

x_make_progress_board_x “ Orchestrator Advance Party

This package deploys the PySide6 progress board ahead of the command center. While the heavyweight GUI wakes up, the board watches the orchestration snapshot feed, mirrors stage outcomes, and broadcasts when it is safe to unlock the deck.

Mission Log

- Display stage checklists, repository detail tables, and orchestrator status pulled from `reports/make_all_progress.json`.
- Poll on an interval, normalize snapshot data, and auto-close after the worker run completes.
- Expose a simple controller that launches the board, monitors worker threads, and hands off to the full GUI once the mission finishes.
- Provide a CLI for manual observation or diagnostic sweeps.

Instrumentation

- Python 3.11 or newer.
- PySide6 6.7+ available in the active environment.
- Optional QA: Ruff, MyPy, pytest, Pyright.

Operating Procedure

```
1. python -m venv .venv
2. \.venv\Scripts\Activate.ps1
3. python -m pip install --upgrade pip
4. pip install -r requirements.txt
5. python -m x_make_progress_board_x --snapshot reports/make_all_progress.json
```

The board opens in a PySide6 window and exits once all stages report completion and the worker thread signals done.

Color Semantics and Streaming Refresh

- Green = completed (stage fully done)
- Yellow = attention (finished with issues; board remains partially checked)
- Red = blocked (hard failure)
- Blue = running, Grey = pending

When visitor streaming is enabled, the board reflects updates as events arrive“repositories populate progressively and the detail table shows the latest messages for the selected stage.

Evidence Checks

Check Command	--- ---	Formatting sweep	<code>python -m black .</code>		Lint interrogation	<code>python -m ruff check .</code>		Type audit	<code>python -m mypy .</code>		Static contract scan	<code>python -m pyright</code>		Functional verification	<code>pytest</code>	
-----------------	-----------	------------------	--------------------------------	--	--------------------	-------------------------------------	--	------------	-------------------------------	--	----------------------	--------------------------------	--	-------------------------	---------------------	--

Reconstitution Drill

During the monthly rebuild the board is launched standalone, pointed at a canned snapshot feed, and observed until it unlocks automatically. Logs, timestamps, and screenshot evidence are captured for Change Control.

Conduct Code

Any UI change must include tests, documentation updates, and a Change Control note. The board is the first interface operators see“precision only.

Sole Architect’s Note

I designed this board to lead the assault: it tracks every stage, syncs with worker telemetry, and knows exactly when to clear the full command center. Packaging it spreads that discipline across every orchestrator in the fleet.