

Chroma Roadmap

A sophisticated color picker for artists and web developers.

Code Improvements

[Priority: High] Extract Magic Numbers and Constants

Current State: The codebase contains hardcoded magic numbers scattered throughout: - `6.283185307179586` (`2*pi`) appears multiple times in `ColorPicker.lean` - Screen scale multipliers like `24 * screenScale`, `32 * screenScale` are inline - Font size calculations `28 * screenScale`, `16 * screenScale` are hardcoded - Widget ID comment `-- Widget IDs (build order): 0: column root, 1: title text...`

Proposed Change: - Define `twoPi` or `tau` constant in a shared location - Create a `Theme` or `Sizes` structure for UI constants - Use semantic names like `titleFontSize`, `bodyFontSize`, `defaultPadding`

Benefits: Improved maintainability, easier theming, self-documenting code

Affected Files: - `/Users/Shared/Projects/lean-workspace/chroma/Chroma/ColorPicker.lean` (lines 54-55, 104, 128, 158, 163) - `/Users/Shared/Projects/lean-workspace/chroma/Chroma/Main.lean` (lines 29-30, 39, 43, 204)

Estimated Effort: Small

[Priority: High] Improve Widget ID Management

Current State: Widget IDs are manually tracked via comments:

```
-- Widget IDs (build order):  
-- 0: column root  
-- 1: title text  
-- 2: color picker  
-- 3: subtitle text  
UIBuilder.register 2 (pickerHandler config)
```

Proposed Change: - Use named widgets via Arbor's `namedCustom` and lookup by name - Alternatively, capture widget ID from builder and use it directly - Consider adding a `colorPickerWidget` function that returns its ID

Benefits: Less fragile code, no need to manually track build order, fewer bugs when UI changes

Affected Files: - `/Users/Shared/Projects/lean-workspace/chroma/Chroma/ColorPicker.lean` (lines 200-215)

Estimated Effort: Small

[Priority: High] Separate Model from UI Configuration

Current State: `PickerModel` only tracks hue and drag state, while `ColorPickerConfig` mixes rendering config with state:

```
structure PickerModel where  
  hue : Float := 0.08  
  dragging : Bool := false
```

```
structure ColorPickerConfig where
```

```
selectedHue : Float := 0.08 -- Duplicated from model!
selectedSaturation : Float := 1.0
selectedValue : Float := 1.0
```

Proposed Change: - Expand `PickerModel` to include saturation, value, and other state - Make `ColorPickerConfig` purely about rendering (sizes, colors, segments) - Pass model values to config at render time instead of duplicating

Benefits: Single source of truth for color state, cleaner separation of concerns

Affected Files: - `/Users/Shared/Projects/lean-workspace/chroma/Chroma/ColorPicker.lean` (structures at lines 15-49) - `/Users/Shared/Projects/lean-workspace/chroma/Chroma/Main.lean` (lines 36-46)

Estimated Effort: Small

[Priority: Medium] Add Comprehensive Type Aliases

Current State: Raw `Float` used everywhere for different semantic meanings (angles, positions, sizes, hue values).

Proposed Change: - Define type aliases: `abbrev Hue := Float`, `abbrev Radians := Float`, `abbrev Degrees := Float` - Consider using Lean's units-of-measure patterns for stronger type safety - Use `Point` from `Arbor` consistently instead of separate x/y floats

Benefits: Self-documenting code, potential for compile-time unit checking

Affected Files: - `/Users/Shared/Projects/lean-workspace/chroma/Chroma/ColorPicker.lean` (throughout)

Estimated Effort: Medium

[Priority: Medium] Modularize Geometry Functions

Current State: Geometry utilities (`circlePoints`, `ringSegmentPoints`, `orientedRectPoints`) are defined inline in `ColorPicker.lean`.

Proposed Change: - Move to `Chroma/Geometry.lean` module - Consider contributing generic versions to `Arbor` if useful there - Add documentation and unit tests for these functions

Benefits: Better code organization, reusable geometry utilities, testable in isolation

Affected Files: - `/Users/Shared/Projects/lean-workspace/chroma/Chroma/ColorPicker.lean` (lines 51-91)

Estimated Effort: Small

[Priority: Medium] Use Tincture Color Type Directly

Current State: Creating colors via `Color.hsv hue 1.0 1.0` in rendering code.

Proposed Change: - Store selected color as `Tincture.Color` in model - Leverage `Tincture`'s harmony, format, and conversion functions - Use `Tincture`'s HSV type for intermediate calculations

Benefits: Full access to `Tincture`'s color manipulation, consistent color handling

Affected Files: - `/Users/Shared/Projects/lean-workspace/chroma/Chroma/ColorPicker.lean` - `/Users/Shared/Projects/lean-workspace/chroma/Chroma/Main.lean`

Estimated Effort: Medium

Code Cleanup

[Priority: High] Expand Test Coverage

Issue: Tests are minimal placeholder tests:

```
test "placeholder" :=  
  ensure true "sanity check"
```

Location: /Users/Shared/Projects/lean-workspace/chroma/ChromaTests/Main.lean (lines 24-25)

Action Required: - Add tests for `hueFromPoint` and `hueFromPosition` functions - Add tests for `circlePoints`, `ringSegmentPoints`, `orientedRectPoints` - Add property-based tests using `Plausible` (already a dependency) - Test edge cases: zero-size picker, boundary hits, angle wraparound

Estimated Effort: Medium

[Priority: Medium] Add Module Documentation

Issue: Module-level documentation exists but function documentation is sparse.

Location: All source files

Action Required: - Add docstrings to public functions - Document expected ranges (e.g., hue is 0.0-1.0, not 0-360) - Add examples in docstrings for key functions

Estimated Effort: Small

[Priority: Medium] Remove Hardcoded Font Path

Issue: Font path is hardcoded to system location:

```
let titleFont <- Font.load "/System/Library/Fonts/Monaco.ttf" ...
```

Location: /Users/Shared/Projects/lean-workspace/chroma/Chroma/Main.lean (lines 29-30)

Action Required: - Accept font path as configuration or command-line argument - Fall back to bundled font if system font unavailable - Consider embedding a default font or using Afferent's font discovery

Estimated Effort: Small

[Priority: Low] Add test.sh Script

Issue: No `test.sh` script like other projects have; testing requires manual command.

Location: Project root (missing file)

Action Required: - Create `test.sh` that runs `./build.sh chroma_tests && .lake/build/bin/chroma_tests` - Mirror pattern from afferent and other sibling projects

Estimated Effort: Small

Feature Proposals

[Priority: High] Add Saturation/Value Picker

Description: Implement the inner triangle or square picker for selecting saturation and value at the current hue.

Rationale: A hue wheel alone is not sufficient for a functional color picker. Users need to select saturation and value/lightness as well.

Affected Files: - /Users/Shared/Projects/lean-workspace/chroma/Chroma/ColorPicker.lean (new widget or extension) - /Users/Shared/Projects/lean-workspace/chroma/Chroma/Main.lean

Estimated Effort: Large

Dependencies: None (Arbor custom widget support already exists)

[Priority: High] Display Selected Color Value

Description: Show the currently selected color as hex, RGB, and HSL text below the picker.

Rationale: Users need to see and copy the color value they've selected. Tincture already provides `toHex`, `toRgbString`, `toHslString`.

Affected Files: - /Users/Shared/Projects/lean-workspace/chroma/Chroma/ColorPicker.lean - /Users/Shared/Projects/lean-workspace/chroma/Chroma/Main.lean

Estimated Effort: Small

Dependencies: None (Tincture.Format already available)

[Priority: High] Add Color Harmony Display

Description: Show complementary, triadic, and analogous colors based on selected hue.

Rationale: Tincture provides full harmony generation (`Color.harmony`, `Color.harmonyOk`). This is a core feature for artists and designers.

Affected Files: - New file: /Users/Shared/Projects/lean-workspace/chroma/Chroma/Harmony.lean - /Users/Shared/Projects/lean-workspace/chroma/Chroma/Main.lean

Estimated Effort: Medium

Dependencies: None (Tincture.Harmony already available)

[Priority: Medium] Add Contrast Checker

Description: WCAG contrast ratio display between selected color and a reference (black/white).

Rationale: Accessibility checking is crucial for web development. Tincture provides `contrastRatio`, `meetsWCAG_AA`, `meetsWCAG_AAA`.

Affected Files: - New file: /Users/Shared/Projects/lean-workspace/chroma/Chroma/Contrast.lean - /Users/Shared/Projects/lean-workspace/chroma/Chroma/Main.lean

Estimated Effort: Medium

Dependencies: None (Tincture.Contrast already available)

[Priority: Medium] Add Color Blindness Simulation

Description: Toggle to preview how the selected color appears under various color vision deficiencies.

Rationale: Critical for accessible design. Tincture provides `simulateColorBlindness` for protanopia, deuteranopia, tritanopia, etc.

Affected Files: - `/Users/Shared/Projects/lean-workspace/chroma/Chroma/ColorPicker.lean` - `/Users/Shared/Projects/lean-workspace/chroma/Chroma/Main.lean`

Estimated Effort: Medium

Dependencies: None (Tincture.Blindness already available)

[Priority: Medium] Add Hex Input Field

Description: Text input for entering colors in hex format (`#RRGGBB`).

Rationale: Users often have a specific color code they want to visualize. Tincture provides `Color.fromHex` for parsing.

Affected Files: - `/Users/Shared/Projects/lean-workspace/chroma/Chroma/Main.lean`

Estimated Effort: Medium

Dependencies: Requires text input widget from Arbor or custom implementation

[Priority: Medium] Add Named Color Picker

Description: Dropdown or grid showing the 140+ CSS named colors from Tincture.

Rationale: Quick access to standard colors. Tincture provides `Named.fromName` and the full named color list.

Affected Files: - New file: `/Users/Shared/Projects/lean-workspace/chroma/Chroma/NamedColors.lean` - `/Users/Shared/Projects/lean-workspace/chroma/Chroma/Main.lean`

Estimated Effort: Medium

Dependencies: None (Tincture.Named already available)

[Priority: Low] Add Gradient Builder

Description: UI for creating multi-stop gradients using the selected colors.

Rationale: Gradients are essential for design work. Tincture provides `Gradient` with multiple interpolation spaces.

Affected Files: - New file: `/Users/Shared/Projects/lean-workspace/chroma/Chroma/Gradient.lean`

Estimated Effort: Large

Dependencies: Saturation/Value picker should be implemented first

[Priority: Low] Add Palette Management

Description: Save, load, and manage color palettes.

Rationale: Users want to collect and organize colors for projects.

Affected Files: - New file: `/Users/Shared/Projects/lean-workspace/chroma/Chroma/Palette.lean` -
New file: `/Users/Shared/Projects/lean-workspace/chroma/Chroma/Storage.lean`

Estimated Effort: Large

Dependencies: Basic color picker should be complete first

[Priority: Low] Add Export Functionality

Description: Export selected color or palette to various formats (CSS, JSON, Swift).

Rationale: Users need to use colors in their projects. `Tinture.Format` provides multiple output formats.

Affected Files: - New file: `/Users/Shared/Projects/lean-workspace/chroma/Chroma/Export.lean`

Estimated Effort: Medium

Dependencies: Basic color picker and palette management

Architectural Improvements

[Priority: Medium] Introduce Application State Management

Current State: State is minimal (`PickerModel` with just hue and dragging).

Proposed Change: - Design a comprehensive `AppState` with: - Current color (HSV and RGB representations) - UI mode (picker, harmony, palette, etc.) - History for undo/redo - Saved palettes - Consider using Collimator lenses for nested state updates

Benefits: Foundation for complex features, undo/redo support, state persistence

Affected Files: - New file: `/Users/Shared/Projects/lean-workspace/chroma/Chroma/State.lean` - All existing files

Estimated Effort: Large

[Priority: Medium] Split UI into Composable Components

Current State: `pickerUI` function builds entire UI in one place.

Proposed Change: - Create separate widget components: - `hueWheel : HueWheelConfig -> WidgetBuilder` - `colorPreview : Color -> WidgetBuilder` - `colorSliders : Color -> WidgetBuilder` - `harmonyDisplay : Color -> HarmonyType -> WidgetBuilder` - Use Arbor's composable widget pattern

Benefits: Reusable components, easier testing, cleaner code organization

Affected Files: - `/Users/Shared/Projects/lean-workspace/chroma/Chroma/ColorPicker.lean` (split into multiple files)

Estimated Effort: Medium

[Priority: Low] Add Keyboard Navigation

Current State: Only mouse interaction is supported.

Proposed Change: - Arrow keys to adjust hue/saturation/value - Tab to move between components - Enter to confirm, Escape to cancel - Number keys for quick hue jumps (1-9 for 10%-90% around the wheel)

Benefits: Accessibility, power-user efficiency

Affected Files: - /Users/Shared/Projects/lean-workspace/chroma/Chroma/ColorPicker.lean - /Users/Shared/Projects/lean-workspace/chroma/Chroma/Main.lean

Estimated Effort: Medium

Dependencies: Afferent keyboard event support (already exists)

Milestones

v0.1 - Foundation (Current + Near-term)

- ☒ Basic hue wheel with drag interaction
- ☒ Color preview in center
- ☐ Saturation/value picker (triangle or square)
- ☐ Hex/RGB/HSL text display
- ☐ Code cleanup: constants, widget IDs, test coverage

v0.2 - Harmony

- ☐ Color harmony visualization
- ☐ 5-color palette view
- ☐ Basic contrast checker

v0.3 - Professional

- ☐ Full accessibility suite (contrast, color blindness)
- ☐ Named colors picker
- ☐ Hex input field

v0.4 - Complete

- ☐ Palette management
- ☐ Export functionality
- ☐ Gradient builder

v1.0 - Release

- ☐ Polished UI with theming
 - ☐ Keyboard navigation
 - ☐ Comprehensive documentation
-

Tincture Features Available for Immediate Use

The following Tincture capabilities can be leveraged without any library changes:

Feature	Module	Key Functions
10 color spaces	<code>Tincture.Space.*</code>	HSL, HSV, HWB, OkLab, OkLCH, Lab, LCH, XYZ, CMYK, Linear RGB
Color harmony	<code>Tincture.Harmony</code>	<code>complementary</code> , <code>triadic</code> , <code>analogous</code> , <code>harmony</code> , <code>harmony0k</code>
WCAG contrast	<code>Tincture.Contrast</code>	<code>contrastRatio</code> , <code>meetsWCAG_AA</code> , <code>meetsWCAG_AAA</code> , <code>apcaContrast</code>
Color blindness	<code>Tincture.Blindness</code>	<code>simulateColorBlindness</code> , <code>isDistinguishableFor</code>
Delta E distance	<code>Tincture.Distance</code>	Color perceptual distance
Named colors	<code>Tincture.Named</code>	140+ CSS named colors
Blend modes	<code>Tincture.Blend</code>	<code>multiply</code> , <code>screen</code> , <code>overlay</code> , etc.
Gradients	<code>Tincture.Gradient</code>	Multi-stop gradients with various interpolation spaces
Color adjustment	<code>Tincture.Adjust</code>	<code>lighten</code> , <code>darken</code> , <code>saturate</code> , <code>rotateHue</code>
Formatting	<code>Tincture.Format</code>	<code>toHex</code> , <code>toRgbString</code> , <code>toHslString</code> , <code>toCssString</code>
Parsing	<code>Tincture.Parse</code>	<code>fromHex</code> , <code>fromCss</code>
Palettes	<code>Tincture.Palette</code>	<code>sequential</code> , <code>diverging</code> , <code>qualitative</code> , <code>accessible</code> palettes

Dependencies on Sibling Libraries

Chroma Feature	Library	Requirement Status
Hue wheel rendering	Arbor	Available (custom widget)
Color manipulation	Tincture	Available (full feature set)
CSS layout	Trellis	Available (flexbox, grid)
GPU rendering	Afferent	Available (Metal backend)
Text input	Arbor/Afferent	Needs implementation
State management	Collimator	Available (optics library)