Arduino Color Vision Deficiency (CVD) Simulator

❑ Context & Scientific Background:

I. Human Color Vision:

- Occurs through retinal photoreceptor cells called “cones” distinct from rods used for night vision.

- We rely on 3 cone types, with each type being primarily sensitive to a specific light wavelength:

    - L-cone - long wavelength (**red**).
    - M-cone - medium wavelength (**green**).
    - S-cone - short wavelength (**blue**).

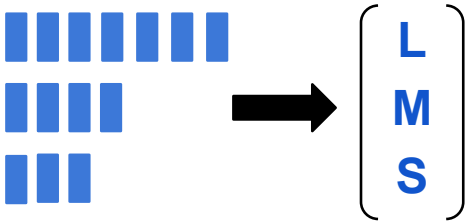
- CVD occurs when one cone type is missing or altered. For example: Protanopia (Red-Green CVD describes L-cone absence).

Arduino Color Vision Deficiency (CVD) Simulator

II. LMS Color Space:

- Colors can be represented as cone cell responses (L, M, S).
- Mathematically, we can model such responses as 3D vector spaces describing how each cone type responds to different light wavelengths.
- Let's pick a Reddish object. Here is a crude illustration to explain what the modeling may look like under normal color vision conditions:

L-cone response (red)
M-cone response (green)
S-cone response (blue)



Arduino Color Vision Deficiency (CVD) Simulator

❑ Research Basis:

I. How to arrive to the LMS space?

- Similarly to LMS, we can model the RGB color intensities as another 3D vector space called the RGB space.

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

- We can convert color information back and forth between the RGB and the LMS vectors using linear matrix transformations.

- Through experiments such as those by *Smith and Pokorny* with retinal cone sensitivities, scientists were able to produce the following 3x3 matrix which converts RGB vectors to LMS ones:

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 17.8824 & 43.5161 & 4.1194 \\ 3.4557 & 27.1554 & 3.8671 \\ 0.0300 & 0.1843 & 1.4671 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

L = 17.8824 R + 43.4161 G + 4.1194 B

https://vision.psychol.cam.ac.uk/jdmollon/papers/colourmaps.pdf

Arduino Color Vision Deficiency (CVD) Simulator

II. How to simulate CVD?

- Once we convert a color from RGB to LMS, we can modify the LMS vector's coefficients to reflect the corresponding CVD. **For instance: "*for protanopia, the L value is reset; and for tritanopia, the S value is reset.*"** https://pmc.ncbi.nlm.nih.gov/articles/PMC12387176/#sec2-jimaging-11-00268

- Then we just need to convert back to RGB using a reverse matrix transformation.

- Research done by *Machado et al.* provides precomputed matrices for different CVDs that allow us to convert from RGB and back to RGB skipping intermediate steps. The following is the protanopia matrix:

$$\begin{bmatrix} 0.152286 & 1.052583 & -0.204868 \\ 0.114503 & 0.786281 & 0.099216 \\ -0.003882 & -0.048116 & 1.051998 \end{bmatrix}$$

https://www.inf.ufrgs.br/%7Eoliveira/pubs_files/CVD_Simulation/CVD_Simulation.html#Abstract

Arduino Color Vision Deficiency (CVD) Simulator

❑ Arduino Implementation:

I. Circuit Main Components:

- Microcontroller:
  - Arduino Uno RV3

- Input:
  - TCS34725 RGB Color Sensor **(acquires raw rgb intensities - later normalized to 0-255 range)**
  - Mini push button **(toggles vision mode and uses built in pull-up resistor)**

- Output:
  - 2-line alphanumeric dot matrix 16-pin LCD screen (LCD)1602 **(displays vision mode and transformed RGB values)**
  - RGB LED (Common Cathode) **(lights up in transformed RGB values)**

Arduino Color Vision Deficiency (CVD) Simulator

II. Snapshots: