Handover Document: SGFPlayerClean (v4.500)

Date: December 28, 2025

Current Status: Building & Verified. Project successfully restructured, "Zombie Play" bug fixed, OGS Lobby populating.

1. Kickoff Script for the Next Session

"I am continuing the SGFPlayerClean project. My Xcode project is organized into Models, ViewModels, Views, and Services. We have successfully reached **v4.208**, which builds and populates the lobby. We have verified the OGS 'Golden DNA' protocol (auth token + player_id required for moves). I have Handover Document v4.500. Please acknowledge the Rules of Engagement. My first task is to fix any remaining broken links in the **joining game logic** to ensure that when I accept a challenge, the board opens and I can play a move."

2. Rules of Engagement for the AI

- **Trust the User:** Accept sidebar states and folder paths as absolute truth.
- **No Guessing:** Never assume or guess the content of a file. If a file hasn't been pasted in the current session, **ask the user** to provide it.
- **Terse Descriptions:** Concise, direct explanations are preferred over long-winded prose.
- **Full Code Drop-ins Only:** Never provide partial snippets. Provide the **entire file** content so the user can "Select All > Paste."
- **Single Source of Truth:** All Enums, Structs, SGF Parsing logic, and Coordinate handling **must** live in `OGSModels.swift`.
- **Zero Redundancy Policy:** If a type is moved to `OGSModels.swift`, any individual files containing that type (e.g., `SGFKit.swift`) **must be deleted** immediately to prevent "Ambiguous Type Lookup" errors.

3. Verified OGS Protocol: The "Golden DNA"

The server ignores moves (the "Zombie Play" bug) unless these exact

specifics are met:

Connection & Keep-Alive
- **Endpoint:** `wss://wsp.online-go.com/` (Do NOT use `/socket.io/` path).
- **Framing:** Plain JSON Arrays. Do NOT use Socket.IO prefixes (e.g., 42).
- **High-Level Heartbeat (net/ping):** Send `["net/ping",{"client": <ms_timestamp>}]` every 5 seconds.

Write-Access Move Payload
To register a move as a **Player**, the payload must contain exactly these four keys:

1. `"game_id"`: Numerical ID.
2. `"move"`: SGF coordinate string (e.g., `"pd"`).
3. `"auth"`: Game-specific secret token (fetched via REST `/api/v1/games/ID`).
4. `"player_id"`: User's numerical OGS ID.

Lobby (Seekgraph) Verified Data
- **Event Name:** `seekgraph/global`.
- **Structure:** Arrives as an **Array** of objects or a **single** object.
- **ID Key:** Uses `challenge_id` (not `id`).
- **Deletion:** Indicated by `"delete": 1` inside the object.

4. Architectural Map

1. The Engine (`SGFPlayerEngine.swift`)
- **Class Name:** `SGFPlayer`
- **Role:** Pure Go logic (captures, liberties, move list). Network-agnostic.

2. The Bridge (`BoardViewModel.swift`)
- **Role:** The sole orchestrator. Bridges the Engine and the Network.
- **Synchronization:** Uses a Combine `.sink` to observe the Engine and trigger UI refreshes.
- **State:** Holds `isOnlineContext` to toggle between live play and SGF playback.

3. The Network (`OGSClient.swift`)
- **Role:** Handles WebSocket traffic and REST calls.
- **State:** Stores `activeGameAuth`, `userJWT`, and the `availableGames` lobby list.

4. The Truth (`OGSModels.swift`)
- **Role:** Central vault for all shared types, the `SGFParser`, and `KeychainHelper`.

5. File Registry & Build Status
- **Xcode Version:** 16.4 (Requires strict pathing for Entitlements).
- **Entitlements Path:** `SGFPlayerClean/Models/SGFPlayerClean.entitlements`.
- **Deleted Files (Do not suggest):** `SGFKit.swift`, `SGFCoordinates.swift`, `SGFGameWrapper.swift`, and all Test folders.

6. Critical Variables to Watch

Variable	Scope	Purpose
`activeGameAuth`	`OGSClient`	**Critical.** If nil, server ignores moves.
`isOnlineContext`	`BoardVM`	Determines if stone clicks go to the server or the local engine.
`currentIndex`	`SGFPlayer`	The pointer for SGF playback.
`gamePhase`	`OGSGameVM`	Used by `SupportingViews` to show game status.

7. Added Intelligence (Moving Forward)
- **The "JSON Mirror" Rule:** When parsing OGS messages, use exact key names found in DevTools (e.g., `challenge_id`), even if they are inconsistent with our internal camelCase naming.
- **Derived Data Wipe:** If "Ambiguous Type" errors return without

reason, the user should delete the `DerivedData` folder in Xcode Settings.

- **Notification Names:** Ensure `AppModel` observers match the names posted in `OGSClient` (e.g., `OGSGameDataReceived`, `OGSMoveReceived`).