

# Terminus Roadmap

This document outlines potential improvements, new features, code cleanup opportunities, and technical debt for the Terminus terminal UI library.

---

## Feature Proposals

### [Priority: High] Mouse Support

**Description:** Add mouse event handling for click, scroll, and drag operations. This would enable interactive widgets like buttons, clickable lists, and drag-to-resize panels.

**Rationale:** Modern terminal UI libraries like ratatui and crossterm provide mouse support. This is essential for building rich interactive applications and would significantly enhance the user experience of terminus-based applications.

**Affected Files:** - /Users/Shared/Projects/lean-workspace/terminus/Terminus/Input/Events.lean (add MouseEvent type) - /Users/Shared/Projects/lean-workspace/terminus/Terminus/Input/Key.lean (extend Event type) - /Users/Shared/Projects/lean-workspace/terminus/ffi/terminus.c (enable mouse tracking) - /Users/Shared/Projects/lean-workspace/terminus/Terminus/Backend/Ansi.lean (mouse escape sequences)

**Estimated Effort:** Medium

**Dependencies:** None

---

### [Priority: High] Window Resize Events

**Description:** Detect and handle terminal window resize events (SIGWINCH) so applications can dynamically adapt to size changes.

**Rationale:** Currently, applications must poll for size changes. Native resize event support would make applications more responsive and efficient.

**Affected Files:** - /Users/Shared/Projects/lean-workspace/terminus/ffi/terminus.c (SIGWINCH handler) - /Users/Shared/Projects/lean-workspace/terminus/Terminus/Input/Events.lean (ResizeEvent type) - /Users/Shared/Projects/lean-workspace/terminus/Terminus/Backend/TerminalEffect.lean

**Estimated Effort:** Medium

**Dependencies:** None

---

### [Priority: Medium] Async Event Polling

**Description:** Implement non-blocking async I/O for event polling using Lean's Task system, allowing applications to perform background work while waiting for input.

**Rationale:** The current synchronous polling model blocks the main thread. Async support would enable more sophisticated applications with background tasks.

**Affected Files:** - /Users/Shared/Projects/lean-workspace/terminus/ffi/terminus.c (select/poll-based reading) - /Users/Shared/Projects/lean-workspace/terminus/Terminus/Input/Events.lean - /Users/Shared/Projects/lean-workspace/terminus/Terminus/Backend/TerminalEffect.lean

**Estimated Effort:** Large

**Dependencies:** None

---

### [Priority: Medium] Cross-Platform Windows Support

**Description:** Add Windows Console API support as an alternative backend to the Unix termios-based implementation.

**Rationale:** Currently terminus only works on Unix-like systems. Windows support would significantly expand the library's utility.

**Affected Files:** - /Users/Shared/Projects/lean-workspace/terminus/ffi/terminus.c (add Windows conditionals) - New file: ffi/terminus\_win.c (Windows-specific implementation) - /Users/Shared/Projects/lean-workspace (conditional compilation)

**Estimated Effort:** Large

**Dependencies:** None

---

### [Priority: Medium] Composable Widget Typeclass

**Description:** Introduce a Widget typeclass with methods for measuring preferred size, handling input events, and rendering. This would enable more sophisticated layout algorithms and event routing.

**Rationale:** The current Widget class only has a render method. A richer interface would enable: - Automatic size calculation for layouts - Event bubbling and focus management - Widget introspection

**Affected Files:** - /Users/Shared/Projects/lean-workspace/terminus/Terminus/Widgets/Widget.lean - All widget files would need to implement new methods

**Estimated Effort:** Large

**Dependencies:** None

---

### [Priority: Medium] Focus Management System

**Description:** Implement a focus management system for navigating between interactive widgets using Tab/Shift+Tab or arrow keys.

**Rationale:** Currently each application must manually track focus state. A centralized focus system would reduce boilerplate and provide consistent behavior across applications.

**Affected Files:** - New file: Terminus/Core/Focus.lean - /Users/Shared/Projects/lean-workspace/terminus/Terminus (focusable trait) - Interactive widgets (TextInput, TextArea, etc.)

**Estimated Effort:** Medium

**Dependencies:** Composable Widget Typeclass

---

### [Priority: Medium] Clipboard Integration

**Description:** Add clipboard read/write support using OSC 52 escape sequences or platform-specific APIs.

**Rationale:** Copy/paste functionality is essential for text input widgets. The TextArea and TextInput widgets would greatly benefit from clipboard integration.

**Affected Files:** - /Users/Shared/Projects/lean-workspace/terminus/Terminus/Backend/Ansi.lean (OSC 52 sequences) - /Users/Shared/Projects/lean-workspace/terminus/Terminus/Widgets/TextInput.lean - /Users/Shared/Projects/lean-workspace/terminus/Terminus/Widgets/TextArea.lean

**Estimated Effort:** Medium

**Dependencies:** None

---

#### [Priority: Medium] **Hyperlink Support**

**Description:** Add support for terminal hyperlinks using OSC 8 escape sequences, allowing clickable links in terminal output.

**Rationale:** Modern terminals support hyperlinks (iTerm2, Windows Terminal, GNOME Terminal). This would enhance the utility of text widgets.

**Affected Files:** - /Users/Shared/Projects/lean-workspace/terminus/Terminus/Backend/Ans1.lean - /Users/Shared/Projects/lean-workspace/terminus/Terminus/Core/Cell.lean (hyperlink attribute) - /Users/Shared/Projects/lean-workspace/terminus/Terminus/Core/Style.lean

**Estimated Effort:** Small

**Dependencies:** None

---

#### [Priority: Medium] **Text Wrapping Modes**

**Description:** Implement multiple text wrapping modes for Paragraph widget: word wrap, character wrap, no wrap with horizontal scroll.

**Rationale:** The current Paragraph implementation has basic text handling. Proper word wrapping would improve text display quality.

**Affected Files:** - /Users/Shared/Projects/lean-workspace/terminus/Terminus/Widgets/Paragraph.lean - /Users/Shared/Projects/lean-workspace/terminus/Terminus/Core/ (new text utilities)

**Estimated Effort:** Medium

**Dependencies:** None

---

#### [Priority: Low] **Markdown Widget**

**Description:** Create a widget that renders basic Markdown (headers, bold, italic, code blocks, lists) with appropriate styling.

**Rationale:** Many applications display documentation or formatted text. A Markdown widget would provide rich text rendering with minimal effort.

**Affected Files:** - New file: Terminus/Widgets/Markdown.lean - /Users/Shared/Projects/lean-workspace/terminus/Terminus/Widgets/Markdown.lean (for styling)

**Estimated Effort:** Medium

**Dependencies:** Text Wrapping Modes

---

#### [Priority: Low] **Notification/Toast Widget**

**Description:** Create a notification/toast widget that displays temporary messages with automatic timeout and stacking.

**Rationale:** Common UI pattern for showing transient feedback to users.

**Affected Files:** - New file: Terminus/Widgets/Notification.lean

**Estimated Effort:** Small

**Dependencies:** None

---

#### [Priority: Low] Progress Bar Variants

**Description:** Add additional progress bar styles: spinner with text, indeterminate progress, multi-step progress.

**Rationale:** The current Gauge widget is good for determinate progress. Additional variants would cover more use cases.

**Affected Files:** - /Users/Shared/Projects/lean-workspace/terminus/Terminus/Widgets/Gauge.lean (or new file)

**Estimated Effort:** Small

**Dependencies:** None

---

#### [Priority: Low] Sixel Graphics Support

**Description:** Add Sixel graphics protocol support for displaying images in compatible terminals.

**Rationale:** While iTerm2 protocol is supported via Image widget, Sixel has broader terminal support (xterm, mlterm, foot).

**Affected Files:** - New file: Terminus/Widgets/Sixel.lean or extend Image widget - /Users/Shared/Projects/lean-workspace/encoding/utilities

**Estimated Effort:** Medium

**Dependencies:** None

---

### Code Improvements

#### [Priority: High] Type-Safe Buffer Indexing

**Current State:** Buffer access in /Users/Shared/Projects/lean-workspace/terminus/Terminus/Core/Buffer.lean uses `cells[idx]!` which can panic on out-of-bounds access.

**Proposed Change:** Use bounds-checked access with proper error handling or proof-carrying code to ensure safe indexing.

**Benefits:** Prevents runtime panics, improves reliability.

**Affected Files:** - /Users/Shared/Projects/lean-workspace/terminus/Terminus/Core/Buffer.lean

**Estimated Effort:** Medium

---

#### [Priority: High] Unicode Width Calculation

**Current State:** Wide characters (CJK, emoji) are treated as single-width in cell positioning, causing display issues.

**Proposed Change:** Implement Unicode width calculation (wcwidth equivalent) and handle double-width characters properly in Buffer operations.

**Benefits:** Correct display of international text and emoji.

**Affected Files:** - /Users/Shared/Projects/lean-workspace/terminus/Terminus/Core/Cell.lean  
- /Users/Shared/Projects/lean-workspace/terminus/Terminus/Core/Buffer.lean - New file: Terminus/Core/Unicode.lean

**Estimated Effort:** Medium

---

#### [Priority: Medium] Style Merging Improvements

**Current State:** Style.merge in /Users/Shared/Projects/lean-workspace/terminus/Terminus/Core/Style.lean has simple override semantics that may not match user expectations.

**Proposed Change:** Implement more nuanced style merging with explicit inheritance rules, possibly using a StyleDiff type.

**Benefits:** More predictable style composition for complex UIs.

**Affected Files:** - /Users/Shared/Projects/lean-workspace/terminus/Terminus/Core/Style.lean

**Estimated Effort:** Small

---

#### [Priority: Medium] Buffer Diff Optimization

**Current State:** The diff function in Buffer.lean iterates through all cells, but could be optimized for common patterns.

**Proposed Change:** Implement dirty region tracking or use a more efficient diffing algorithm (e.g., row-level checksums).

**Benefits:** Reduced rendering overhead for large terminals.

**Affected Files:** - /Users/Shared/Projects/lean-workspace/terminus/Terminus/Core/Buffer.lean

**Estimated Effort:** Medium

---

#### [Priority: Medium] Layout Algorithm Optimization

**Current State:** The layout algorithm in /Users/Shared/Projects/lean-workspace/terminus/Terminus/Layout/Layout recalculates constraints on every split.

**Proposed Change:** Cache constraint calculations and implement incremental layout updates.

**Benefits:** Better performance for complex nested layouts.

**Affected Files:** - /Users/Shared/Projects/lean-workspace/terminus/Terminus/Layout/Layout.lean  
- /Users/Shared/Projects/lean-workspace/terminus/Terminus/Layout/Constraint.lean

**Estimated Effort:** Medium

---

### [Priority: Medium] Escape Sequence Parser State Machine

**Current State:** Escape sequence parsing in `/Users/Shared/Projects/lean-workspace/terminus/Terminus/Input/Events.lean` uses nested pattern matching which is hard to extend.

**Proposed Change:** Implement a proper state machine parser that can handle arbitrary escape sequences, including CSI parameters.

**Benefits:** More robust input handling, easier to add new key sequences.

**Affected Files:** - `/Users/Shared/Projects/lean-workspace/terminus/Terminus/Input/Events.lean`

**Estimated Effort:** Medium

---

### [Priority: Medium] Centralized Error Handling

**Current State:** FFI functions return IO errors but error messages are hardcoded strings.

**Proposed Change:** Define a `TerminalError` inductive type with structured error information.

**Benefits:** Better error handling, testability, and user feedback.

**Affected Files:** - New file: `Terminus/Core/Error.lean` - `/Users/Shared/Projects/lean-workspace/terminus/Terminus/ffi/terminus.c`

**Estimated Effort:** Small

---

### [Priority: Low] Canvas Anti-Aliasing

**Current State:** The BrailleGrid in Canvas uses simple pixel-level rendering without anti-aliasing.

**Proposed Change:** Implement anti-aliased line drawing using sub-pixel coverage calculation.

**Benefits:** Smoother lines in charts and canvas drawings.

**Affected Files:** - `/Users/Shared/Projects/lean-workspace/terminus/Terminus/Widgets/Canvas.lean`

**Estimated Effort:** Medium

---

### [Priority: Low] Float.sin/cos Performance

**Current State:** Several widgets (charts, animations) use `Float.sin/cos` which may be slower than lookup tables for animation use cases.

**Proposed Change:** Implement fast approximate trig functions or lookup tables for animation-quality rendering.

**Benefits:** Better performance for animated UIs.

**Affected Files:** - `/Users/Shared/Projects/lean-workspace/terminus/examples/KitchenSink.lean` - Chart widgets

**Estimated Effort:** Small

---

## Code Cleanup

### [Priority: High] Missing Widget Tests

**Issue:** Tests only cover backend/input functionality. Widget rendering is untested.

**Location:** - /Users/Shared/Projects/lean-workspace/terminus/Tests/Main.lean

**Action Required:** 1. Add test utilities for comparing buffer output 2. Create tests for each widget's rendering logic 3. Add tests for edge cases (empty areas, overflow, etc.)

**Estimated Effort:** Medium

---

### [Priority: Medium] Inconsistent Float Conversion

**Issue:** Various widgets use different patterns for Float to Nat conversion: `toUInt32.toNat`, `Int.ofNat`, direct `.toNat`. This is inconsistent and some conversions may lose precision or behave unexpectedly with negative values.

**Location:** - /Users/Shared/Projects/lean-workspace/terminus/Terminus/Widgets/Canvas.lean  
(lines 250-275) - /Users/Shared/Projects/lean-workspace/terminus/Terminus/Widgets/LineChart.lean  
(lines 227-230) - /Users/Shared/Projects/lean-workspace/terminus/Terminus/Widgets/BarChart.lean  
(line 96)

**Action Required:** 1. Define a consistent helper function for Float to Nat conversion 2. Handle negative values explicitly 3. Apply consistently across all widgets

**Estimated Effort:** Small

---

### [Priority: Medium] Duplicated innerArea Pattern

**Issue:** Every widget has the same boilerplate for handling optional blocks and computing inner areas.

**Location:** All widget files in /Users/Shared/Projects/lean-workspace/terminus/Terminus/Widgets/

**Action Required:** 1. Extract common pattern into a helper function 2. Consider a Widget wrapper that handles block rendering

**Estimated Effort:** Small

---

### [Priority: Medium] Magic Numbers in Widgets

**Issue:** Several widgets contain hardcoded numbers without explanation.

**Location:** - /Users/Shared/Projects/lean-workspace/terminus/Terminus/Widgets/LineChart.lean  
(line 102: `height < 3`) - /Users/Shared/Projects/lean-workspace/terminus/Terminus/Widgets/PieChart.lean  
(line 55: `donut ratio 0.9 max`) - /Users/Shared/Projects/lean-workspace/terminus/Terminus/Widgets/BarChart.lean  
(line 141: `label width 10`)

**Action Required:** 1. Replace magic numbers with named constants 2. Add documentation explaining the rationale for each value

**Estimated Effort:** Small

---

### [Priority: Medium] Documentation Gaps

**Issue:** Many public functions lack documentation comments.

**Location:** - Most widget files - /Users/Shared/Projects/lean-workspace/terminus/Terminus/Core/Buffer.lean  
- /Users/Shared/Projects/lean-workspace/terminus/Terminus/Layout/Layout.lean

**Action Required:** 1. Add doc-strings to all public functions and types 2. Include examples in documentation for key APIs 3. Document expected behavior for edge cases

**Estimated Effort:** Medium

---

### [Priority: Low] Unused Imports

**Issue:** Some files may have unused imports.

**Location:** Various files (requires static analysis)

**Action Required:** 1. Review imports in each file 2. Remove unused imports

**Estimated Effort:** Small

---

### [Priority: Low] Consistent Naming Conventions

**Issue:** Some naming inconsistencies exist: - `withX` vs `setX` for builder methods - `new` vs module-level constructors

**Location:** Throughout widget files

**Action Required:** 1. Establish naming convention in documentation 2. Refactor for consistency where feasible (may require deprecation warnings)

**Estimated Effort:** Small

---

### [Priority: Low] Example Code Organization

**Issue:** KitchenSink.lean is very large (~1700 lines) and could be split into separate files.

**Location:** - /Users/Shared/Projects/lean-workspace/terminus/examples/KitchenSink.lean

**Action Required:** 1. Split into separate demo modules 2. Create a shared state management abstraction

**Estimated Effort:** Small

---

## Architectural Considerations

### [Priority: High] Backend Abstraction

**Issue:** The current backend implementation is tightly coupled to Unix termios. A proper backend abstraction would make it easier to: - Add Windows support - Create test backends - Support alternative rendering targets

**Proposed Change:** Define a `Backend` typeclass with operations for: - Raw mode management - Terminal size queries - Input reading - Output writing

**Affected Files:** - /Users/Shared/Projects/lean-workspace/terminus/Terminus/Backend/Terminal.lean  
- /Users/Shared/Projects/lean-workspace/terminus/Terminus/Backend/TerminalIO.lean - /Users/Shared/Projects

---

### [Priority: Medium] Widget State Management

**Issue:** Interactive widgets (TextInput, TextArea, Tree, etc.) manage their own state but there's no unified pattern for state updates and event handling.

**Proposed Change:** Consider an Elm-like architecture with:  
- Immutable widget state  
- Update functions that return new state  
- Event system for inter-widget communication

---

### [Priority: Medium] Layered Rendering

**Issue:** Popups and overlays are rendered inline with other widgets, which can cause Z-ordering issues.

**Proposed Change:** Implement a layered rendering system with:  
- Background layer (normal widgets)  
- Overlay layer (popups, dropdowns)  
- Top layer (tooltips, notifications)

---

## Testing Improvements

### [Priority: High] Expand Test Coverage

**Current Coverage:** - TerminalEffect mock (raw mode, size, I/O): Well tested  
- Key/escape sequence parsing: Well tested  
- Event polling: Basic tests

**Missing Coverage:** - Widget rendering  
- Layout calculations  
- Style merging  
- Buffer operations  
- Canvas/chart drawing

**Action Required:** 1. Create buffer comparison utilities for widget tests  
2. Add unit tests for each widget  
3. Add property-based tests for layout algorithms

---

### [Priority: Medium] Visual Regression Testing

**Proposed:** Create a system for comparing rendered buffers against golden files, enabling detection of rendering regressions.

---

### [Priority: Low] Performance Benchmarks

**Proposed:** Add benchmarks for:  
- Buffer diff performance  
- Layout calculation speed  
- Widget rendering throughput