**Backgammon Project — Session Summary**

Backgammon Project — Session Summary  
Date: 2025-09-25 20:32

1) Project Overview  
- Goal: A closed backgammon platform with a server that manages users, matches, rules, dice, cube and move validation; and clients that render the board and allow interactive, sub-move streaming (step-by-step) as in a physical game.  
- Transport & Protocol: gRPC + Protocol Buffers (Protobuf). Server exposes Auth and Match services; client maintains a bidirectional stream (MatchService.Stream) for real-time updates (snapshots, errors, etc.).  
- Platforms: Start with a terminal (ncurses) client for fast iteration; later add desktop/mobile GUIs (Qt, iOS, Android).

2) History (highlights from this session)  
- Core Board: Implemented abstract board with rule enforcement for moves, bar/off, dice, cube offer/take/drop, undo/commit, and an opening-roll mechanic.  
- Renderer: Started with ASCII BoardRenderer; added NcursesRenderer for a cleaner terminal board using UTF‑8 line-drawing and checker glyphs.  
- CLI → TUI: Initial single-process CLI evolved into an ncurses TUI (bg\_tui) that connects to a gRPC server.  
- Split to server/client:  
 • server/: minimal gRPC server (AuthService, MatchService) with an in-memory 'match m1' prototype, login gating, and live Stream broadcasting snapshots and errors.  
 • client-tui/: TUI client connects, logs in, creates/joins matches, streams board updates, and sends commands (roll, step, set, undo, commit, double, take, drop, snap).  
- Build System: CMake for both server and client. Fixed several Homebrew/macOS issues (Protobuf double-discovery, include-order with ncurses OK macro, generated headers pathing/order). Generated sources placed under build/generated.  
- Rendering Iterations: Several fixes for board chrome (borders, bar, joints), numeric headers (spacing, bar gap), bar origins, off-area gutter, and resize handling. Introduced 3‑space group gap across the bar to align numbers.  
- Logging: Optional logs enabled with BG\_SERVER\_LOG / BG\_CLIENT\_LOG to write bg\_server.log and bg\_client.log in the working directory.

3) Current State (as of this summary)  
- Server:  
 • Listens on 127.0.0.1:50051 (override via client env).  
 • Services: AuthService (Login), MatchService (Stream with commands: CreateMatch, JoinMatch, LeaveMatch, RequestSnapshot, RollDice, SetDice, ApplyStep, UndoStep, CommitTurn, OfferCube, TakeCube, DropCube).  
 • Maintains a single in-memory match (extensible to multiple).  
 • Board engine enforces phases: OpeningRoll, AwaitingRoll, Moving, CubeOffered; rejects illegal actions with error codes (409 not-in-phase, 403 not-your-turn, 404 not-found, 401 auth).  
 • Broadcasts Snapshot/Error envelopes to all match participants (players + observers).  
- Client (bg\_tui):  
 • ncurses UI showing board, status (phase/side/dice/cube holder/last message), and input line.  
 • Commands: login u p | create MID | join MID [white|black|obs] | leave | roll | set d1 d2 | step FROM PIP | 'a b' shorthand | undo | commit | double | take | drop | snap | quit.  
 • Resize handling: redraws on KEY\_RESIZE and after actions; UTF‑8 line drawing for border/bar; checker glyphs ● (white) and ○ (black).  
 • Known working flow: login → create mX → join mX white/black → request snapshot → opening via set d1 d2 (or roll if using RNG) → step per die → commit → opponent turn, etc.

4) Known Issues / Todos  
- Renderer: Ensure PO[] (point-to-column map) remains consistent with chrome; we saw a regression where numbers were right but stacks were offset. Keep a “golden snapshot” test to catch spacing regressions.  
- Roles & Permissions: Continue to tighten who can act when (only side-to-move acts; observers read-only). We already block most illegal-phase actions with 409, but polish is ongoing.  
- Game flow:  
 • Permit immediate commit when a player is stuck on the bar and opponent’s home board is closed (no legal entries).  
 • Detect game over (bear-off complete) and surface match result; scoring, gammons/backgammons with cube value.  
- Stability:  
 • Improve TUI redraw after resize to avoid occasional stale input artifacts.  
 • Continue to keep gRPC/proto headers before <ncurses.h> to avoid OK macro collisions.  
- Multiplicity: Move beyond the prototype single match to multiple concurrent matches with a registry; per-match access control and watcher lists.  
- Admin: Plan for a small web admin (separate process) to inspect users/matches/logs (future).

5) Directory Structure (current)  
bg/  
 board.hpp, board.cpp  
 boardrenderer.hpp, boardrenderer.cpp  
 ncurses\_renderer.hpp, ncurses\_renderer.cpp  
 CMakeLists.txt (legacy for mono build; current per-target CMake under server/ and client-tui/)  
 server/  
 CMakeLists.txt  
 main.cc  
 proto/bg/v1/bg.proto  
 build/ (generated on configure/build; contains proto copy and generated C++)  
 client-tui/  
 CMakeLists.txt  
 main.cc  
 build/ (generated on configure/build; contains proto copy and generated C++)  
 (Optional) tests/ (recommend adding a golden\_board.txt to catch renderer spacing regressions)

6) Build & Run (macOS + Homebrew)  
Prereqs:  
 brew install grpc protobuf  
Environment:  
 export LC\_ALL=en\_US.UTF-8  
 export TERM=xterm-256color

Server:  
 cd server  
 rm -rf build  
 cmake -S . -B build \  
 -DgRPC\_DIR="$(brew --prefix grpc)/lib/cmake/grpc" \  
 -DProtobuf\_DIR="$(brew --prefix protobuf)/lib/cmake/protobuf"  
 cmake --build build -j  
 BG\_SERVER\_LOG=1 ./build/bg\_server

Client (TUI):  
 cd client-tui  
 rm -rf build  
 cmake -S . -B build \  
 -DgRPC\_DIR="$(brew --prefix grpc)/lib/cmake/grpc"  
 cmake --build build -j  
 BG\_CLIENT\_LOG=1 ./build/bg\_tui

7) Runtime Controls (TUI)  
- Enter: commit (if in Moving phase)  
- Shorthand: “FROM PIP” → step FROM PIP (e.g., “13 5”)  
- Commands: help, login, create, join, leave, roll, set, step, undo, commit, double, take, drop, snap, quit

8) Environment Variables  
- BG\_SERVER\_LOG (server) — if set, writes bg\_server.log  
- BG\_CLIENT\_LOG (client) — if set, writes bg\_client.log  
- BG\_SERVER\_ADDR (client) — target server (default 127.0.0.1:50051)

9) Protocol Notes  
- Envelope { header{ proto\_version, match\_id }, oneof cmd/evt }  
- Events: Snapshot (BoardState + diceRemaining + phase + cubeHolder + sideToMove), Error (code/message)  
- Commands: CreateMatch, JoinMatch, LeaveMatch, RequestSnapshot, RollDice, SetDice, ApplyStep, UndoStep, CommitTurn, OfferCube, TakeCube, DropCube  
- BoardState carries points[24]{side,count}, white\_bar, black\_bar, white\_off, black\_off, cube\_holder, side\_to\_move, phase, dice\_remaining[]

10) Resuming in a New Session (Quick Primer)  
Provide:  
- Branch + short SHA, toolchain versions  
- Exact CMake lines used for both server and client  
- Terminal size (cols×rows)  
- Current ncurses\_renderer.hpp constants: X\_LEFT, X\_RIGHT, X\_BAR\_L/R/C, X\_OFF  
- A screenshot or pasted TUI render of initial board  
With this, we can lock instantly to your tree and avoid regressions.

11) Next Steps (suggested)  
- Finalize PO[] vs chrome invariants and freeze a golden snapshot test  
- Enforce role permissions strictly in MatchService  
- Implement game-over detection and scoring  
- Add multiple matches and a small registry (id → match struct, user sets, streams)  
- Optional: Add seedable RNG and server-side dice by default; client “set d1 d2” remains for testing