Heaven Interface Usage Guide

Version: 0.4.0
Last Updated: October 17, 2025
Completion Level: >97%

📚 Table of Contents

Introduction

Heaven Interface is the user interface layer for Solo Git, implementing a minimalist, keyboard-first approach inspired by Dieter Rams and Jony Ive. It provides three complementary interfaces:

- **CLI**: Command-line interface with Rich formatting
- **TUI**: Full-screen text interface with live updates
- **GUI**: Desktop application with visual components

All interfaces share state seamlessly, allowing you to work however you prefer.

Design Principles

1. **As Little Design as Possible**: Only essential elements
2. **Keyboard-First**: Every action accessible via keyboard
3. **Code-Centric**: Code always front and center
4. **Real-Time**: Live updates across all interfaces
5. **Unified State**: Switch between interfaces seamlessly
6. **Fast Startup**: <100ms launch time

Philosophy

Tests Are The Review

Solo Git eliminates traditional code review with:
- ✅ Green tests = auto-merge to trunk
- ❌ Red tests = stay in workpad
- No PRs, no waiting, no ceremony

Single Trunk

- **One branch**: `main` (protected)
- **No branches**: Use ephemeral workpads instead
- **Fast-forward merges**: Clean history

AI Pairing

- **Resident AI**: GPT-4/Claude via Abacus.ai
- **Context-aware**: Understands your codebase
- **Interactive**: Pair programming, not code generation

Getting Started

Installation

```
# Install Solo Git with Heaven Interface
pip install solo-git[heaven]

# Or install from source
git clone https://github.com/yourusername/solo-git
cd solo-git
pip install -e .[heaven]
```

First Run

```
# Configure API credentials
evogitctl config setup

# Initialize a repository
evogitctl repo init --zip myapp.zip

# Launch Heaven Interface
evogitctl heaven

# Or use CLI
evogitctl repo list
```

Quick Tour

```
# Take a 2-minute tour
evogitctl hello           # See welcome message
evogitctl version         # Check installation
evogitctl repo list       # View repositories
evogitctl pad list        # View workpads
evogitctl heaven          # Launch TUI
```

Interface Overview

Three Interfaces, One State

```
┌──────────────┬──────────────┬──────────────┐
│     CLI      │     TUI      │     GUI      │
├──────────────┼──────────────┼──────────────┤
│  Terminal    │  Full-screen│   Desktop    │
│  Commands    │  Interactive│   Visual     │
│  Scriptable │  Live        │   Mouse      │
└──────────────┴──────────────┴──────────────┘
               │
               ├───────────────┐
               │               │
          JSON State      Git Repo
       ~/.sologit/state   ~/.sologit/data
```

When to Use Each

CLI: When you...
- Want to script operations
- Prefer pure terminal workflow
- Need command history
- Are SSH'd into a server

TUI: When you...
- Want live updates
- Need to monitor tests
- Want keyboard navigation
- Prefer full-screen focus

GUI: When you...
- Want visual commit graph
- Need code editing
- Prefer mouse interaction
- Want side-by-side views

CLI Usage

Basic Commands

```
# Repository management
evogitctl repo init --zip app.zip
evogitctl repo list
evogitctl repo info <repo-id>

# Workpad management
evogitctl pad create "add feature"
evogitctl pad list
evogitctl pad info <pad-id>
evogitctl pad promote <pad-id>

# Testing
evogitctl test run <pad-id>
evogitctl test run <pad-id> --target full

# AI pairing
evogitctl pair "add login feature"
evogitctl ai generate "auth module"
evogitctl ai review <pad-id>
```

Rich Formatting

The CLI now uses Rich for beautiful output:

Tables:

```
evogitctl repo list
# Shows:
┌──────────┬──────────┬──────┬──────────┬────────┐
│ ID       │ Name     │ Trunk │ Workpads │ Created │
├──────────┼──────────┼──────┼──────────┼────────┤
│ repo_abc │ MyApp    │ main  │ 3        │ 10/17  │
└──────────┴──────────┴──────┴──────────┴────────┘
```

Panels:

```
evogitctl repo info repo_abc
# Shows:
┌──────────────── Repository: MyApp ────────────────┐
│ Repository: repo_abc                              │
│ Name: MyApp                                       │
│ Path: /home/user/.sologit/data/repos/repo_abc    │
│ Trunk: main                                       │
│ Created: 2025-10-17 10:30:00                      │
│ Workpads: 3 active                                │
│ Source: zip                                       │
└───────────────────────────────────────────────────┘
```

Progress Indicators:

```
evogitctl test run pad_123
# Shows:
┌──────────────── Test Execution ────────────────┐
│ Workpad: Add login feature                     │
│ Tests: 3                                       │
│ Mode: Parallel                                 │
│ Target: fast                                   │
└────────────────────────────────────────────────┘

⠹ Running fast tests... ━━━━━━━━━━━━━━━━ 100%

┌──────── Test Summary ────────┐
│ Total: 3                     │
│ Passed: 3                    │
│ Failed: 0                    │
│ Status: PASSED               │
└──────────────────────────────┘
✅ All tests passed! Ready to promote.
```

Interactive Shell

Launch an enhanced shell with autocomplete:

```
evogitctl interactive

# Features:
# - Tab completion
# - Command history (↑/↓)
# - Fuzzy search (Ctrl+R)
# - Auto-suggest from history
```

TUI Usage

Launching

```
evogitctl heaven              # Production version
evogitctl heaven --repo /path # With specific repo
evogitctl tui                 # Basic version
```

Layout

```
┌──────────────────────────────────────────────────────────────┐
│ Heaven Interface - Solo Git                                  │
├──────────────┬──────────────────┬──────────────────────────┤
│              │                  │                          │
│  Commit      │  Workpad Status  │   Test Runner            │
│  Graph       │                  │                          │
│              │                  │   ✅ unit-tests          │
│  ●────●      │  📝 Add Login    │   ✅ integration         │
│  │    │      │  Status: active  │   ⏳ e2e                 │
│  ●    ●      │  Tests: passed   │                          │
│              │                  │   3/3 passed             │
├──────────────┼──────────────────┼──────────────────────────┤
│              │                  │                          │
│  File        │  AI Activity     │   Diff Viewer            │
│  Tree        │                  │                          │
│              │  🤖 Planning...  │   + new line             │
│  📁 src/     │  Cost: $0.03     │   - old line             │
│  📄 auth     │  Tokens: 1.2k    │   ~ modified             │
│  📄 api      │                  │                          │
│              │                  │                          │
├──────────────┴──────────────────┴──────────────────────────┤
│ 📦 MyApp │ 🔧 Add Login │ ✅ Tests │ Ctrl+P: Commands      │
└──────────────────────────────────────────────────────────────┘
```

Panels

Left Column:
- **Top**: Commit graph with visual history
- **Bottom**: File tree with git status indicators

Center Column:
- **Top**: Current workpad status
- **Bottom**: AI activity log with cost tracking

Right Column:
- **Top**: Real-time test runner
- **Bottom**: Diff viewer for changes

Essential Keys

- `Ctrl+P` : Open command palette
- `?` : Show help with all shortcuts
- `R` : Refresh all panels
- `Ctrl+T` : Run tests
- `Ctrl+Q` : Quit

[See KEYBOARD_SHORTCUTS.md for complete reference]

Command Palette

Press `Ctrl+P` to open:

```
┌──────────────────────────────────┐
│  Command Palette                 │
├──────────────────────────────────┤
│  > test run                      │
│                                  │
│  Results (5):                    │
│                                  │
│  🧪 Test › Run Fast Tests        │
│  🧪 Test › Run Full Tests        │
│  📊 Test › View Results          │
│  📈 Test › View Coverage         │
│  🔄 Test › Rerun Failed          │
└──────────────────────────────────┘
```

Features:
- **Fuzzy matching**: Type partial words
- **Categories**: Commands grouped logically
- **Shortcuts**: Shows keyboard shortcuts
- **Recent**: Shows recently used commands

GUI Usage

Launching

```
# From CLI
evogitctl gui

# Or launch standalone
heaven-gui
```

Features

1. **Monaco Editor**: Full code editing with syntax highlighting
2. **Visual Commit Graph**: Interactive D3/visx visualization
3. **Test Dashboard**: Charts and metrics with Recharts
4. **AI Assistant**: Chat interface for AI pairing
5. **File Browser**: Tree view with git status
6. **Settings Panel**: Configuration UI

Components

Code Viewer (Monaco)
- Syntax highlighting for 50+ languages
- Git diff annotations
- Find/replace
- Multi-cursor editing
- IntelliSense (if available)

Commit Graph
- Visual tree layout
- Click to inspect commits
- Color-coded status (passed/failed)

- Jenkins CI indicators
- Zoom and pan

Test Dashboard
- Pass/fail trends over time
- Test duration charts
- Coverage visualization
- Failed test details

AI Assistant
- Chat interface
- Code suggestions
- Real-time streaming responses
- Cost tracking
- Context-aware

Workflows

Workflow 1: Add a Feature

CLI Way:

```
# Create workpad
evogitctl pad create "add email validation"

# AI generates code
evogitctl pair "add email validation to signup form"

# Review and test
evogitctl pad diff pad_abc
evogitctl test run pad_abc

# Merge if green
evogitctl pad promote pad_abc
```

TUI Way:

```
1. Launch: evogitctl heaven
2. Press: Ctrl+N → Enter title
3. Press: Space → Type prompt
4. Watch: AI generates, tests run
5. Press: Ctrl+M to merge
```

GUI Way:

```
1. Open Heaven GUI
2. File → New Workpad
3. Click AI Assistant → Enter prompt
4. Review changes in diff viewer
5. Click "Run Tests" button
6. Click "Merge to Trunk" if green
```

Workflow 2: Fix a Bug

Quick Fix (CLI):

```
# Jump straight to pairing
evogitctl pair "fix null pointer in auth.login"

# Tests run automatically
# Auto-merges if green
```

Investigative Fix (TUI):

```
1. evogitctl heaven
2. Ctrl+L → View commit history
3. Navigate to bug introduction
4. Ctrl+Z → Revert that commit
5. Ctrl+N → Create fix workpad
6. Fix code manually
7. Ctrl+T → Run tests
8. Ctrl+M → Merge fix
```

Workflow 3: Refactor Code

AI-Assisted (CLI):

```
evogitctl ai refactor "extract common auth logic"
# Or
evogitctl pair "refactor auth module for clarity"
```

Manual (TUI):

```
1. evogitctl heaven
2. Navigate to file in tree
3. Edit in external editor
4. Ctrl+T → Run tests
5. Iterate until green
6. Ctrl+M → Merge
```

Workflow 4: Review History

CLI:

```
evogitctl history log
evogitctl history show <sha>
evogitctl ci status
```

TUI:

```
1. Commit graph panel shows visual history
2. Click on commit node
3. View diff in right panel
4. See test status indicators
```

Workflow 5: AI Pairing Session

Interactive (TUI):

```
1. Press Space
2. Type: "add JWT authentication"
3. Watch AI:
   - Plan the feature
   - Generate code
   - Create tests
   - Run tests
4. Auto-merges if green
```

Advanced Features

State Synchronization

All interfaces share state via JSON files:

```
~/.sologit/state/
├── global.json           # Active repo/workpad
├── repositories/
│   └── repo_*.json       # Repo metadata
├── workpads/
│   └── pad_*.json        # Workpad state
├── test_runs/
│   └── run_*.json        # Test results
└── ai_operations/
    └── op_*.json         # AI activity
```

Changes in CLI appear immediately in TUI/GUI and vice versa.

Command History & Undo

```
# View history
evogitctl history log

# Undo last command
evogitctl history undo

# Redo
evogitctl history redo

# In TUI
Ctrl+Z  # Undo
Ctrl+Y  # Redo
```

History is preserved across sessions.

Fuzzy Search

Both command palette and interactive shell support fuzzy matching:

```
Type: "tml"
Matches:
- Test › Run Fast Tests
- Test › Rerun Failed
- Test › View Results

Type: "prcr"
Matches:
- Pad › Create
- Pair › Create Session
```

Real-Time Updates

The TUI updates automatically when:
- Tests complete
- AI generates code
- State changes from CLI
- Files change on disk

No manual refresh needed.

Cost Tracking

All AI operations track cost:

```
evogitctl ai costs
# Shows:
# Operation    Model        Tokens  Cost
# generate     gpt-4        1,234   $0.03
# review       claude-2     2,456   $0.05
# refactor     gpt-4        3,789   $0.08
#
# Total: $0.16
```

In TUI, see cost in AI Activity panel.

Configuration

Edit `~/.sologit/config.yaml`:

```yaml
solo:
  review: "tests"
  promote_on_green: true
  rollback_on_ci_red: true
  keep_pads_days: 7

tests:
  fast: ["pytest tests/ -q"]
  full: ["pytest tests/ --cov"]

models:
  planner:
    kind: "gpt-4"
    provider: "abacus"
  executor:
    kind: "qwen2.5-coder:3b"
    provider: "ollama"

heaven_interface:
  theme: "dark"
  panels:
    left_width: 30
    center_width: 40
    right_width: 30
  keyboard:
    command_palette: "ctrl+p"
    run_tests: "ctrl+t"

budget:
  daily_usd_cap: 10
```

Troubleshooting

CLI Issues

Problem: “Rich formatting not working”

```bash
# Check if Rich is installed
pip install rich

# Force color output
export FORCE_COLOR=1
evogitctl repo list
```

Problem: “Commands not found”

```bash
# Verify installation
pip install -e .

# Check PATH
which evogitctl
```

TUI Issues

Problem: “TUI won’t launch”

```
# Check Textual installation
pip install textual

# Run with debug
evogitctl heaven --verbose
```

Problem: “Panels not updating”

```
Press: R (refresh)
Or: Ctrl+R (hard refresh)
```

Problem: “Keyboard shortcuts not working”

```
Check: ? for help screen
Verify: No key conflicts in config
Try: Default shortcuts (Ctrl+P, etc.)
```

GUI Issues

Problem: “GUI won’t build”

```
# Frontend build
cd heaven-gui
npm install
npm run build

# Tauri build (requires Rust)
cargo build --release
```

Problem: “State not syncing”

```
# Check state files
ls -la ~/.sologit/state/

# Verify permissions
chmod -R u+rw ~/.sologit/state/
```

State Issues

Problem: “State corruption”

```
# Backup first
cp -r ~/.sologit/state ~/.sologit/state.backup

# Reset state
rm -rf ~/.sologit/state
evogitctl repo list  # Regenerates state
```

Problem: “Undo/redo not working”

```
# Check history
evogitctl history log

# Clear if needed
rm ~/.sologit/state/history.json
```

Tips & Best Practices

💡 CLI Tips

1. **Use interactive mode** for exploratory work
2. **Script repetitive tasks** with CLI commands
3. **Pipe output** to other tools: `evogitctl repo list | grep myapp`
4. **Use –help** liberally: `evogitctl pad --help`

💡 TUI Tips

1. **Learn Ctrl+P first** - it's your entry point to everything
2. **Use R to refresh** when in doubt
3. **Watch the test panel** for real-time feedback
4. **Let it run in background** for monitoring

💡 GUI Tips

1. **Use split view** for code + tests
2. **Keep AI assistant open** for quick queries
3. **Click commit nodes** to inspect history
4. **Use search** to find files quickly

💡 Workflow Tips

1. **Start small**: Create workpad → Change → Test → Merge
2. **Trust the tests**: Green = safe to merge
3. **Use AI liberally**: It's fast and cheap
4. **Commit often**: Workpads are cheap

💡 Performance Tips

1. **Use fast tests first**: Get quick feedback
2. **Run full tests before merge**: Catch edge cases
3. **Keep workpads short-lived**: Merge or delete
4. **Clean up old state**: `evogitctl cleanup`

Next Steps

Learn More

- **Keyboard Shortcuts**: `docs/KEYBOARD_SHORTCUTS.md`
- **Testing Guide**: `docs/TESTING_GUIDE.md`
- **API Reference**: `docs/API.md`

- **Examples**: `examples/` directory

Try Examples

```
cd examples/
./demo_basic.sh      # Basic workflow
./demo_ai.sh         # AI pairing
./demo_testing.sh    # Test-driven workflow
```

Join Community

- GitHub Issues: Report bugs and request features
- Discussions: Ask questions and share workflows
- Contributing: `CONTRIBUTING.md`

Summary

Heaven Interface provides three complementary ways to use Solo Git:

- **CLI**: Fast, scriptable, terminal-native
- **TUI**: Interactive, live updates, full-screen
- **GUI**: Visual, mouse-friendly, feature-rich

All interfaces share state seamlessly. Use whichever fits your workflow.

Core Philosophy:
- Tests are the review
- Single trunk, no branches
- AI-augmented development
- Keyboard-first interface
- Minimalist design

Quick Start:

```
evogitctl config setup
evogitctl repo init --zip app.zip
evogitctl pair "add feature"
```

Get Help:

```
evogitctl --help
evogitctl heaven  # Press ?
```

Heaven Interface - As little design as possible.