Cognitive Loops as Game Architecture

The Insight

QRF isn't just a card game — it's a **compressed OODA loop trainer**. Every swipe is Observe (read the card), Orient (context from your mission, rules of engagement, prior cards), Decide (left or right), Act (swipe). The genius is that the loop is so compressed the player doesn't consciously decompose it. They just *feel* faster and sharper after reps. The OODA loop is the invisible skeleton; the card mechanic is the skin.

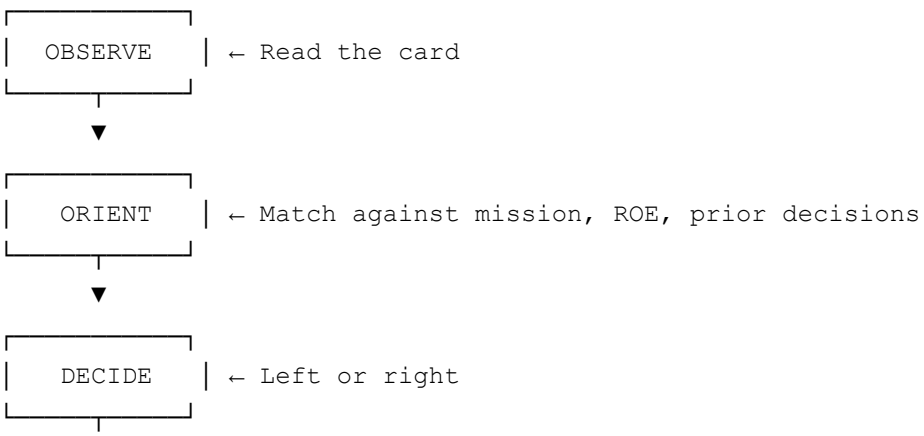
Every game in the suite should have a loop like this. Not the same loop — a *different* loop, targeting a different cognitive skill. The loops are what make these training tools rather than just games. And crucially, the loops are what structure the decision-capture data: each loop iteration is one data point for the AI.

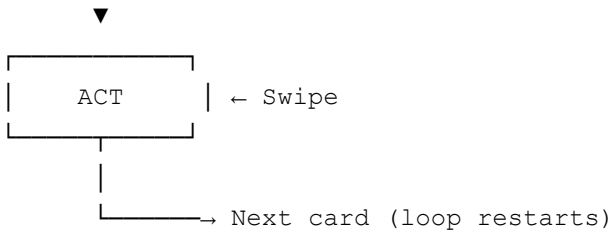
The Loops

1. OODA Loop — QRF

Observe → Orient → Decide → Act

Boyd's original loop. The key insight most people miss: the competitive advantage isn't in the speed of any single phase — it's in the **Orient** step, where you match the incoming observation against your mental model of the situation. Experts have richer mental models, so they orient faster, so they decide faster. QRF trains this by giving hundreds of reps where the "orient" step is the bottleneck: the card is simple, the context is complex.





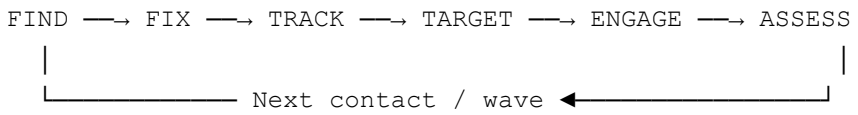
Loop duration: 3–10 seconds. **Reps per session**: 50–100+. **What the AI captures**: The Orient→Decide mapping at scale.

---

2. Kill Chain — GHOST GRID

Find → Fix → Track → Target → Engage → Assess (F2T2EA)

GHOST GRID already embodies this, whether you've named it or not. The radar scope presents contacts (Find). The player classifies and geolocates them (Fix). They monitor trajectory and behavior (Track). They select the right effector — jammer, airburst, interceptor (Target). They commit the weapon (Engage). They see the result (Assess). Then the next wave comes.



Loop duration: 10–30 seconds per contact, overlapping across multiple contacts simultaneously. **What makes it hard**: Multiple contacts in different phases of the kill chain at the same time. The player is Finding contact 7 while Tracking contact 4 while Assessing the result on contact 2. **What the AI captures**: Prioritization decisions — given N contacts at various kill chain stages, which one does the expert advance next?

Design implication: If you haven't already, consider making the kill chain phases *explicit* in the UI. Let the player see which phase each contact is in. This reinforces the mental model and makes the decision-capture data cleaner.

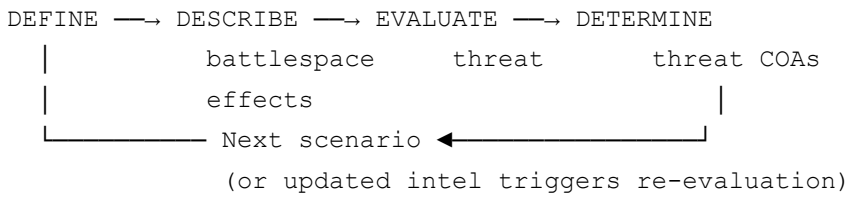
---

3. IPB Cycle — WARLORD

Define → Describe → Evaluate → Determine (D-D-E-D)

Intelligence Preparation of the Battlefield is a four-step cycle that maps perfectly to WARLORD's game phases:

IPB Step	WARLORD Phase	What the Player Does
Define the battlespace	SITUATION	Receive the AO, boundaries, and constraints
Describe the battlespace effects	SITUATION	Read the terrain, weather, and civil considerations
Evaluate the threat	ASSESS	Analyze the enemy OOB, commander profile, and intelligence fragments
Determine threat COAs	ASSESS → COMMIT	Predict MLCOA and MDCOA, mark on the map, lock it in



Loop duration: 3–7 minutes per scenario. **What makes it hard**: The Evaluate step. Novices skip it — they see "aggressive commander" and jump to a COA without weighing the indicators. Experts dwell in Evaluate, cross-referencing fragments, looking for inconsistencies, checking for deception. **What the AI captures**: The Evaluate→Determine mapping — given these inputs, what do experts predict?

Design implication: Consider making the IPB steps visible in the UI as a progression. The player physically moves through Define → Describe → Evaluate → Determine. This teaches the process, not just the outcome. A player who jumps straight to Determine without spending time in Evaluate should see that reflected in their reasoning score.

---

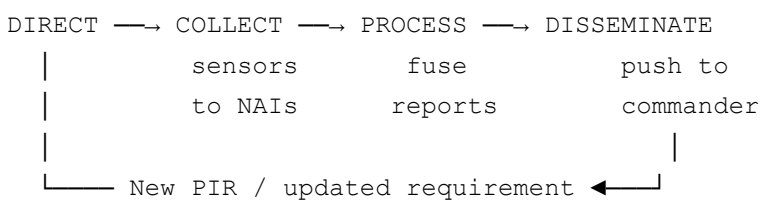
4. Intelligence Cycle — OVERWATCH

Direct → Collect → Process → Disseminate (D-C-P-D)

The intelligence cycle governs how information moves from a question to an answer:

Step	OVERWATCH Mechanic
Direct	Player assigns ISR assets to NAIs based on PIRs
Collect	Time passes; assets return fragmentary reports

Process	Player fuses reports, resolves conflicts, builds the picture
Disseminate	Player pushes assessment and recommendation to the commander



Loop duration: 2–5 minutes per cycle, multiple cycles per scenario. **What makes it hard**: The feedback delay. You direct a sensor now, but the collection comes back in 60 seconds. By then, the situation may have changed. The player has to think ahead: "What will I need to know in two minutes?" This is *anticipatory* collection, not reactive. **What the AI captures**: The Direct step — given a partial picture and a set of questions, where do experts point their sensors?

Design implication: The cycle should literally loop within a single scenario. The player directs assets, gets results, processes them, disseminates, then redirects based on what they learned. Each loop refines the picture. A good OVERWATCH player completes 3–4 intelligence cycles in a scenario; a novice completes 1–2 and runs out of time.

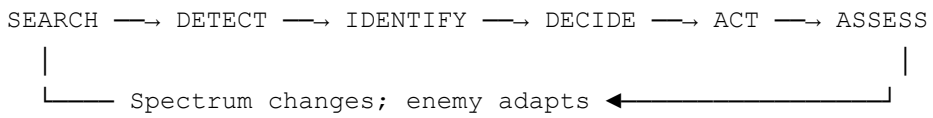
---

5. EW Battle Management Cycle — BLACKGLASS

Search → Detect → Identify → Decide → Act → Assess (Modified kill chain for the EM spectrum)

This is the kill chain adapted for electronic warfare, where "engage" might mean jamming, exploiting, or going silent:

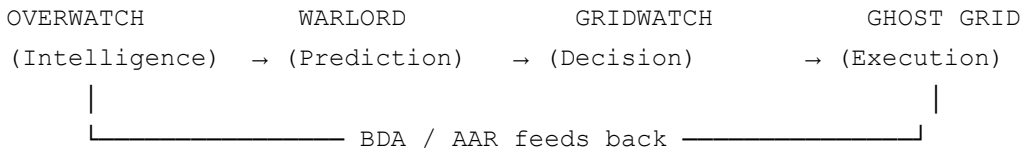
Step	BLACKGLASS Mechanic
Search	Scan the spectrum for emitters
Detect	Notice a new signal on the waterfall
Identify	Classify the signal — friendly, enemy, unknown, civilian
Decide	Jam it? Exploit it? Ignore it? Protect your own adjacent frequencies?
Act	Execute the EW action
Assess	Did the jam work? Did the enemy adapt? Did you break your own comms?



Loop duration: 15–45 seconds per emitter decision. **What makes it hard**: The Assess step feeds back *negatively*. Unlike kinetic engagement where success = target destroyed, EW success often looks like absence — the drone stopped transmitting, the enemy went quiet. But did they go quiet because you jammed them, or because they're switching frequencies? The player has to assess ambiguous results and decide whether to continue the action or adapt. **What the AI captures**: The Decide step — given a spectral environment and a set of identified emitters, what do experts prioritize and what trade-offs do they accept?

The Meta-Loop

Zoom out and the games themselves form a loop:



This mirrors how tactical operations actually work: intelligence feeds prediction, prediction informs decision, decision drives execution, and the results of execution generate new intelligence. Each game trains one segment of this meta-loop. A leader who plays all four understands the entire cycle.

This also suggests a **campaign mode** that chains the games together: an OVERWATCH session produces an intelligence picture, which feeds a WARLORD session to predict the enemy, which feeds a GRIDWATCH session to plan and execute, which triggers a GHOST GRID session when the enemy sends drones. The outcomes loop back to OVERWATCH for the next cycle.

Other Loops Worth Considering

Targeting Cycle (D3A): Decide → Detect → Deliver → Assess

This is the fires version of the kill chain. Could be the backbone of the FIRESTORM concept

(joint fires coordination game). The player decides which targets to prosecute, detects them through sensor-to-shooter links, delivers the appropriate munition, and assesses battle damage. The loop repeats as new targets emerge and the fire support plan adapts.

PDCA / Deming Cycle: Plan → Do → Check → Act

More relevant to logistics and sustainment games than tactical combat. If you ever build a convoy / logistics game, this is the natural loop: plan the movement, execute it, check progress against the plan, and adapt when reality diverges.

SALUTE → SPOTREP → SITREP → INTSUM

Not a decision loop, but a *reporting* loop. Every game could have a reporting mechanic layered on top: after making decisions, the player has to file a concise report. This trains a completely different skill — communication under pressure — and the captured reports are valuable NLP training data for the AI. A leader who makes the right call but can't communicate it clearly is almost as dangerous as one who makes the wrong call.

Design Principles Across All Loops

1. The loop should be invisible to the player

QRF doesn't say "you are now in the Orient phase." The player just swipes. The loop structures the experience and the data capture, but the player experiences flow, not a framework. Name the loops in internal docs and instructor guides, not in the player-facing UI.

2. Reps over duration

Each loop iteration should be short enough that the player completes many of them per session. The learning comes from volume and variation, not from single high-fidelity experiences. This is the QRF insight applied universally.

3. One loop per game

Don't try to train the full OODA loop *and* the kill chain in the same game. Each game should be about one cognitive loop. The suite covers the full space; each game masters one slice.

4. The bottleneck step is the training target

In every loop, one step is the hardest and most undertrained. That step should get the most time and attention in the game:

Game	Loop	Bottleneck Step
QRF	OODA	Orient
GHOST GRID	Kill Chain	Target (effector selection under time pressure)
WARLORD	IPB	Evaluate (weighing intelligence)
OVERWATCH	Intel Cycle	Direct (anticipatory sensor tasking)
BLACKGLASS	EW BM Cycle	Decide (accepting trade-offs)

5. Every loop iteration is one AI data point

The loop defines the capture schema. One OODA cycle in QRF = one decision record. One IPB cycle in WARLORD = one prediction record. One intel cycle in OVERWATCH = one tasking record. This keeps the data clean and directly trainable. The loop *is* the data structure.

Summary Table

Game	Cognitive Loop	Loop Duration	Reps/Session	Bottleneck Step	AI Captures
QRF	OODA	3–10 sec	50–100+	Orient	Decision mappings
GHOST GRID	Kill Chain (F2T2EA)	10–30 sec	20–50	Target	Prioritization
WARLORD	IPB (D-D-E-D)	3–7 min	10–20	Evaluate	Predictions
OVERWATCH	Intel Cycle (D-C-P-D)	2–5 min	15–30	Direct	Sensor tasking
BLACKGLASS	EW BM Cycle	15–45 sec	30–60	Decide	Trade-off acceptance