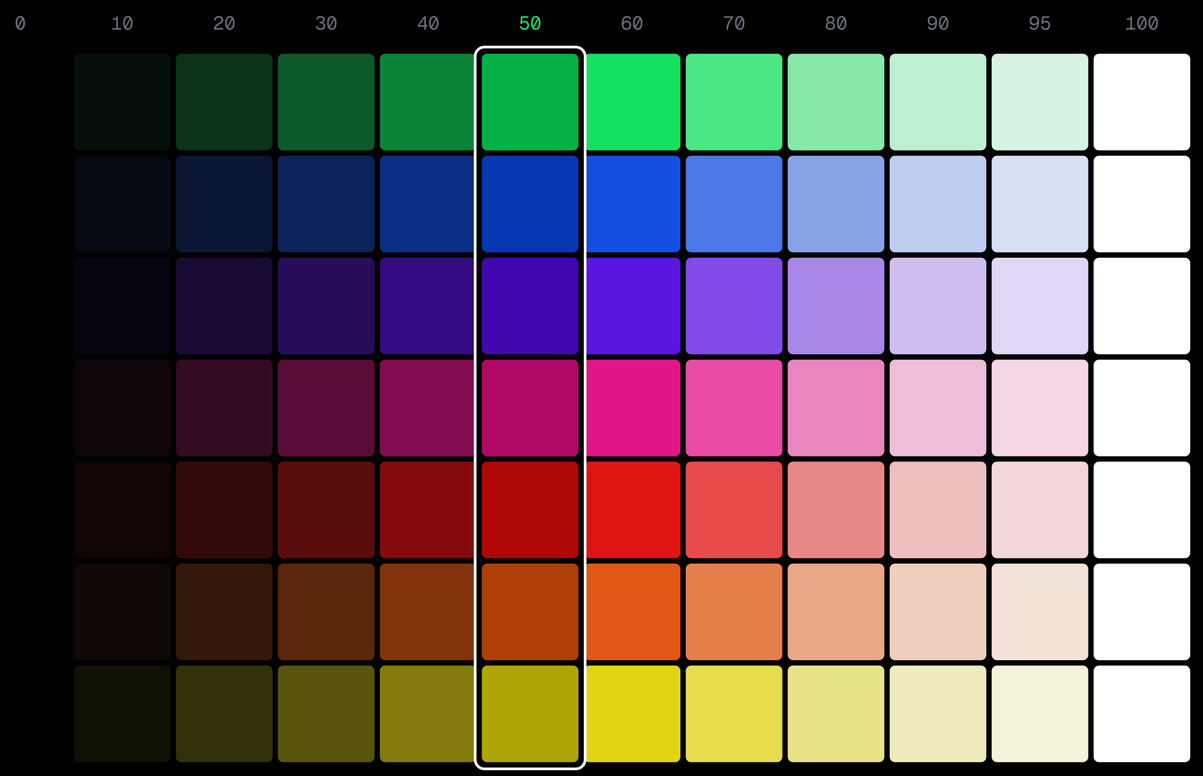
The Greenlit Color System

How a formula-driven palette and intent-named tokens made color consistent, legible to AI, and ready to scale.



Maximillian Grasselino
Founding Design Engineer, Greenlit

Color was the first thing to drift.

Greenlit is a professional platform for film and TV production talent, built around discovery, networking, and agentic talent matching. As its founding design engineer, I design the interface and ship it, often with AI coding tools in the loop.

PROBLEM 01

Values chosen screen by screen

Greens and grays were picked as each screen was built, so the same role could render in slightly different values from page to page.

PROBLEM 02

No shared vocabulary

Design referred to colors by how they looked. Code referred to them by hex. Neither said when a color should be used.

PROBLEM 03

A new kind of teammate

AI coding tools were writing UI next to me. Without named intent, they reached for whichever hex value was nearby.

Three requirements shaped every decision.

01

Consistency

The same role always resolves to the same value, and every hue behaves identically at the same stop.

THE TEST

Swap the hue, keep the hierarchy.

02

AI legibility

A token carries enough intent in its name and description that a person or an agent can choose it without seeing the palette.

THE TEST

Pick the right token from the name alone.

03

Scalability

New hues, states, and themes extend the system instead of forking it.

THE TEST

Add a color family by setting one value.

One formula generates every hue.

HUE RAMP FORMULA / HSL

```
n = (stop - 50) / 10

L = L50 + 12 * n     (n ≥ 0, light side)
L = L50 + 8 * n      (n < 0, dark side)
S = S50 - 8 * |n|
H = H50
A = A50

clamp L and S to 0-100
```

ANCHOR

Each ramp starts from one approved value at stop 50. Emerald 50 is #07B045.

LIGHTNESS

Lightness rises 12 points per stop toward white and falls 8 per stop toward black, which gives the dark end finer steps.

SATURATION

Saturation eases off 8 points per stop away from the anchor, so tints stay clean and shades stay rich.

LOCKED

Hue and alpha never move. All 12 stops read as one color family.



Seven hues. One tonal structure.

Every hue ramp shares the same 12 stops. Ruby 30 carries the same visual weight as Sapphire 30, so status colors can be swapped without rebalancing a layout.

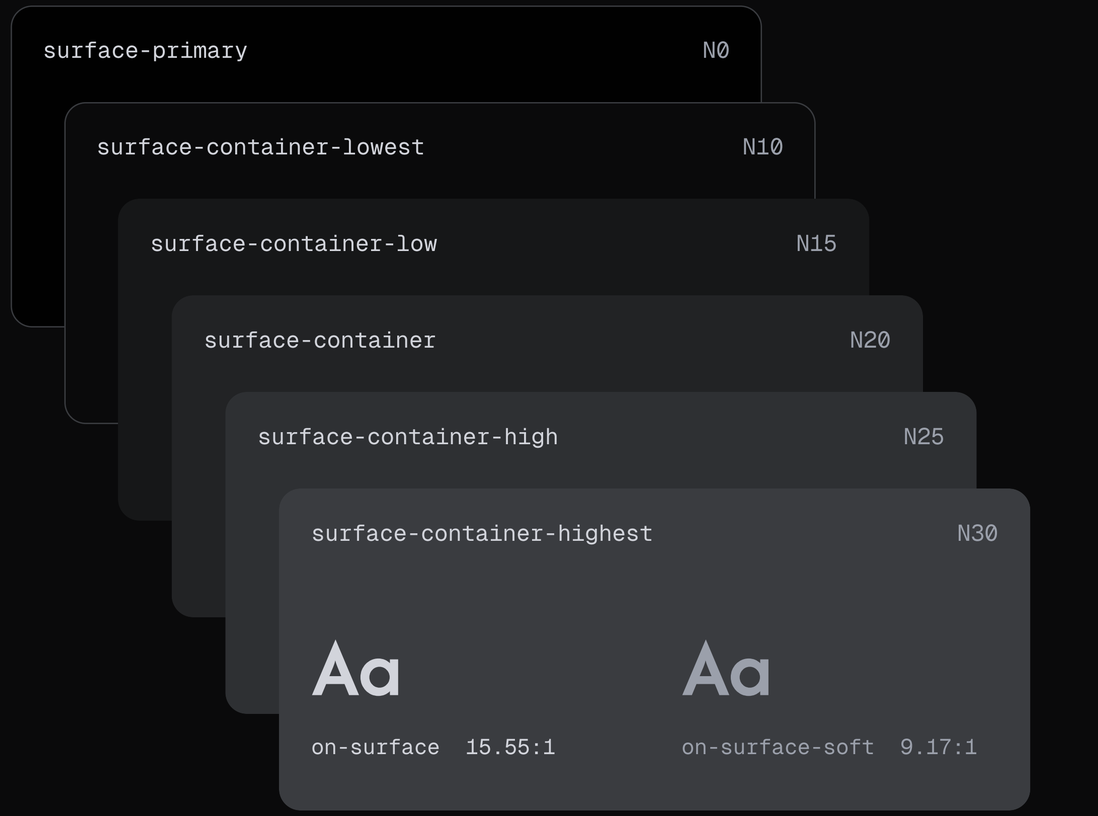


A neutral built for a dark-native product.

Neutral runs on its own formula: a fixed cool hue of 223, saturation that rises gently toward white, and lightness tied directly to the stop number. Half steps at 15, 25, 35 and 45 add resolution where a dark UI spends most of its pixels.

NEUTRAL RAMP FORMULA

```
n = stop / 10
H = 223
S = 2 + ΔS * n
L = clamp(stop - 6, 0, 100)
A = 100
```





Values and intent live in separate layers.

01 REFERENCE

What the color is

`ref/emerald/emerald-50`

#07B045
H 142 S 92 L 36

02 SYSTEM

What the color is for

`button-primary`

“Primary button fill.”

`-hover` `-active` `on-button`

03 CODE

Where it ships

`--button-primary`

```
.btn-primary {
  background: var(--button-primary);
}
```

100 colors across 8 ramps. Components never reference this layer directly.

58 role-based tokens in 7 groups. The only layer components consume.

CSS custom properties that mirror the system token name.

Every name follows one grammar.

support -success -on -subtle

GROUP | INTENT | RELATIONSHIP | EMPHASIS

STATES / PREDICTABLE SUFFIXES

button-primary · button-primary-hover · button-primary-active

PAIRS / EVERY FILL HAS AN ON- TOKEN

surface-container + on-surface · button-primary + on-button · field-primary + on-field

EMPHASIS / ONE SCALE FOR EVERY STATUS

support-danger · -bold · -subtle · -on-subtle

PROMPT

“Show a success state on the shortlist badge.”

RESOLVES TO

fill support-success-subtle

label support-success-on-subtle

AAA 7.19:1, decided by the system

Descriptions written for people and models.

Every token documents its role, its value, and its contrast. Each role is one plain sentence, so a designer, an engineer, or an AI tool reading the token can choose correctly without seeing the color.

Token	Role	Value	Contrast
`on-surface` `--on-surface`	Text and icons on any surface.	**Neutral 100** #D2D4DB	AAA 17.37:1
`button-primary` `--button-primary`	Primary button fill.	**Emerald 50** #07B045	AA 7.50:1 on surface-primary
`border-bold` `--border-bold`	High-emphasis border.	**Neutral 70** #9BA0AB	AA 8.85:1 on surface-primary
`link-primary` `--link-default`	Hyperlink text.	**Sapphire 70** #4B77E7	AAA 7.50:1 on surface-primary
`support-success-on-subtle` `--support-success-on-subtle`	Text, icons and border on support-success-subtle.	**Emerald 20** #0A3319	AAA 7.19:1 on success-subtle

Contrast lives in the token definition, so accessibility is decided once, at the system layer, instead of in every review.

Components consume roles, never values.

Send
button-primary
on-button

Connect
button-secondary
on-button-secondary

Filter
button-outline
on-button-outline

Settings
button-ghost
on-button-inverse

Overview Manage Explore Activity
link-tab
link-tab-active

Discover Requests Sent
button-toggle-active
on-button-toggle-active
button-toggle

Director
scrim-primary
scrim-canary
on-chip-director

Producer
scrim-primary
scrim-citrine
on-chip-producer

Crew
scrim-primary
scrim-neutral
on-chip-crew

Writer
scrim-primary
scrim-sapphire
on-chip-writer

Search people, studios, or communities
field-primary
on-field-primary

Max Grasselino
field-primary
on-field-primary-active
border-field-focus



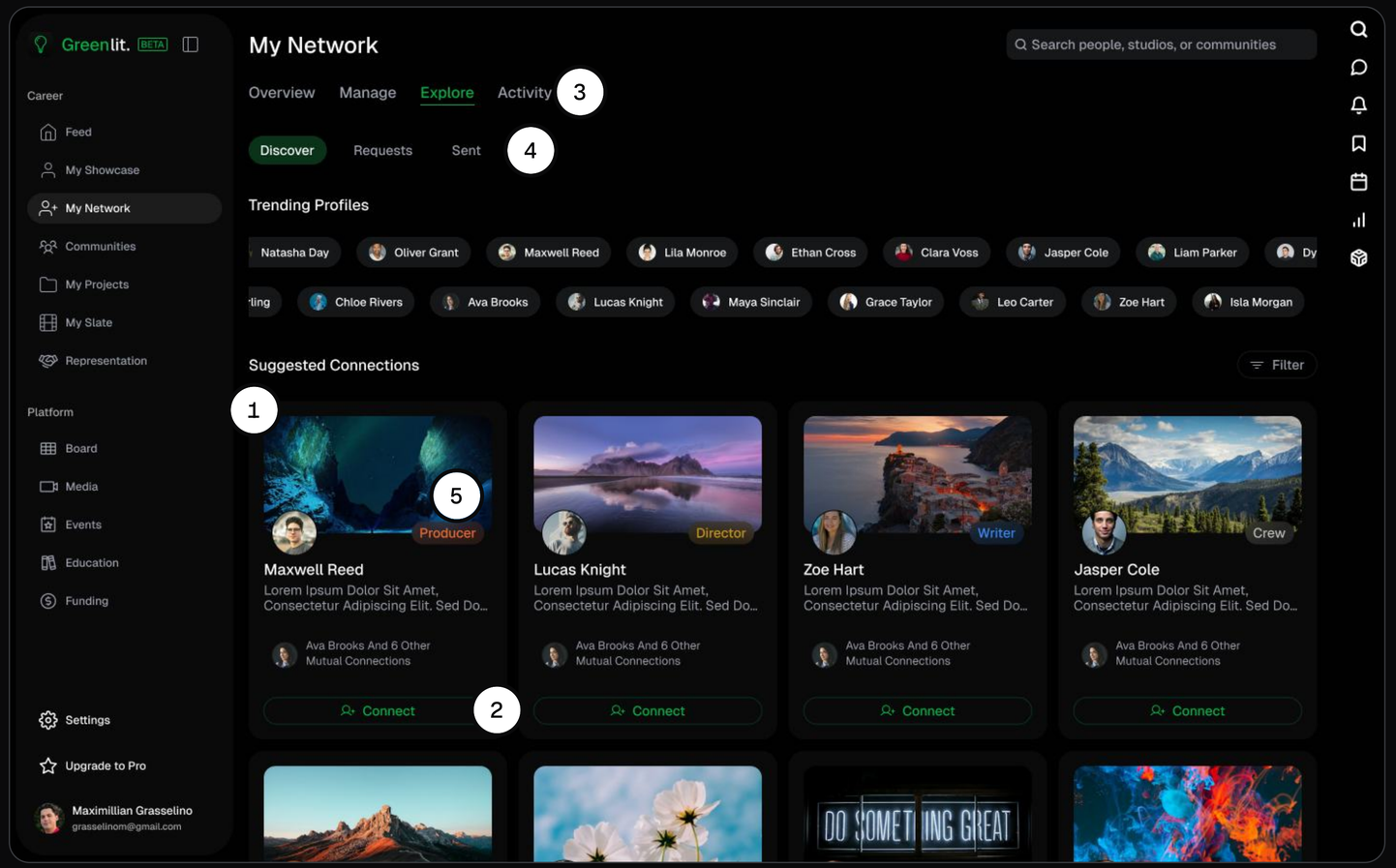
Jasper Cole
Lorem Ipsum Dolor Sit Amet,
Consectetur Adipiscing Elit. Sed Do...

Ava Brooks And 6 Other
Mutual Connections

Connect

surface-container-lowest
on-surface
on-surface-subtle

The system in production.



1 **Surfaces**
Cards sit on surface-container-lowest, with on-surface for names and on-surface-subtle for supporting copy. Search uses field-search with on-field-search.

2 **Connect**
The card action uses button-secondary with on-button-secondary.

3 **Selection**
The active tab uses link-tab--active. The Discover pill uses button-toggle-active with on-button-toggle-active.

4 **Rest states**
Inactive tabs use link-tab-primary, and unselected toggles use button-toggle.

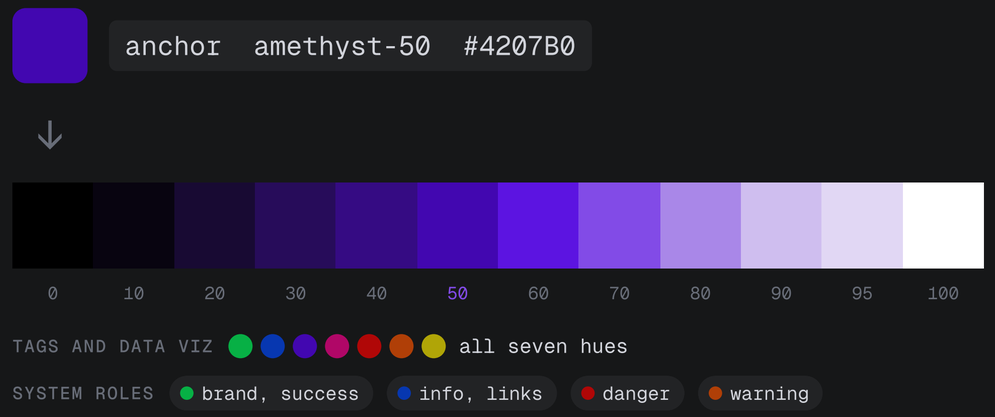
5 **Role tags**
Each tag layers scrim-primary under a hue scrim, such as scrim-citrine for Producer or scrim-sapphire for Writer, with a matching label token like on-chip-producer.

Extending the system is a parameter change.

NEW HUE

One value in, twelve stops out.

Choose an anchor at stop 50 and the formula returns all 12 stops at the same tonal positions as every existing hue. All seven hues color-code role tags, such as Producer, Director and Writer, and data visualization. Four of them also carry system roles.



NEW THEME

Light mode is a remap, not a rebuild.

The system ships dark-native. A light theme becomes a second mode on the system layer: names and component code stay put, and only the stop each token points to changes. The inverse tokens already define most of those values.

TOKEN	DARK / SHIPPED	LIGHT / EXAMPLE
surface-primary	N0	N100
on-surface	N100	N0
button-primary	E50	E40
link-primary	S70	S40
focus-default	E50	E40

What the system gives the team.

8
TONAL RAMPS

100
REFERENCE COLORS

58
SEMANTIC TOKENS

7
TOKEN GROUPS

Consistency

One formula and one tonal structure. Hierarchy holds no matter which hue is in play.

AI legibility

Intent lives in names and role sentences, so AI tools choose tokens instead of hex values.

Scalability

New hues and themes extend the token map without touching component code.

Color that designers, engineers, and agents read the same way.

WHAT COMES NEXT

01 Transition the system to a custom Storybook.

02 Ship light mode as a second mode on the system layer.

03 Create additional accessible color themes for users that are color blind.

Maximillian Grasselino
https://www.mvximillivn.com/

