Here is a clear opening prompt you can use to start the next session, followed by the expanded and refined **Handover Document v10.500**.

Part 1: Opening Prompt for the Next Session

"We are continuing the **SGFPlayerClean** project, specifically focusing on the OGS (Online Go Server) Live Game Integration. I am about to upload **Handover Document v10.500**, which contains the results of a high-resolution protocol sniffing session.

We have successfully captured the 'Golden Payloads' for both authentication and game moves from the official OGS web client. Our current hurdle is a persistent 'Invalid JWT' error when connecting via terminal, which we suspect is due to missing Session Cookies or User-Agent headers.

Before we start coding, please acknowledge the **Rules of Engagement** found in the document—specifically the requirement for **Full Code Drop-ins** and the **'No Guessing'** policy regarding the OGS protocol. Once processed, we will begin by mimicking the browser's handshake to establish a stable authenticated terminal connection."

Part 2: Handover Document v10.500 (Extended)

1. Project Status & Immediate Priority

- **Status:** Protocol Sniffing phase.
- **Immediate Priority:** Stabilize the WebSocket authentication. We have the correct JSON payloads but are being rejected at the handshake/auth level.
- **Goal:** Once the terminal (`websocat`) can successfully place a stone on the live OGS board, we will port that exact logic into the Swift codebase (`OGSClient.swift`).

2. The "Truth Chain" (Captured Protocols)

The following payloads were captured directly from the official OGS web client in a live match. **These are the absolute sources of truth.**

A. Authentication Payload (Full Metadata)

The server expects a rich metadata object, not just a JWT string.

code

JSON

```
["authenticate", {
  "jwt": "...",
  "client_version": "5.1-8906-g13c006f1",
  "device_id": "b49fccee-5257-4c2c-b1b0-d1be2e9942e8",
  "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X
10_15_7) AppleWebKit/537.36...",
  "language": "en",
  "language_version": "5f2d999d34b5e30d26ea920e1f415047"
}]
```

- **Observation:** The bracket `[` start suggests that OGS may be using a raw WebSocket implementation or a modified Socket.io that does *not* always require the `42` prefix in this specific environment.

B. Outgoing Move Payload (Client → Server)

code

JSON

```
["game/move", {
  "game_id": 82742150,
  "move": "cc",
  "blur": 20994,
  "clock": { "main_time": 3562925, "timed_out": false }
}, 2]
```

- **Coordinate Truth:** Moves are sent as **Strings** (e.g., `"cc"`), not arrays.
- **Metadata Truth:** The `blur` (likely milliseconds of hover time) and `clock` objects appear to be mandatory.
- **Framing Truth:** The trailing integer (`2`) is a Message ID/Sequence number.

C. Incoming Move Payload (Server → Client)

code

JSON

```
["game/82725606/move", {
  "game_id": 82725606,
  "move_number": 12,
  "move": [0, 8, 46562]
}]
```

- **Path Truth:** The server sends moves to a specific path: `game/{ID}/move`.

3. Lessons Learned & Areas for Caution

- **JWT Sensitivity:** Fresh tokens from the `ui/config` API were repeatedly rejected as "Invalid."
    - **Caution:** OGS likely validates the JWT against the `sessionid` cookie and `User-Agent`. When using `websocat`, we **must** provide these headers to mimic the browser session exactly.
- **Terminal Buffer Overload:** The server sends an `active-bots` list exceeding 100k characters.
    - **Caution:** This crashes standard terminal displays or causes massive lag. Always use `websocat -B 200000` to expand the buffer and use `grep` to filter for relevant data.
- **The "10-Second Rule":** Failing to respond to `net/ping` with a `net/pong` within ~10 seconds results in an immediate "I/O failure" disconnect.
- **Retina Coordinate Mismatch (3D):** (Carried from v9.000) Click detection in the 3D stack is currently failing on high-DPI screens. We suspect `UnifiedInteractiveSceneView` needs a coordinate conversion from points to pixels for the `hitTest`.

4. Rules of Engagement (Strict Enforcement)

1. **Full Code Drop-ins:** Do not provide "code snippets" or "diffs." Provide the **entire** file content to ensure no lost brackets or misaligned closures.
2. **No Guessing:** Never assume a protocol change. If a move fails, go back to the browser DevTools and sniff the "Truth Chain."
3. **Linear State Chain:** The `SGFPlayerEngine.swift` is the ONLY source of truth for board state. `BoardViewModel` must only reflect what the Engine says.
4. **Preserve the Logic:** When updating `OGSClient.swift`, ensure that the "Handshake Mimicry" (Headers/Cookies) is implemented as a

configuration, not a hardcoded hack.

5. Technical Reference (Current IDs)

- **NewClient Account ID:** `1872928`
- **Most Recent Game ID:** `82742150` (9x9)
- **Handshake URL:** `wss://wsp.online-go.com/`
- **Origin Header:** `https://online-go.com`

6. Methodology for Next Session

1. **Step 1: Header Capture.** User will retrieve the `sessionid` cookie from the browser.
2. **Step 2: Authenticated Handshake.** Execute `websocat` with `-H "Cookie: sessionid=..."` and `-H "User-Agent: ..."`
3. **Step 3: Manual Move.** Once authenticated, the user will manually paste the "Golden Move Payload" to confirm the server accepts it.
4. **Step 4: Swift Integration.** Once Step 3 succeeds, the LLM will provide a full-file replacement for `OGSClient.swift` to automate this behavior.

End of Handover Document v10.500