Piper - Living Context

📁 Drives — Piper Guides

- **Key Files.txt** = File role map (GUI orchestration, state dot, hb text, layout const, scroll, avatar).
- **Anchor.txt** = Project rails/constitution (rules, chapters, invariants, runbook).
- **CLI+GUI Trailing & Testing Guide.txt** = 2-terminal run instructions + trailing mechanics + smoke/fix.
- **State Color Coding Guide.txt** = State→color map + dwell + code snippet + smoke test.
- **Config Files.md** = Explains dotfiles/configs: `.pre-commit-config.yaml`, `.editorconfig`, `pyproject.toml` vs runtime configs (like `layout_constants.py`).

Architecture

Root: `C:\Piper\scripts`

```
common/
  config.py        # app config helpers (shared)
  log_utils.py     # project log helpers (renamed from logging.py; no stdlib
shadow)
  types.py         # simple datatypes
  utils.py         # small shared utilities

core/
  bg_poller.py     # background tick/poll loop helpers
  bridge.py        # glue between core events and UI/log
  core_app.py      # top-level core bootstrap
  core_commands.py # command handling (CLI/Dev tools)
  core_machine.py  # state machine (SLEEPING→…→SPEAKING)
  event_queue.py   # queued events
  events.py        # event definitions
  flags.py         # runtime feature/DEV flags
  poll_helpers.py  # polling helpers
  router.py        # routes events to handlers
  startup.py       # init order, env
  state_defs.py    # state enums/labels
  timers.py        # timing utilities
  transition_plan.py # transition map/guard rails

entries/
  _app_gui_entry_impl.py # GUI glue: imports dpg_app/panes; schedules refresh
(target ≤150 lines)
  app_cli_entry.py        # CLI entrypoint
  app_gui_entry.py        # thin shim → _app_gui_entry_impl.run()
```

```
    app_wake_entry.py        # wake-only demo/entry

services/
  adapters/mock_asr_wake.py   # test double (wake+ASR)
  asr/vosk_adapter.py         # ASR adapter (modular)
  memory/                     # memory svc placeholder
  persona/                    # persona svc placeholder
  tts/speak_once.py           # single-shot TTS
  tts/tts_manager.py          # TTS manager stub/interface
  wake/porcupine_adapter.py   # wake-word adapter (modular)
  base.py                     # service base helpers
  cli_prompt.py               # current_prompt(), format_line()
  persona_adapter.py          # persona bridge
  asr_vosk.py                 # ⚠️ legacy shim → move to services/old/
  wake_porcupine.py           # ⚠️ legacy shim → move to services/old/

ui/
  components/
    chat_pane.py        # chat render area (DPG; dynamic refresh)
    controls_pane.py    # dev controls (flag-gated)
    logs_pane.py        # live log tail
    status_pane.py      # small indicators
  helpers/
    avatar_fix.py       # avatar sizing/crop helpers (pending tune)
    chatlog_writer.py   # (GUI) chat logging
    dev_adapters.py     # dev-only bridges
    dev_controls_mount.py # wire dev controls
    gui_ingest.py       # parse tail lines → UI state cues
    gui_loop.py         # GUI main loop helpers
    header_bridge.py    # fallback hb_label writer (shim)
    header_utils.py     # header utilities
    init_core.py        # init orchestration helpers
    layout_utils.py     # layout helpers
    refresh_core.py     # push-only refresh helpers
    scroll_utils.py     # autoscroll logic (single callback; multi-tag aware)
    sink_utils.py       # sinks for tailed lines
    state_dot.py        # state→color mapping utils
    tag_utils.py        # tag parsing
    theme_utils.py      # theme helpers
    viewport_utils.py   # viewport sizing/pos
  pane_parts/
    avatar_pane.py      # avatar region
    chat_region_pane.py # chat region (layout-only; renamed to avoid name
collision)
    header_bar.py       # SINGLE AUTHORITY: set_state_dot(), heartbeat label
    logs_pane.py        # logs region
  dpg_app.py            # DearPyGui bootstrap + viewport
```

```
heartbeat.py          # hb text source
layout_constants.py   # ALL geometry/theme numbers
panes.py              # compose components/pane_parts (no header logic)
state_header.py       # state label helpers (reads from log/model)
tailer.py             # file tailer for core.log
theme.py              # theme palette
_panes_impl.py, dev_tools.py, ipc_child.py
```

KGB

Latest realized snapshot tag: `KGB-2025-09-13_H03_wrap_ready`

Rails — Next Sprint (Post-H03)

Only unfinished items from **README_RAILS** + integrated parked items. Each rail is tiny, reversible, and mirrors-first. Use BEFORE→AFTER with bookends, then smoke and KGB.

I01 — Sleep reply parity (CLI vs DevTools)

- **Goal:** `sleep` produces a spoken line in **Chat** for both paths.
- **Microsteps:**
- Mirrors: confirm `core/core_commands.py` sleep branch output.
- **Patch:** normalize to `[TTS] Going to sleep.` so ingest gates trigger consistently.
- Residue: list any duplicate sleep-line logic in `gui_ingest` and park removals.
- **Smoke:** CLI `wake → sleep → help`: Chat shows `Hello sir!` → `Going to sleep.` → help list; Logs show `[STATE]` lines.
- **KGB:** `KGB-YYYY-MM-DD_I01_sleep_parity``

I02 — UTF-8 mojibake scrub (Chat/Logs)

- **Goal:** Replace `âœ"`, `â†'` with proper `✓`, `→` end-to-end.
- **Microsteps:**
- Env: verify PowerShell UTF-8 (UTF8 envs + `OutputEncoding` + `chcp 65001`).
- Tailer: confirm `encoding="utf-8", errors="replace"` on file open.
- Ingest shim: map known mojibake sequences → canonical glyphs; keep tiny dict.
- Residue: once clean, park removal of shim for later.
- **Smoke:** `wake → sleep`: no mojibake in Chat/Logs; arrows and check marks render.
- **KGB:** `KGB-YYYY-MM-DD_I02_utf8_clean`

I03 — Header symbol & separator hygiene

- **Goal:** No dangling glyph or separator when nothing follows; sarcasm toggle behaves.
- **Microsteps:**
- Mirrors: `ui/pane_parts/header_bar.py`, `ui/helpers/header_utils.py`.

- Guard rendering: show separator only when adjacent label exists; hide otherwise.
- Residue: remove duplicate header writers; keep **single authority** comment.
- **Smoke:** Toggle sarcasm on/off → header updates with no stray symbol.
- **KGB:** `KGB-YYYY-MM-DD_I03_header_hygiene`

I04 — Theme: light-blue palette (non-intrusive)

- **Goal:** Switch from gray to light-blue via feature flag; layout untouched.
- **Microsteps:**
- Mirrors: `ui/theme.py`, `ui/helpers/theme_utils.py` — add `LightBlue` palette constants.
- Apply behind flag in `ui/dpg_app.py` (`PIPER_UI_THEME=lightblue`).
- Residue: keep gray palette; document flag in guides.
- **Smoke:** GUI boots light-blue; readability unchanged.
- **KGB:** `KGB-YYYY-MM-DD_I04_theme_lightblue`

I05 — Avatar pane tiny tweak

- **Goal:** Keep stable fit; add small padding per note.
- **Microsteps:**
- Mirrors: nudge constants in `ui/layout_constants.py` only (few px).
- Ensure no duplicate geometry sources; update comments.
- Residue: list/remove any stray per-pane magic numbers.
- **Smoke:** Avatar not cropped; breathing room intact.
- **KGB:** `KGB-YYYY-MM-DD_I05_avatar_pad`

I06 — H02b Slice 2 — DPG bootstrap peel (park-lift)

- **Goal:** Move DPG bootstrap into `ui.dpg_app.run` wrapper without freezes.
- **Microsteps:**
- Mirrors: `_app_gui_entry_impl.py`, `ui/dpg_app.py`, `ui/panes.py`.
- Redirect a **single** callsite to wrapper (signature identical).
- Residue: note remaining callsites; do not change in this bite.
- **Smoke:** GUI opens; autoscroll & avatar fit OK; Dev Tools inject works.
- **KGB:** `KGB-YYYY-MM-DD_I06_bootstrap_peel`

I07 — H02b Slice 3 — Move tick/parse helpers (park-lift)

- **Goal:** Extract one helper behind stable seam without attach-mode drift.
- **Microsteps:**
- Mirrors: `_app_gui_entry_impl.py`, `ui/helpers/gui_loop.py`, `ui/helpers/gui_ingest.py`.
- Extract `_refresh_needed` (only) into `gui_loop.py`; import from entry.
- Residue: list remaining helpers for later bites.
- **Smoke:** Double-window attach OK; Chat/Logs unchanged; no freeze.
- **KGB:** `KGB-YYYY-MM-DD_I07_tick_parse_extract`

I08 — Final wrap & snapshot

- **Goal:** Close sprint cleanly.
- **Microsteps:** run attach smoke; tag snapshot; update Canvas KGB.
- **KGB:** `KGB-YYYY-MM-DD_I08_wrap`

PARKED (reference while executing rails)

- Nothing new beyond items integrated above; this section stays empty unless we re-park a failing step. (reference while executing rails)

- Nothing new beyond items integrated above; this section stays empty unless we re-park a failing step.

Known Issues

- Sleep reply mismatch (CLI vs DevTools → Chat)
- Mojibake characters (`âœ"` , `â†'`) appear instead of proper glyphs
- Header rightmost symbol/separator dangling when no label
- Current gray theme undesired; light-blue requested
- Avatar still slightly zoomed/tight; needs padding tweak
- Weird symbols rendering in Chat (`Flow:` line, arrows)
- CLI vs GUI divergence on "Going to sleep" reply