Handover Document v8.600: SGFPlayerClean Project

1. Project Kickoff Script

"I am continuing the SGFPlayerClean project. We have reached **v8.600**, which features a high-performance 3D interaction layer with 'Object Pulling' physics, viewport persistence, and instant move playback. The system has been hardened against Main-Thread stalls and SceneKit deadlocks. I have Handover Document v8.600. Please acknowledge the Rules of Engagement. My next priority is to ensure **Lid Variety** (Clam assets) and complete the **OGS REST integration** for joining live games."

2. Rules of Engagement (Strict Enforcement)

- **Trust the User:** Accept sidebar states and folder paths as absolute truth.
- **No Guessing:** Never assume the content of a file. If a file hasn't been pasted in the current session, ask the user.
- **Full Code Drop-ins Only:** Provide entire file contents so the user can "Select All > Paste."
- **Single Source of Truth:** All Enums, Structs, SGF Parsing, and Coordinate handling must live in `OGSModels.swift`.
- **Zero Redundancy:** If a type is moved to `OGSModels.swift`, any individual files containing that type must be deleted immediately.
- **Spatial Consistency:** Maintain the **1:1.0773** aspect ratio for the board and the **6.5%** wood margin.
- **Linear Spatial Rendering:** Always render stones using the flat `stonesToRender` array calculated in a background Task to prevent Z-order flickering and Main Thread locking.

3. Updated Architectural Map

1. **The Root Coordinator (`AppModel.swift`):** Owns the lifecycle. Manages the transition between Local and Online modes.
2. **The Engine (`SGFPlayerEngine.swift`):** Pure Go logic. Finalizes board snapshots and pre-flattens stone lists to prevent UI-layer

scanning ($O(N^3)$

$O(N$

3

3

4)

fix).

5. **The Bridge (`BoardViewModel.swift`):** Orchestrates synchronization. Performs Jitter Math in background Tasks and signals the 3D view via `onRequestUpdate3D`.
6. **The Persistence Layer (`AppSettings.swift`):** Persists visual preferences and the **3D Viewport state** (Rotation, Zoom, Pan).
7. **The Truth (`OGSModels.swift`):** Central vault for shared types, SGF Parser, and Network models.
8. **The 3D Scene (`SceneManager3D.swift`):** High-performance imperative rendering. Uses **Node Diffing** (only adding/removing changed stones) instead of scene reconstruction.

4. Performance & Threading Lessons (The "Lag Fix")

- **Bypassing SwiftUI Diffing:** SwiftUI's `.onChange(of: array)` is too slow for 3D. The 3D view now uses `.onReceive(boardVM.onRequestUpdate3D)` to trigger updates manually.
- **SCNAction over SCNTransaction:** To prevent deadlocks, animations (stone drops) use `SCNAction`. This offloads the animation

timeline to the GPU/SceneKit thread.

- **Settings Caching:** To avoid `os_unix.c` Disk I/O stalls, `AppSettings` are cached into local variables once per move update.
- **Anti-Aliasing:** The project standard is `.multisampling4X`. Higher values or improper implementation on macOS can cause 5-second driver stalls.

5. 3D Interaction Logic ("Object Pulling")

The 3D interaction follows **User Intent**:

- **The Rotation Signs:** Dragging **Down** increases X-rotation (tilts board forward). Dragging **Right** increases Y-rotation (turns board clockwise).
- **Overhead View:** Rotation limit is **1.57 radians (90°)**, enabling a perfect top-down rectangular view.
- **World Anchor:** Panning and Rotating are applied to a `worldAnchor` node. This makes it feel like the user is "grabbing and moving the board" while the camera remains a stationary observer.
- **Modifiers:**
  - **Default Drag:** Rotate board.
  - **Shift + Drag:** Pan/Translate board.
  - **Ctrl + Shift + Drag:** Zoom in/out.

6. Visual Marker & Coordinate Logic

- **The Sibling Anchor Strategy:** To prevent markers (Numbers, Dots) from being squashed by the stone's `0.486` scale, we use a shared parent `anchorNode`. The stone (scaled) and the markers (unscaled) are siblings.
- **Apex Math:** Markers are placed at exactly `stoneRadius * stoneScaleY * 2.0` (the true top surface).
- **Move Number Contrast:** Logic is implemented to use **White text on Black stones** and **Black text on White stones**.
- **Board Glow:** Implemented as an `SCNPlane` with a pre-generated red

radial gradient. `renderingOrder` is **3000** with `writesToDepthBuffer = false`.

7. Build Stability & Compiler Safety

- **The buildExpression Error:** SwiftUI's compiler times out if a View body contains too much math. Perform all layout math in private helper functions (e.g., `getLayout()`) or local constants.
- **Button Safety:** Avoid placing logic calls (e.g. `boardVM.step()`) inside Button labels; use `Image` or `Text` only.
- **Contextual Base Errors:** Modifiers like `.padding(.bottom)` can fail if the compiler loses track of the View type. Use `EdgeInsets` for complex layouts.

8. Diagnostic Breadcrumb Legend

The console logs the pipeline sequence using emojis to identify stalls:

- 🎹 **Input:** User interaction detected.
- 📢 **Engine:** Move processed by the Go rules engine.
- 🧪 **VM Math:** Jitter/Coordinate math completed in background.
- 🎭 **Actor:** Data handed off to the Main Actor.
- 🎬 **GPU:** SceneKit transaction committed to hardware.

9. Verified OGS Protocols

- **Endpoint:** `wss://wsp.online-go.com/` (No `/socket.io/` path).
- **Heartbeat:** `["net/ping", {"client": <ms_timestamp>}]` every 5 seconds.
- **Authentication:** `["authenticate", {"jwt": "<token>"}]` sent immediately upon connection.
- **Seekgraph:** `["seek_graph/connect", {"channel": "global"}]` for live game list.
- **Write-Access Payload:** Requires `game_id`, `move`, `auth` (from REST), and `player_id`.

10. Critical Variable Mapping (The "Truth Chain")

Entity	Variable Name	Physical Meaning
Engine	`whiteStonesCaptured`	White stones physically pulled off the board.
ViewModel	`blackCapturedCount`	Black player's loot (Count of White stones he holds).
3D Lid	`upperLidStones`	The visual White stones appearing in Black's (top) lid.
2D View	`SimpleLidView(.black)`	Displays the `whiteCapturedCount` as White stones.

11. Near-Term Agenda

1. **Lid Variety:** Refactor `SceneManager3D.updateCapturedStones` to deterministically pick from 5 Clam assets.
2. **OGS REST Bridge:** Finalize the `/api/v1/games/{id}` call to fetch the `auth` token.
3. **UI Polish:** Ensure the 3D viewport persistence handles window resizing gracefully.