

AI Table Tennis Referee System

Project Plan — FabLab Collaboration Proposal

Project Lead	Kareem (Software & Web) — 42 Wolfsburg Student
Hardware Partner	To be confirmed — CV & hardware integration
Timeline	8 weeks (Phase 1–3), ongoing refinement
Budget Request	~€215 hardware (Raspberry Pi 5 + Camera + Lens + Accessories)
Deliverable	Working prototype: camera-based referee that tracks ball, detects bounces, and scores points

1. What We're Building

An AI-powered table tennis referee that uses a single camera mounted above the table to track the ball in real-time, detect bounces, net hits, and out-of-bounds events, and automatically keep score according to official ITTF rules. The system runs on a Raspberry Pi 5 with a global shutter camera — compact, affordable, and self-contained.

This combines computer vision (OpenCV), real-time processing, and embedded systems — core FabLab competencies — into a tangible product that can be demonstrated, iterated on, and potentially commercialized.

2. Why This Project Will Be Completed

I understand the concern about unfinished projects. Here's how this one is different:

- **Software-first approach:** The web simulation (already in development) proves the algorithms work before any hardware is purchased. No wasted hardware if the concept fails.
- **Incremental milestones:** Each phase produces a working, demonstrable result — not just code that 'will work eventually.'
- **Personal investment:** This is central to my 42 School portfolio and entrepreneurial roadmap. I have direct incentive to ship it.
- **Small hardware scope:** Only 3 components needed (Pi, camera, lens). No complex multi-part builds. Setup takes one afternoon.
- **Clear success metric:** Can the system correctly score a rally? Yes or no. No ambiguity about 'done.'

3. Team & Responsibilities

Role	Person	Responsibilities
Software Lead	Kareem	Web simulation, scoring logic, UI/dashboard, deployment, project management
Hardware / CV	Partner (TBC)	Camera setup, OpenCV pipeline, Raspberry Pi optimization, physical mounting
Advisor / Sponsor	FabLab	Hardware funding, workspace, guidance, potential showcase platform

4. Project Timeline & Milestones

Phase	Duration	Deliverable	Status
0: Simulation	Week 1	Web-based ball physics sim with scoring logic + camera FPS comparison. Proves detection algorithms work in software.	In Progress ✓

1: Hardware Setup	Week 2–3	Raspberry Pi 5 + Arducam OV9281 mounted above table. Camera calibrated, capturing frames. Basic ball detection with OpenCV.	Pending funding
2: Detection MVP	Week 4–5	Real-time ball tracking at 120+ FPS. Bounce detection on table surface. Net hit detection. System scores a simple rally correctly.	—
3: Full Referee	Week 6–8	Complete scoring (11-point, serve rotation). Edge case handling. Dashboard UI on screen/tablet. Live demo-ready.	—
4: Polish	Ongoing	Spin detection, multi-camera support, player stats tracking, mobile app.	Stretch goals

5. Hardware Budget

Component	Model	Why This Choice	Cost
Single-board computer	Raspberry Pi 5 (8GB)	Sufficient GPU for real-time OpenCV at 120+ FPS. Large community, proven for CV projects.	~€95
Camera	Arducam OV9281 Global Shutter	Global shutter eliminates motion blur on fast balls. 120 FPS at 1280x800. Monochrome = better contrast.	~€75
Lens	6mm f/1.4 CS-mount	Wide enough to cover full table from ~2m height. Fast aperture for indoor lighting.	~€25
Accessories	SD card, mount, cables, case	Standard components. Mount can be 3D printed in FabLab.	~€20
Total			~€215

Note on Pi pricing: Yes, prices have increased. The Pi 5 8GB is currently ~€95. Alternatives (e.g. Orange Pi 5) were considered but lack the camera ecosystem and community support critical for a CV project. The Pi 5's dedicated camera connector and optimized libcamera stack make it the lowest-risk choice.

6. Commitment & Accountability

To ensure this project reaches completion, I propose the following accountability structure:

- Bi-weekly progress demos:** Every two weeks, I'll show a working demonstration of the current state — not slides, but live software or hardware running.
- Phase-gated hardware:** Phase 0 (simulation) is completed using zero FabLab resources. Hardware is only ordered once the simulation proves the concept.
- Open documentation:** All code on GitHub, all decisions documented. If I can't continue, someone else can pick it up.
- Defined 'done':** The project is complete when the system can correctly score a full 11-point game in a live demo at the FabLab.