

Guided Chinese Chess

Charlie Ai, Claire Lee, Yoyo Zhong

Chessboard that teaches you chinese chess fundamentals

Real-time LED move guidance on physical board

LLM-driven move explanations

CV board state detection

Adaptive guidance modes to accommodate players with different skill levels:

Full guidance -> Assistive -> Challenge

Cultural preservation: Cross cultural exchange

- Accessibility: Low cost (~\$200) enables broad access
- Educational: Lowers barrier to Xiangqi



Use Case & Design Requirement - Group

Use Case Requirement	Design requirement	Justification
Board state detection feels instantaneous	CV board detection latency $\leq 500\text{ms}$; piece recognition accuracy $\geq 95\%$	Users perceive delays $> 1\text{s}$ as lag; Cannot guarantee 100% accuracy with CV based board state detection
Move validation feels instant	Require a stable board state for $\geq 200\text{ms}$ Fixed overhead camera	Below human reaction time ($\sim 200\text{ms}$); appears instantaneous
Legal moves are clearly visible on board	90 individually addressable SK6812 LEDs; distinct colors	Physical board feedback minimizes screen reliance; color coding is intuitive
AI suggestions are always legal	Engine bounded LLM prompting; LLM restricted to explain engine evaluations and candidate moves only	Illegal suggestions destroy user trust; validator guarantees correctness

Use Case & Design Requirement - Group

Use Case Requirement	Design requirement	Justification
System warns before blunders	Blunder detection triggers LED warning $\leq 500\text{ms}$ of piece lift	Proactive warning to encourage take back and help beginner players avoid easy to spot blunders
Guidance adapts to skill level	Three modes: Full Guidance (auto LEDs + explanations), Assistive (on-request), Challenge (minimal)	Supports learning progression; prevents information overload for advancing players
AI explanations are helpful and timely	LLM explanation latency $\leq 2.5\text{s}$; fine-tuned Qwen2-7B with RAG for Xiangqi knowledge	Acceptable wait for quality content; domain-specific training prevents hallucination
Players practice tactics	Puzzle mode activates when tactical pattern detected	Active recall more effective than passive hints

Board Layout and manufacturing

Chess Pieces

Laser cut frosted acrylic, paint engraved symbol and letters
(instead of 3D printing)

- Allow LED light to be seen
- Clearer details with small letters
- Less time to manufacture

Why Laser-Cut Acrylic?

- Precise hole placement for LED alignment
- Translucent material allows light diffusion
- Durable, professional appearance
- Available at CMU maker spaces

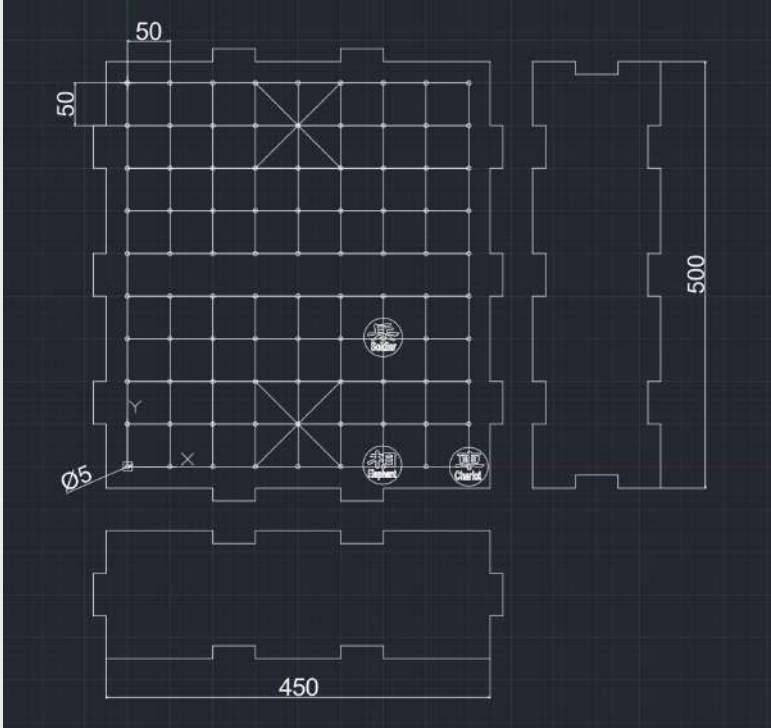
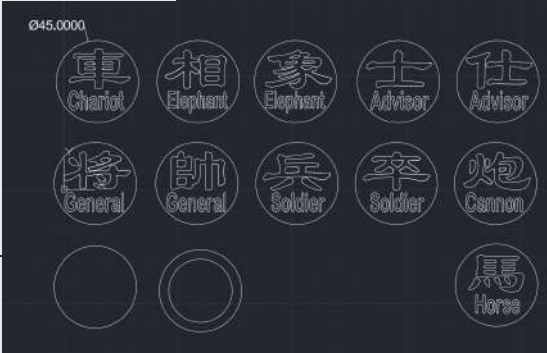
Chess Board

Laser cut plywood, holes on each intersection of grid for led
(instead of frosted acrylic)

- Cover up raspberry pi board and wires
- Less disturbance to computer vision
- Cheaper

Manufacturing Plan

Component	Method	Status
Board surface	Laser cutting + engraving	Design complete
LED holes	5mm drill at intersections	Template ready
Housing	3D printed frame	CAD in progress
Piece labels	Existing traditional pieces	Acquired



Board State Detection, Trade off - Yoyo

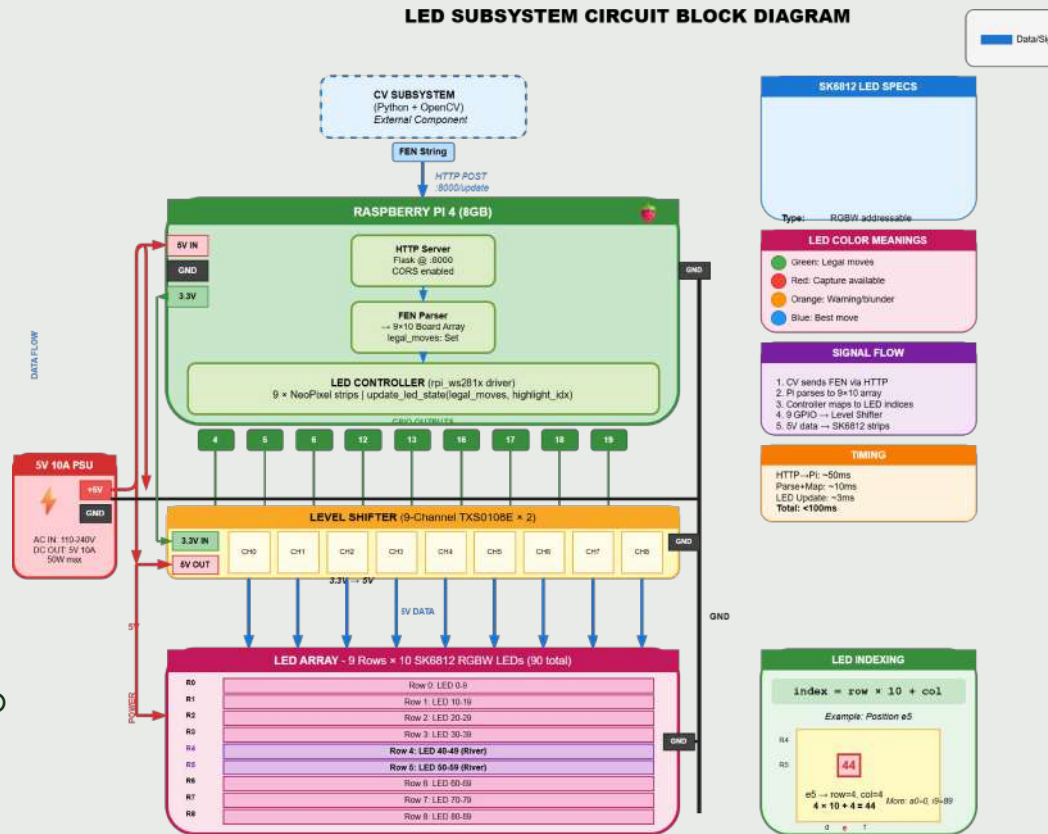
- Overhead camera + OpenCV preprocessing (board alignment via ARUCO + homography)
- YOLOX object detection to identify Xiangqi pieces and positions
- Reconstruct full board state and convert to FEN representation
- Handles dataset training + occlusion challenges for robust real-world detection

Computer Vision (YOLOX)	Capacitive Touch Grid	Hall Effect Sensor and Magnets
Fixed camera position	Dense wiring and calibration required	90+ sensors required
Button-triggered stable frame capture	Cannot distinguish piece types	Custom PCB fabrication
Scalable model improvement	Sensitive to hand interference	Magnets required in each piece
YOLOX: Detects and classifies multiple pieces in one frame Strong performance on small object detection	Higher system complexity	High cost and difficult scaling

Circuit block diagram - Claire

components: Raspberry Pi 4 8 GB, 9 rows of 10 SK6812 LEDs, 5V power supply

- RPi received FEN string from CV subsystem
 - sent over HTTP
- Parses FEN into 9 x 10 internal board representation
 - each LED will be indexed with a single digit, we will just have to keep in mind that it translates to a 2D board
- 9 independent GPI data outputs
 - one data_in per row
- each row of LEDs will receive a connection to the power supply, ground, and data_in
 - will need to level shift the data



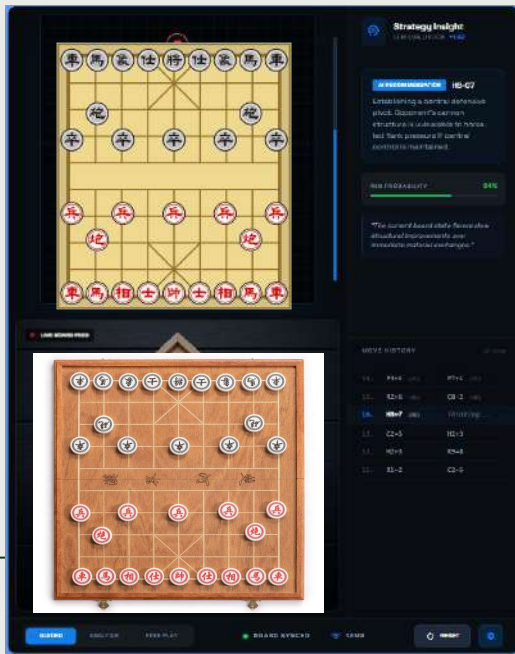
Chess Engine, Move Generation

Chess Engine implemented in Rust with 1D array for cache efficiency.

State tracking, Move validation, Search-based move generation

Move Generation:

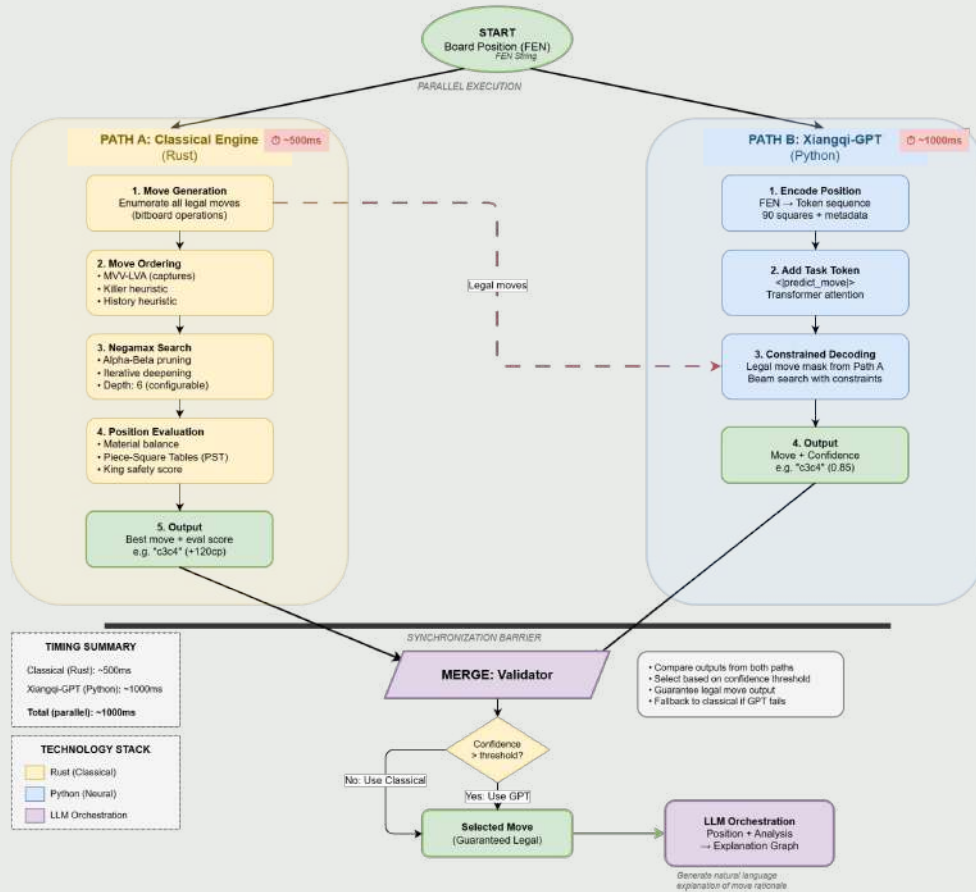
- Baseline - Negamax search with alpha-beta pruning (depth 6)
- Move order - MVV-LVA + killer + history heuristic
- Evaluating based on material, piece square tables, king safety, mobility
- ML Extension - Xiangqi GPT with legal move mask and FEN encoded token sequence
- LoRA fine-tuned Qwen2-7B to output move and confidence score



Why Qwen2-7B:

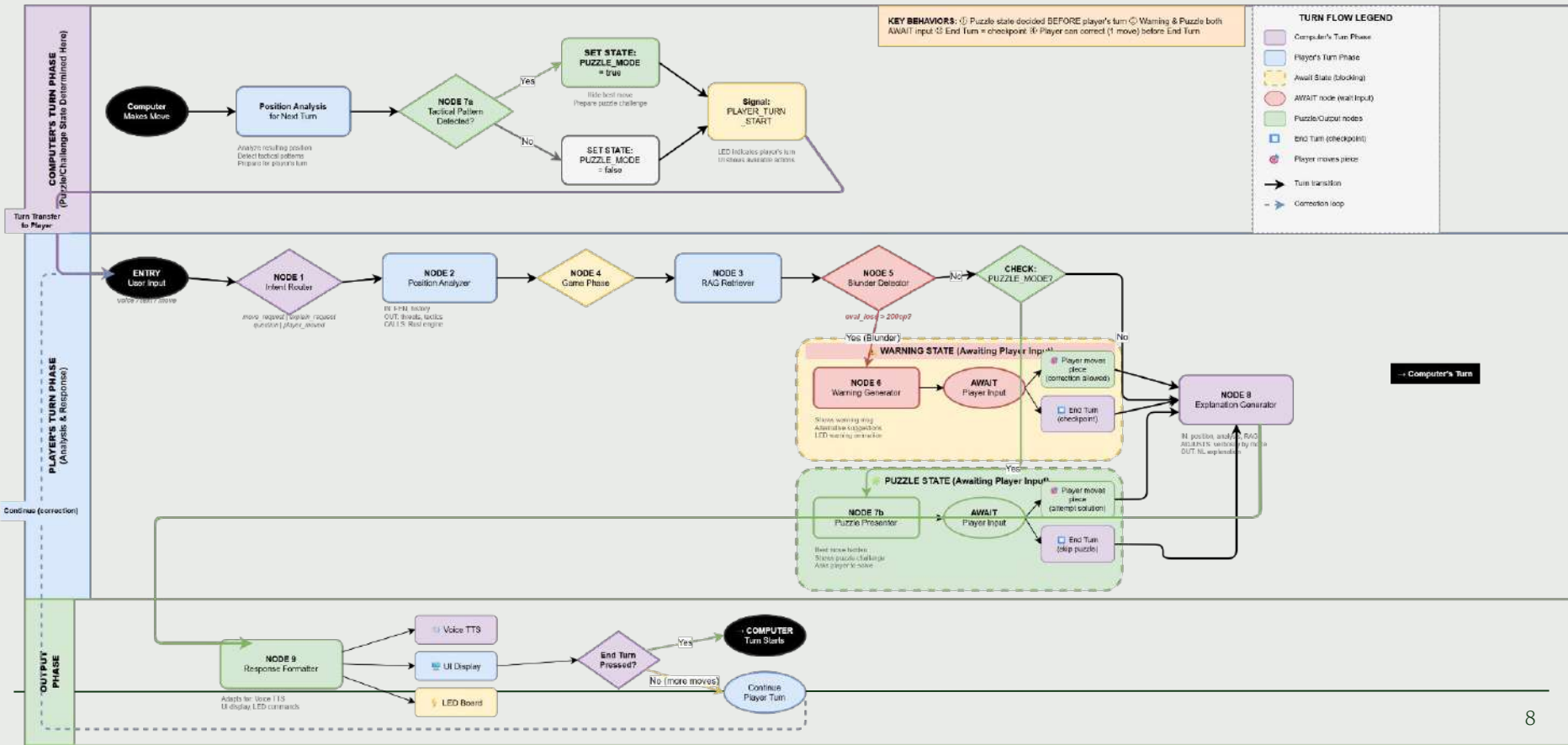
Native Chinese tokenization (炮, 馬, 車 as single tokens)
3x fewer tokens than Llama for Chinese text
Apache 2.0 license (allows fine-tuning + deployment)
Strong reasoning for board understanding

HYBRID CHESS ENGINE ARCHITECTURE Classical + Neural Network Ensemble



LLM Agent Orchestration - Charlie

LLM ORCHESTRATION STATE MACHINE Xiangqi AI Assistant - Turn-Based Flow with Await States

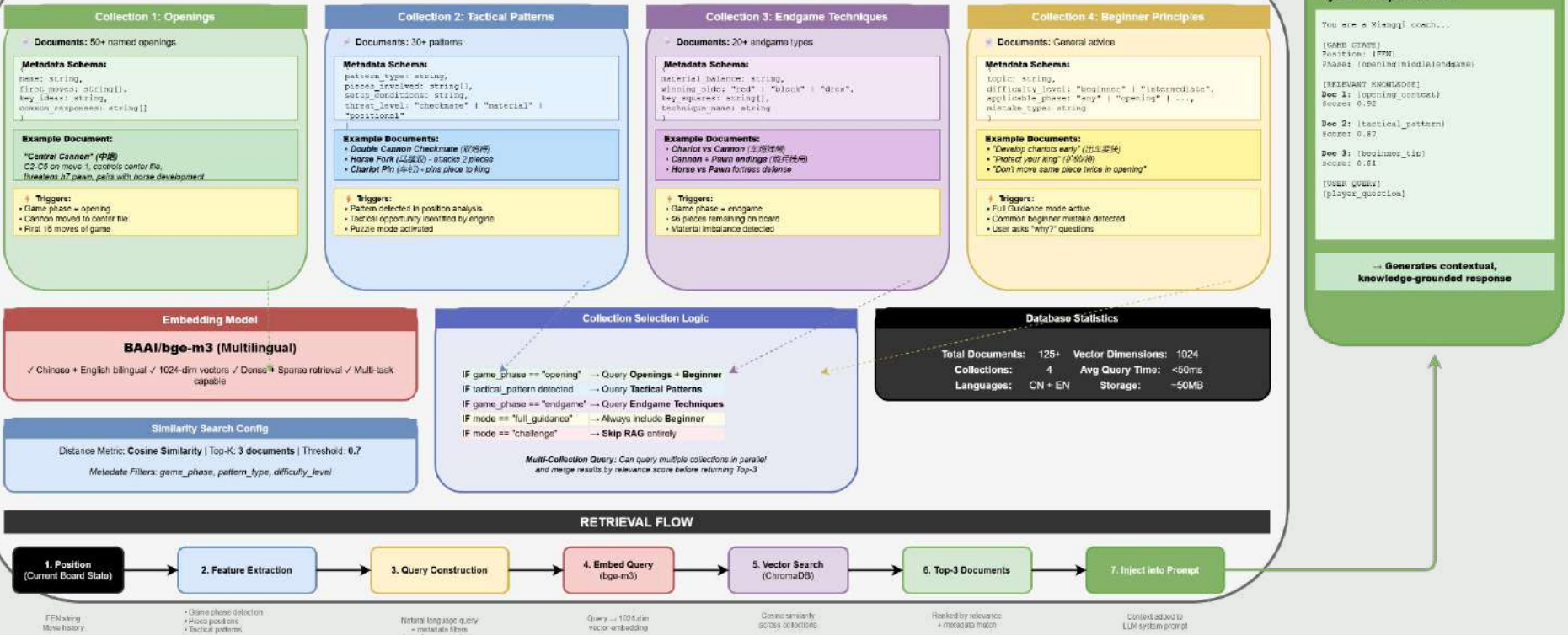


RAG - Charlie

RAG VECTOR DATABASE STRUCTURE

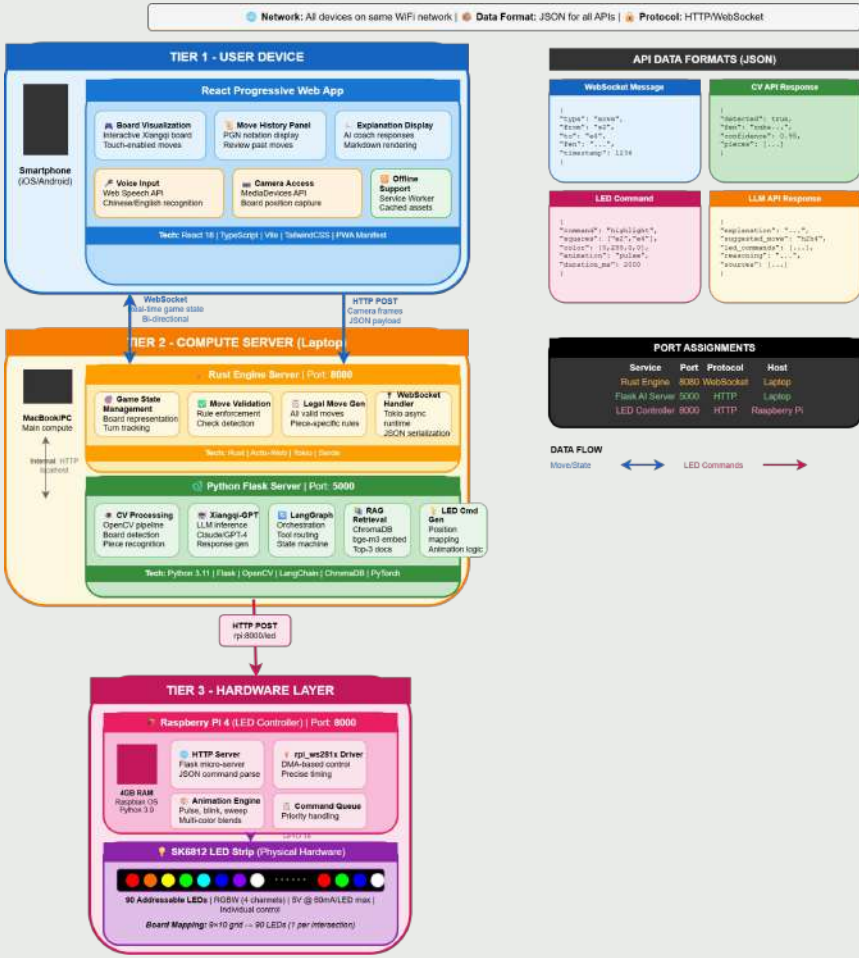
ChromaDB - Xiangqi Knowledge Base

ChromaDB - Vector Database



Deployment Diagram

- | Component | Type | Source |
- | ----- | ---- | ----- |
- | Rust Xiangqi engine | Implement | Custom Engine|
- | Xiangqi-GPT model | Adapt | LoRA fine-tune Qwen2-7B |
- | LangGraph orchestration | Adapt | Custom state machine |
- | RAG knowledge base | Curate | Compile from Xiangqi sources |
- | CV pipeline | Adapt | YOLOX & ARUCO
- | React frontend | Assemble | Standard React + custom UI |
- | LED driver code | Adapt | rpi_ws281x + custom mapping |
- | Board and pieces | Manufacturing | laser cutting |
- | Microcontroller| Rent | Pi, power supply |
- | LED strips | Buy |



Testing and verification - Claire

Latency

- Time between player completes their move and opponents move must be $<5s$
Metric: Average latency across all computer generated moves across 10 full games

- LEDs light up $<100ms$ after player completes their move

Metric: Print the timestamp of FEN string input and output of LED display and subtract the difference

Accuracy

- Board State Detection: $> 95\%$ correct detection rate across 100 trials.

Metric: Set up board with 100 different states with different numbers of pieces

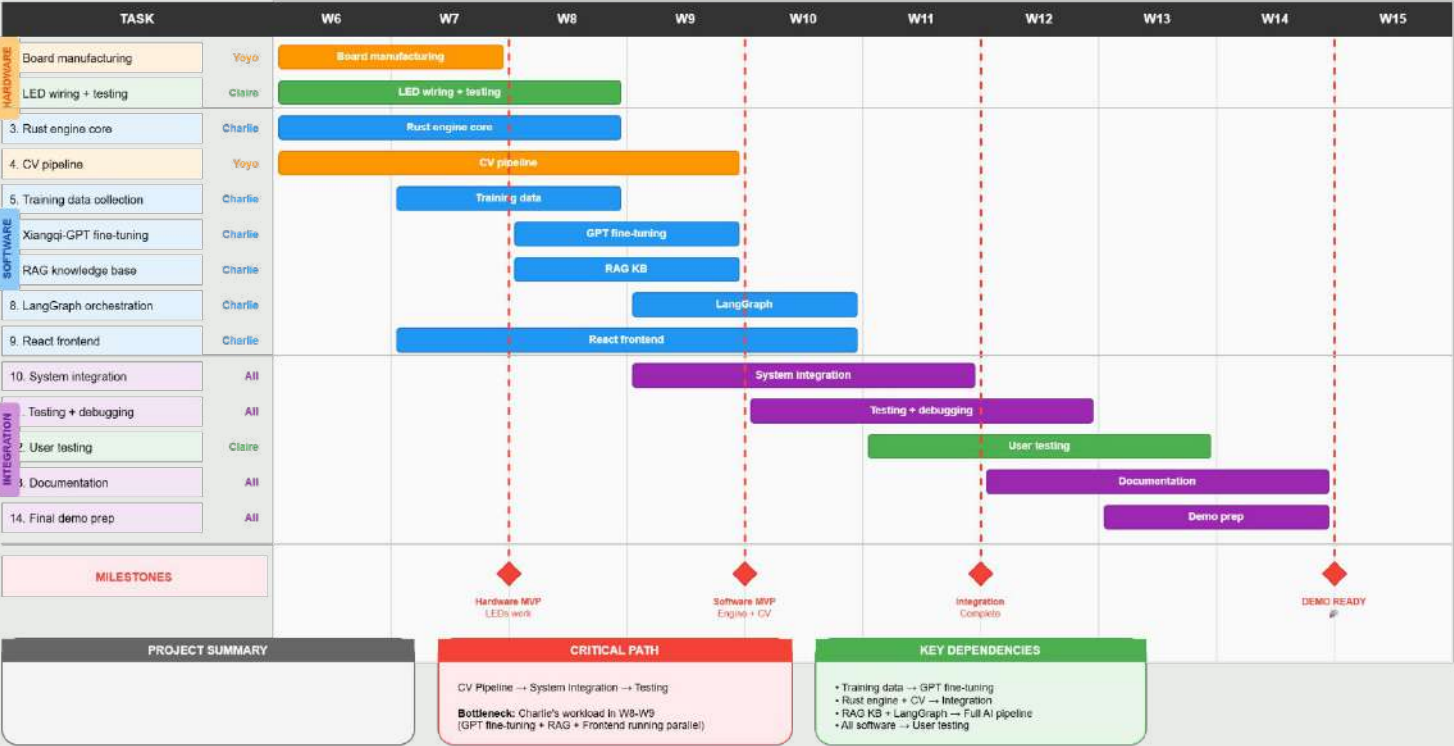
- Move generation: the opponent's generated moves should have $>98\%$ accuracy of being legal

Metric: Across 10 full games, log the percentage of correctly generated moves

UI

- Verify correct board rendering and validation of moves
 - Recruit 3-5 beginners with the task of finding the best move through the UI
 - keep note of any confusion of what certain information/button mean
 - if they are looking of specific information but we don't include it
 - time to successfully complete the task
-

Division of Labor, Tasks - Charlie



Duration: 9 weeks (W6-W14) Tasks: 14
Milestones: 4 Team size: 3

Charlie: 7 tasks Claire: 2 tasks
Yoyo: 2 tasks Ali: 5 tasks

Guided Lighting Configuration - Claire

starting sequence: Xiangqi board layout - highlight the palace area, then river area, then player side, then opponent side

ending sequence: the winner's side of the board is flash for 10s in all white

LED key:

color	meaning
yellow	legal move
green	optimal move
orange	capture move
red	invalid move
purple	computer generated opponent's move
blue	starting and ending sequence