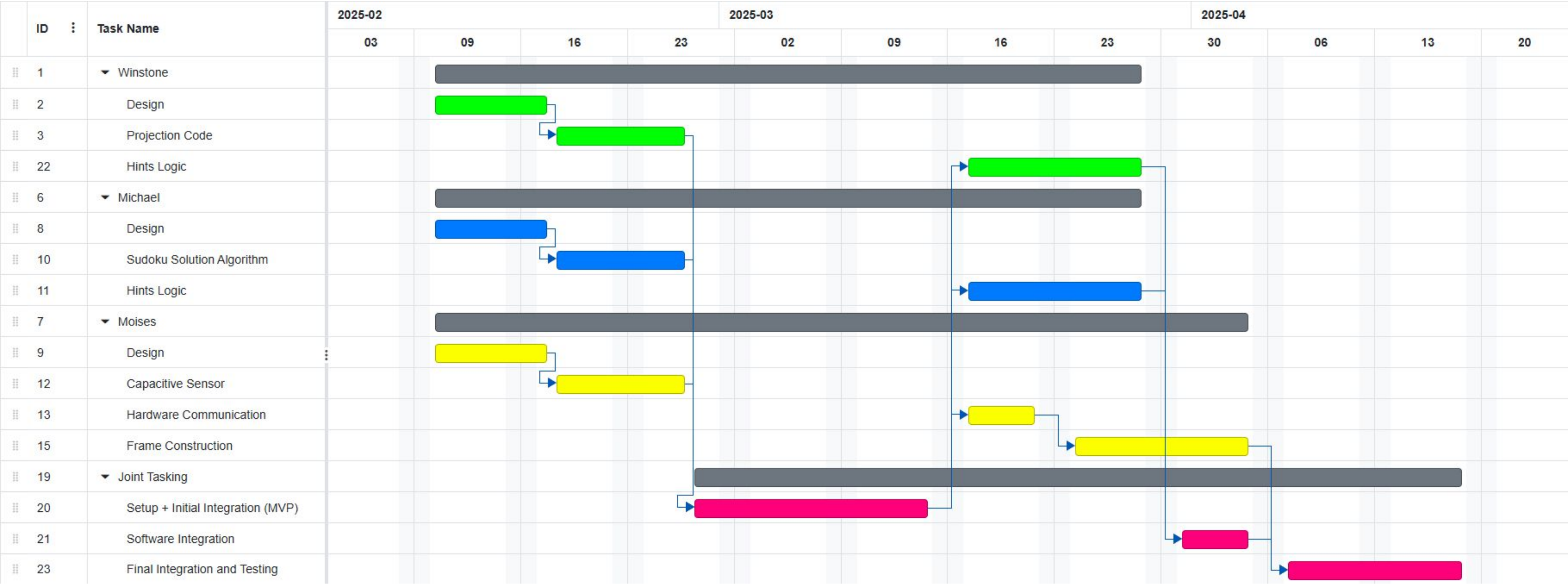
Tasking and Division of Labor + Minimum Viable Product

Moises	Winstone	Michael
Raspberry Pi setup along with other hardware components and signals	**Project rendering of board + real-time updates** based on user moves	**Game logic and system integration**, ensuring robustness of software
Copper capacitive touch sensor system - building and testing	**Alignment of projection** onto grid along with readability and contrast testing	**Hints and backtracking system** later on beyond the MVP phase

Minimum viable product:

- Basic sudoku board projection
- Touch input detection
- Real time updates of the board and onto the projecting surface
- Projection-touch alignment

Gantt Chart


ID
Task Name
2025-02
2025-03
2025-04
03
09
16
23
02
09
16
23
30
06
13
20
1 Winstone
2 Design
3 Projection Code
22 Hints Logic
6 Michael
8 Design
10 Sudoku Solution Algorithm
11 Hints Logic
7 Moises
9 Design
12 Capacitive Sensor
13 Hardware Communication
15 Frame Construction
19 Joint Tasking
20 Setup + Initial Integration (MVP)
21 Software Integration
23 Final Integration and Testing