Automating Color Ramp
Detection and
Modification
to Enhance
Pixel Art

This image "burns" eyes ❌

Only 1 edit with our tool ! ✔️

Author: Lenka Hake (l.hake@student.tudelft.nl)
Responsible professors: Elmar Eisemann, Petr Kellnhofer
Supervisor: Mathijs Molenaar

TUDelft

Color Ramps

What did we do?
Make a tool that can identify color ramps in pixel art images and assist users in editing them while preserving the relative color relationships.

Background: Pixel artists...
- ...use very little colors
- ...organize colors into ramps (meaningful sequences, e.g., from dark to light)
- ...frequently change and adjust their palettes
- ...have to manage ramps manually since existing tools don't support ramp editing

☐1 Build a Graph
- Take all unique colors in image:
  - Which colors are similar?
  - Which colors appear next to each other?
  - ⇨ Connect colors based on **Similarity** and **Adjacency**

- Two color similarity metrics:
  - **Hue, Saturation, Value (HSV)**
  - **Perceptual (CIEDE2000)**

- **Relative Adjacency:**
  - $RA(u, v) = A(u, v) / T(u)$
  - $A(u, v)$ = number of adjacent pixel pairs of colors u and v
  - $T(u)$ = total number of pixels of color u in the image

How can we Automate the detection and modification of Color Ramps in Pixel Art to streamline Palette Editing?

☐2 Identify Color Ramps
- Walk through the graph (depth-first search)
- At each step, validate if the current path meets the **3 Ramp Criteria**:
  - **Smoothness** = steps between colors are consistent (same size)
  - **Meaningfulness** = steps are not too small or too large
  - **Monotonous direction** = the direction of change is clear
- If validation succeeds, continue to the next node
- If validation fails, add the last valid path to the results and backtrack

☐4 Modify Colors
- When user modifies a color the tool automatically adjusts all colors in all related ramps
- Relative relationships between colors are preserved:
  - **Hue Offsets**
  - **Saturation** and **Value Ratios**

☐3 Filter Results
- Filter out redundant ramps:
  - **Permutations**
  - **Subsequences** that are fully contained in longer ramps

- Apply **Hierarchical Clustering** to group similar ramps and select representatives
  - Based on **Edit Distance** (How many insertions, deletions, and swaps are needed to change one ramp into another?)
  - Uses average linkage