# Trellis Roadmap

This document outlines feature proposals, code improvements, and cleanup tasks for the Trellis CSS layout library.

---

## Feature Proposals

### [Priority: High] CSS Flexbox `flex-wrap` Multi-line Layout Improvements

**Description:** The current `flex-wrap` implementation handles basic wrapping but lacks full CSS specification compliance. Items that wrap should properly recalculate line breaks when items shrink, and `wrap-reverse` should reverse the cross-axis direction of lines.

**Rationale:** Multi-line flex layouts are common in responsive designs, and accurate wrapping behavior is essential for real-world use cases.

**Affected Files:** - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Algorithm.lean` (lines 239-280, `partitionIntoLines`)

**Estimated Effort:** Medium

**Dependencies:** None

---

### [Priority: High] CSS Grid `minmax()` Track Sizing

**Description:** While `TrackSize.minmax` exists in the type definition, the resolution logic in `sizeTracksToContent` only partially handles it. Full implementation should clamp track sizes between minimum and maximum values during both intrinsic sizing and fr-unit distribution phases.

**Rationale:** `minmax()` is one of the most commonly used CSS Grid features for responsive layouts.

**Affected Files:** - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Algorithm.lean` (lines 521-527, 549-559) - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Grid.lean` (lines 11-16)

**Estimated Effort:** Medium

**Dependencies:** None

---

### [Priority: High] CSS Grid `repeat()` Function Support

**Description:** Add support for the CSS `repeat(count, track-size)` function in grid template definitions, including `repeat(auto-fill, ...)` and `repeat(auto-fit, ...)` for responsive track generation.

**Rationale:** The `repeat()` function dramatically simplifies grid template definitions and enables responsive grids that automatically adjust column count.

**Affected Files:** - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Grid.lean` (new `RepeatFunction` type and modifications to `GridTemplate`) - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/`
(track initialization logic)

**Estimated Effort:** Large

**Dependencies:** None

---

**[Priority: Medium] CSS Grid Named Lines and Areas**

**Description:** Implement support for named grid lines and `grid-template-areas` for semantic grid definitions. Currently, `GridLine.named` exists but is not processed (returns 0 in `resolveGridLine`).

**Rationale:** Named lines and areas make complex grid layouts more maintainable and readable.

**Affected Files:** - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Grid.lean` (lines 68-73, `GridLine`) - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Algorithm.lean` (lines 457-470, 486-505)

**Estimated Effort:** Large

**Dependencies:** None

---

**[Priority: Medium] Flexbox `order` Property**

**Description:** Add support for the CSS `order` property that allows reordering flex items visually without changing the DOM order.

**Rationale:** The `order` property is commonly used for responsive designs where visual order differs from source order.

**Affected Files:** - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Flex.lean` (add `order` field to `FlexItem`) - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Algorithm.lean` (sort items by order before layout)

**Estimated Effort:** Small

**Dependencies:** None

---

**[Priority: Medium] CSS `aspect-ratio` Property**

**Description:** Add support for the CSS `aspect-ratio` property to maintain width-to-height ratios during layout calculation.

**Rationale:** Aspect ratio constraints are essential for responsive images, videos, and maintaining consistent proportions.

**Affected Files:** - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Types.lean` (add `aspectRatio` to `BoxConstraints`) - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Algorithm.lean` (apply aspect ratio during dimension resolution)

**Estimated Effort:** Medium

**Dependencies:** None

---

**[Priority: Medium] Margin Collapse Support**

**Description:** Implement CSS margin collapsing behavior where adjacent vertical margins combine into a single margin equal to the larger of the two.

**Rationale:** Margin collapse is a fundamental CSS behavior that affects vertical spacing between elements.

**Affected Files:** - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Algorithm.lean` (new margin collapse logic) - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Types.lean` (possibly add `marginCollapse` option)

**Estimated Effort:** Medium

**Dependencies:** None

---

[**Priority: Medium**] **CSS Grid `subgrid` Support**

**Description:** Implement `subgrid` to allow nested grids to participate in their parent's track sizing.

**Rationale:** Subgrid enables consistent alignment across nested grid structures.

**Affected Files:** - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Grid.lean` (new `Subgrid` track type) - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Algorithm.lean` (subgrid resolution logic)

**Estimated Effort:** Large

**Dependencies:** None

---

[**Priority: Low**] **Flexbox `visibility: collapse` Support**

**Description:** Support the `visibility: collapse` behavior for flex items where the item is hidden but its cross-size contribution to the line is preserved.

**Rationale:** Useful for implementing togglable content without layout shifts.

**Affected Files:** - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Flex.lean` (add visibility property) - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Algorithm.lean`

**Estimated Effort:** Small

**Dependencies:** None

---

[**Priority: Low**] **Debug/Introspection API**

**Description:** Add an API to inspect intermediate layout state (flex lines, track sizes, item measurements) for debugging purposes.

**Rationale:** Debugging layout issues is challenging without visibility into intermediate calculations.

**Affected Files:** - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Algorithm.lean` (new debug result types) - New file: `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Debug.lean`

**Estimated Effort:** Medium

**Dependencies:** None

---

[**Priority: Low**] **Layout Caching/Memoization**

**Description:** Implement layout caching to avoid recomputing layouts for unchanged subtrees.

**Rationale:** Performance optimization for large layout trees with incremental updates.

**Affected Files:** - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Node.lean` (add cache key/hash) - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Algorithm.lean` (caching logic)

**Estimated Effort:** Large

**Dependencies:** None

---

## Code Improvements

**[Priority: High] Replace Magic Number for Unbounded Max**

**Current State:** In `collectFlexItems` (Algorithm.lean line 201), unbounded max constraints use a magic number `1000000.0`.

**Proposed Change:** Define a constant `Float.infinity` or `Length.unbounded` and use it consistently throughout the codebase.

**Benefits:** Improved clarity, easier to audit for overflow issues, more idiomatic.

**Affected Files:** - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Algorithm.lean` (lines 201-203)

**Estimated Effort:** Small

---

**[Priority: High] Iterative Flex Resolution for Constraint Compliance**

**Current State:** The flex grow/shrink distribution (lines 289-346) uses a single-pass algorithm with a comment noting it lacks "iterative constraint handling."

**Proposed Change:** Implement the full CSS Flexbox algorithm that iteratively freezes items that violate min/max constraints and redistributes remaining space.

**Benefits:** Correct behavior per CSS Flexbox specification when items hit their min/max constraints during flex resolution.

**Affected Files:** - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Algorithm.lean` (lines 282-346, `distributeGrowth`, `distributeShrinkage`)

**Estimated Effort:** Medium

---

**[Priority: Medium] Separate Flexbox and Grid into Distinct Modules**

**Current State:** All layout algorithms are in a single 959-line `Algorithm.lean` file.

**Proposed Change:** Split into `FlexAlgorithm.lean` and `GridAlgorithm.lean` with shared utilities in a `LayoutUtils.lean` module.

**Benefits:** Better separation of concerns, easier navigation, more focused testing.

**Affected Files:** - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Algorithm.lean` (split into multiple files)

**Estimated Effort:** Medium

---

**[Priority: Medium] Use StateM Monad for Mutable Layout State**

**Current State:** Layout algorithms use `Id.run do` with explicit `let mut` patterns for mutable state.

**Proposed Change:** Consider using `StateM` or `StateT` for cleaner accumulation patterns, especially in grid auto-placement and track sizing.

**Benefits:** More idiomatic Lean 4 code, easier to compose with other effects if needed.

**Affected Files:** - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Algorithm.lean`

**Estimated Effort:** Medium

---

[**Priority: Medium**] **Type-Safe Node IDs**

**Current State:** Node IDs are raw `Nat` values that can be confused with other numeric values.

**Proposed Change:** Create an opaque `NodeId` type to prevent accidental misuse of numeric values as node identifiers.

**Benefits:** Compile-time prevention of ID misuse, clearer API contracts.

**Affected Files:** - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Node.lean` (lines 78-85) - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Result.lean` - `/Users/Shared/Projects/lean-workspac`

**Estimated Effort:** Small

---

[**Priority: Medium**] **Improve LayoutResult Lookup Performance**

**Current State:** `LayoutResult.get` uses `Array.find?` which is O(n) lookup (line 107-108).

**Proposed Change:** Use a `HashMap NodeId ComputedLayout` or maintain sorted array for binary search.

**Benefits:** O(1) or O(log n) lookups instead of O(n) for large layout trees.

**Affected Files:** - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Result.lean` (lines 98-141)

**Estimated Effort:** Small

---

[**Priority: Medium**] **Add Proper Baseline Alignment**

**Current State:** Baseline alignment for both `AlignItems.baseline` and grid alignment is simplified as `flexStart` with comments indicating this (Flex.lean line 52, Algorithm.lean lines 415-416, 637, 644).

**Proposed Change:** Implement proper baseline calculation based on first text baseline of items.

**Benefits:** Correct text alignment across flex items with different font sizes.

**Affected Files:** - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Flex.lean` - `/Users/Shared/Projects/` - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Node.lean` (add baseline info to `ContentSize`)

**Estimated Effort:** Medium

**Dependencies:** Requires text measurement integration

---

[**Priority: Low**] **Remove Inhabited Instance Boilerplate**

**Current State:** Multiple structures define default values both as an `Inhabited` instance and as explicit `default`/`empty`/`zero` functions.

**Proposed Change:** Use `deriving Inhabited` with `@[default_instance]` where possible, or consolidate to a single source of truth.

**Benefits:** Less redundant code, single source of truth for defaults.

**Affected Files:** - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Flex.lean` (lines 79, 109) - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Grid.lean` (lines 51, 144) - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Types.lean`

**Estimated Effort:** Small

---

[**Priority: Low**] **Use `partial` Strategically with Termination Proofs**

**Current State:** `layoutNode` is marked `partial` without a termination proof (line 891). Similarly, `nodeCount` and `allIds` in Node.lean (lines 221-226).

**Proposed Change:** Add `termination_by` clauses or refactor to use `Nat.rec` patterns to prove termination.

**Benefits:** Stronger correctness guarantees, avoidance of potential infinite loops.

**Affected Files:** - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Algorithm.lean` (line 891) - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Node.lean` (lines 221-226)

**Estimated Effort:** Medium

---

[**Priority: Low**] **Add `@[inline]` Annotations for Performance-Critical Functions**

**Current State:** Many small helper functions like `AxisInfo.mainSize`, `EdgeInsets.horizontal`, etc. are not annotated for inlining.

**Proposed Change:** Add `@[inline]` or `@[always_inline]` to frequently-called small functions.

**Benefits:** Better runtime performance by avoiding function call overhead.

**Affected Files:** - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Axis.lean` - `/Users/Shared/Projects/`

**Estimated Effort:** Small

---

## Code Cleanup

[**Priority: High**] **Unused Parameter Warning in `autoPlaceItem`**

**Issue:** The `_flow` parameter in `autoPlaceItem` (line 649) is prefixed with underscore indicating it's unused, but `GridAutoFlow` should affect placement behavior (row vs column major, dense packing).

**Location:** `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Algorithm.lean`, line 649

**Action Required:** Either implement flow-aware placement or document why it's not implemented yet.

**Estimated Effort:** Small (for TODO comment) or Medium (for implementation)

---

[**Priority: Medium**] **Consistent Error Handling in `get!` Functions**

**Issue:** `LayoutResult.get!` uses `panic!` (line 113-114) which is not recoverable. Consider returning `Option` or using an error monad.

**Location:** `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Result.lean`, lines 111-114

**Action Required:** Either remove the `get!` function, rename to make the panic clearer (e.g., `getOrPanic`), or convert to a proper error type.

**Estimated Effort:** Small

---

**[Priority: Medium] Add Documentation Comments to Public API**

**Issue:** While there are some doc comments, many public functions lack comprehensive documentation. For example, the main `layout` function (line 955-956) has no doc comment explaining parameters and return value.

**Location:** Throughout all files, particularly: - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Algorithm.l`
- `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Node.lean`

**Action Required:** Add `/-!` section comments and `/--` doc comments to all public functions.

**Estimated Effort:** Medium

---

**[Priority: Medium] Consolidate Test Helper Functions**

**Issue:** `floatNear` and `shouldBeNear` in tests duplicate functionality that could be in the test framework (Crucible).

**Location:** `/Users/Shared/Projects/lean-workspace/trellis/TrellisTests/Main.lean`, lines 16-22

**Action Required:** Either move to Crucible as reusable assertions or keep but add to a shared test utilities module.

**Estimated Effort:** Small

---

**[Priority: Low] Remove Redundant `namespace/end` Pairs**

**Issue:** Some small namespaces could be combined or simplified. For example, `ContainerKind` and `ItemKind` in Node.lean have very few members each.

**Location:** `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Node.lean`

**Action Required:** Consider combining related types into a single namespace or using dot notation directly.

**Estimated Effort:** Small

---

**[Priority: Low] Standardize Naming Conventions**

**Issue:** Some inconsistencies in naming: - `finish` vs `end` (GridSpan uses `finish` to avoid keyword, but could use backticks) - `mk'` constructors vs named constructors - `fromSizes` vs `pixels` (different patterns for similar functionality)

**Location:** - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Grid.lean` (line 78, `finish`) - `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Result.lean` (line 21, `mk'`)
- `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Grid.lean` (lines 53-63)

**Action Required:** Establish and document naming conventions, then apply consistently.

**Estimated Effort:** Small

---

**[Priority: Low] Add Property-Based Tests**

**Issue:** Current tests are example-based. Property-based tests would provide better coverage for edge cases.

**Location:** `/Users/Shared/Projects/lean-workspace/trellis/TrellisTests/Main.lean`

**Action Required:** Add property-based tests using plausible or similar library. Examples: - "Total width of flex items equals container width when all items have grow $> 0$" - "Grid items never overlap" - "Layout positions are always non-negative"

**Estimated Effort:** Medium

**Dependencies:** Property-based testing library (plausible)

---

**[Priority: Low] Clean Up Import Structure**

**Issue:** `Algorithm.lean` imports all other modules but some imports may be transitively satisfied.

**Location:** `/Users/Shared/Projects/lean-workspace/trellis/Trellis/Algorithm.lean`, lines 5-10

**Action Required:** Review import graph and minimize direct imports to only what's directly needed.

**Estimated Effort:** Small

---

## Testing Improvements

**[Priority: High] Add Tests for Edge Cases**

**Issue:** Missing test coverage for: - Negative margins - Zero-width/height containers - Very large numbers of children - Deeply nested containers - Mixed flex/grid nesting

**Location:** `/Users/Shared/Projects/lean-workspace/trellis/TrellisTests/Main.lean`

**Action Required:** Add comprehensive edge case test suite.

**Estimated Effort:** Medium

---

**[Priority: Medium] Add Performance Benchmarks**

**Issue:** No benchmarks to track layout performance over time.

**Location:** New file(s) in TrellisTests

**Action Required:** Create benchmark suite measuring layout time for various tree sizes and complexities.

**Estimated Effort:** Medium

---

**[Priority: Medium] Add Visual Regression Tests**

**Issue:** It's hard to verify that layout results "look correct" without visual output.

**Location:** New test infrastructure

**Action Required:** Create ASCII art or image-based visual tests that render layouts for visual verification.

**Estimated Effort:** Medium

---

## Architecture Considerations

### Dependency Injection for Content Measurement

**Current State:** `getContentSize` only looks at the `content` field of nodes. Real applications need to measure text and other dynamic content.

**Recommendation:** Define a `ContentMeasurer` typeclass or pass a measurement function to `layout`. This aligns with how arbor handles text measurement.

### Integration with Widget Systems

**Current State:** Trellis is used by arbor and afferent. The integration point is primarily through `LayoutNode` construction.

**Recommendation:** Consider a builder API or DSL to make tree construction more ergonomic:

```
layout do
  row (gap := 10) {
    leaf (width := 100, height := 50)
    column {
      leaf (flexGrow := 1)
      leaf (flexGrow := 2)
    }
  }
```

### Consider CSS Variables/Custom Properties

**Long-term:** For theming support, consider how CSS custom properties could be represented and resolved during layout.

---

## Summary

| Category | High | Medium | Low |
|---|---|---|---|
| Features | 3 | 5 | 3 |
| Improvements | 2 | 5 | 3 |
| Cleanup | 2 | 3 | 4 |
| Testing | 1 | 2 | 0 |

**Key Priorities:** 1. Complete CSS Flexbox iterative constraint resolution 2. Implement full `minmax()` track sizing for Grid 3. Add `repeat()` function support for responsive grids 4. Split Algorithm.lean into focused modules 5. Replace magic numbers with named constants