Kickoff Prompt for Next Session

"I am continuing the **SGFPlayerClean** project.

Current Context:
We have a working UI (Lobby, Challenge Creation, Board Display) and a working REST-based initialization (Handicap stones load correctly).
The Critical Bug: We are in a **"Zombie Play" state**. We can place stones locally, and the socket sends a `game/move` message, but the OGS server **ignores it** (no confirmation, no update on opponent's screen).

Primary Objective:
Fix the **Socket Protocol** in `OGSClient.swift` to match the Web Client's behavior exactly. We strongly suspect we are sending malformed payloads (extra fields) or connecting out of sequence (before authentication confirms).

Files to Analyze:

1. `OGSClient.swift` (The networking core - needs protocol fixes).
2. `SGFCoordinates.swift` (To verify we are sending valid SGF strings like "pd").
3. `AppModel.swift` (To verify the sequence of connection steps).
   "

Handover Document: SGFPlayerClean (v3.181)

Date: December 20, 2025
System Time: Saturday, 2:45 AM (PST)
Focus: Protocol Compliance, Authentication Sequencing, and Payload Strictness

1. Executive Summary

The application correctly joins games visually (via REST fetch), but the **Socket Connection** is dysfunctional for gameplay.

- **Symptoms:** User moves -> Socket sends `game/move` -> Server stays silent -> Local board desynchronizes.
- **Root Cause Analysis:** Comparison with Web Client logs indicates we are **"Polluting the Payload"**. We send `auth` and `player_id` in messages where the Web Client sends *only* `game_id`. The server likely rejects these malformed packets or treats the connection as unauthorized because the Global Auth handshake wasn't confirmed first.

2. Technical Debt & Architectural Risks

A. The "Payload Pollution" Issue

We observed that the Web Client sends minimal JSON, whereas we send maximal JSON.

- **Web Client (Connect):** `["game/connect", {"game_id": 123, "chat": true}]`
- **Our Client (Connect):** `["game/connect", {"game_id": 123, "chat": true, "player_id": 456, "auth": "..."}]`
- **Web Client (Move):** `["game/move", {"game_id": 123, "move": "qd"}]`
- **Our Client (Move):** `["game/move", {"game_id": 123, "move": "qd", "player_id": 456, "auth": "..."}]`

Hypothesis: The OGS server validates payload schema strictly. Sending extra fields might cause the server to drop the packet as "Malformed" without sending an error back.

B. The Authentication Handshake Sequence

Currently, `OGSClient` fires `authenticate` and then immediately fires `game/connect` (via AppModel).

- **Risk:** Socket.IO messages are asynchronous. If `game/connect` arrives before the server processes `authenticate`, the game connection is established as **Anonymous**.
- **Requirement:** We must implement a listener for the `authenticate` **response** (likely a message or a specific ACK) before we allow `connectToGame` to run.

C. Variable Types & Coordinate Systems

- **Coordinates:** We assume OGS uses standard SGF `aa`-`ss`. If `SGFCoordinates.swift` is zero-indexed but OGS expects 1-indexed (or vice-versa), or if we are inverting Y-axis, we might be sending "Invalid Moves" (e.g., placing a stone on top of an existing one).
  - *Check:* Web logs show "qd". We must ensure our (16, 3) converts to exactly "qd".
- **Move Format:** `game/move` (outgoing) expects a **String**. `game/move` (incoming broadcast) uses an **Array** `[x,y,dt]`. We must maintain this asymmetry.

D. Missing "Keep-Alive" Subscriptions

Web logs showed subscriptions to `active_game`, `hostinfo`, etc.

- **Risk:** We only subscribe to `seekgraph/global`. It is possible that `game/connect` requires the socket to *also* be subscribed to user-specific notification channels to work correctly as a player.

3. Immediate Action Plan

1. **Strict Payload Cleanup:** Modify `OGSClient.swift` to strip `auth` and `player_id` from `connectToGame` and `sendMove`. Match the Web Client *exactly*.
2. **Auth Gating:** Modify `OGSClient` to pause game connections until `authenticate` has been acknowledged (if possible) or ensure a strict delay/callback structure.
3. **Coordinate Verification:** Review `SGFCoordinates.swift` to ensure `toSGF` matches OGS expectations (0-18 range, a-s char mapping).

4. Key Files for Next Session

- **`OGSClient.swift`** (Target for fixes)
- **`AppModel.swift`** (Orchestrator)
- **`SGFCoordinates.swift`** (Data integrity check)

41.7s

Use Arrow Up and Arrow Down to select a turn, Enter to jump to it, and Escape to return to the chat.