VLM Data Flow Debugging Guide

Overview

This guide explains the complete data flow of VLM (Visual Layout Metadata) data from DrillForge AI generation to display on the interactive field.

Data Flow Chain

1. DrillForge AI Generation

Location: `/api/ai-drills/route.ts` and `lib/drillforge/api.ts`

DrillForge AI generates drill with VLM enhancements (zones, goals, cones, movementArrows).
VLM data is returned in the `aiEnhancement` object and merged to root level.

2. Saving to Database

Location: `/api/generated-drills/route.ts` (POST handler)

VLM data is extracted and stored inside the `formationData` JSON field:
- zones → formationData.zones
- goals → formationData.goals
- cones → formationData.cones
- movementArrows → formationData.movementArrows

3. Retrieving from Database

Location: `/api/generated-drills/[id]/route.ts` (GET handler)

Returns drill with formationData containing VLM elements.
NEW: Added debug logging to verify VLM data presence.

4. Field Page Transformation

Location: `/dashboard/field/page.tsx`

Extracts VLM data from formationData and moves to root level for InteractiveField.
NEW: Added debug logging to track transformation.

5. InteractiveField Display

Location: `components/dashboard/interactive-field.tsx`

Receives aiDrillData prop and displays VLM elements.
NEW: Added debug logging to verify data reception.

Debugging Steps

1. Generate a new drill with DrillForge AI
2. Save the drill
3. Launch drill to field

4. Check console logs in this order:
   - Server: API GET response with VLM counts
   - Browser: formationData structure
   - Browser: transformedDrill VLM data
   - Browser: InteractiveField received data

What to Look For

Check these logs in browser console:
- “DEBUG - formationData structure” - shows if data exists in API response
- “DEBUG - transformedDrill VLM data” - shows if extraction worked
- “DEBUG - aiDrillData structure in InteractiveField” - shows if data reached component

Compare the counts at each step to find where data is lost.