

Midpoint Design Discussion

COLOuR PALETTE GENERATION FOR NOMINAL ENCODINGS

Ketrina Yim, Simon Tan, Calvin Ardi
CS 294-10 | Fall 2008

Project

- **Q:** How can we produce unique, quality palettes with minimal human intervention?
- **A:** Algorithmically (simulated annealing) using Brewer's [Color Use Guidelines](#) as heuristics for “good palettes”
- Added contributions:
 - Colorblind-friendly palettes
 - Color-harmonious (“themed”) palettes
 - Extension of prefuse / flare

Status

Task

- Heuristic/evaluation function
- Simulated Annealing
- User Interface

Progress

Currently involves contrast, closeness of neighboring values

Implemented and functional with placeholder functions

Overall look mostly implemented

Prototype

- DEMO!

Palette Assistant

Enter Palette Size (coming soon):

Choose Palette Preference:

- None
- Warm Colors
- Cool Colors

Choose Colorblindness Compatibility:

- None
- Red-Green Colorblindness
- Blue-Yellow Colorblindness

Generate

- 0xff576f4b
- 0xff485542
- 0xffc04f5c
- 0xff474834
- 0xff1e799c
- 0xffb34170
- 0xff2eb4fd
- 0xff404830
- 0xff89a39c
- 0xff1025b6

Evolution

- Definition of ‘colorblind’ discovered to have two dimensions
 - Red-green
 - Blue-yellow
- HSV believed to be best way to iterate through color space
- Put off to future work: perhaps use machine/supervised learning for computing weights in our heuristic

Questions

- Should we put work in the starting palette or let the heuristic function do all the work?
- How to display the list of generated colors when it grows too large?

Thank You



Questions?