Color Theory "Game"

⊖ not started
⊜ in progress
⊕ Complete
? Stretch goals / unsure of implementation

ArrayList ActiveColors<Color>
forEach ... colorAction()
repaint visuals() Buttons/visuals

Base Functions

⊕ Color Storage (singular)
  ⊕ Attributes RGB
⊕ Color Palette storage

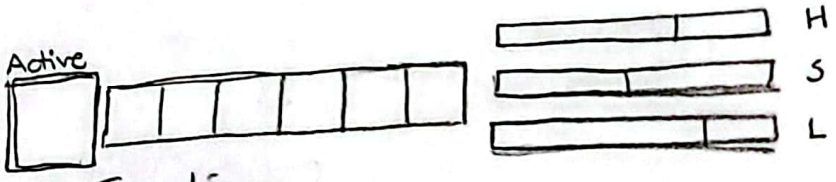
⊕ Adjusting color values
  ⊖ Adjusting palette

⊕ Draw simple objects of given color

⊕ Select single color ⊕
⊖ Select color palette ⊖
⊖ Select multiple colors ⊖
?⊖ Combine colors ($\frac{r1+r2}{2}$, $\frac{g1+g2}{2}$, $\frac{b1+b2}{2}$) ⊖
  ⊕ add lightness / darkness (all RGB) ⊖
  ⊕ increase / dec saturation difference between low + high R, G, and B values ⊖
  ⊕ shift hue convert HSV? ⊖
  ⊕ Warmth Red/Blue relationship ⊖

⊕ Visuals for palette
  ⊕ update visuals @ user input (repaint)
⊖ Export palette ⊖

⊖ Color Analysis → ⊖ Comparison of two colors
  ⊕ Saturation difference between lowest + highest RGB values
  ⊕ Value adjust all RGB values at once
  ⊕ Hue easiest in HSV space
  ⊖ Warmth balance between Red and Blue



Stretch Goals / Extended Functions

⊖ Color in context, ask user to
  ⊖ choose 'most' or 'least' ____ object
  ⊖ choose color that best fits in context
  ⊖ adjust color to fit in context
  ⊖ choose complementary color / color palette questions
⊖ Color palette prompts, user can shift individual colors or entire palette until satisfied
⊖ Simple color wheel

CamScanner

Get values

- ⊕ Color
- ⊕ R
- ⊕ G
- ⊕ B
- ⊕ Hue
- ⊕ Saturation
- ⊕ Brightness

Set Values w/ restriction of range

- ⊕ Color (0-255)
- ⊕ R (0.0-1.0)
- ⊕ G
- ⊕ B
- ⊕ Hue
- ⊕ Saturation
- ⊕ Brightness

Shift Values

- ⊕ R
- ⊕ G
- ⊕ B
- test ⊕ Hue
- ⊕ Saturation
- ⊕ Brightness
- ⊕ Warmth
- ⊕ Value (RGB)

Select colors

- *⊕ single active color
- ○ entire palette
- ○ active colors

int[] ArrayList (indexes)

Compare values (less priority)

- return Color Object of greater value / return index?

input objects or index?

- ○ Hue
- ○ Saturation
- ○ Brightness
- ○ Warmth - closeness to arbitrary warm hue value (abs value diff) (255, 200, 0)
- ⊖ R
- ○ G
- ○ B

Wait to see needed function

- ○ Color (equals()) ? $\left(\frac{r_1+r_2+\dots+r_n}{n}, \frac{g_1+g_2+\dots+g_n}{n}, \frac{b_1+b_2+\dots}{n}\right)$

Scale values

- ⊕ R
- ⊕ G
- ⊕ B
- ○ ~~Hue~~
- ⊕ Saturation
- ⊕ Brightness
- ○

Visuals / Quality

- ⊖ appealing layout
- *○ javadoc comments
- ○ Combine colors

Alchemy

- ○ Select colors to combine, store ~~to assign new color~~
- ○ Select active color, then other colors to add (can be weighted or restricted to certain attributes)

Strength of other colors to add: (out of 100)

Attributes to modify:

- ☐ Hue
- ☐ Saturation
- ☐ Brightness

Future: Which color added to A most closely matches B?

Buttons / Controls

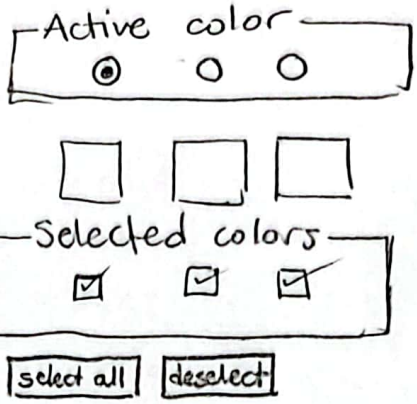
- ⊖ select Color (s)
- ⊖ decide which modifiers to show
- ⊖ decide visual layout
- ⊕ ~~Buttons/incremental change~~ or * text boxes for num values (easier start)

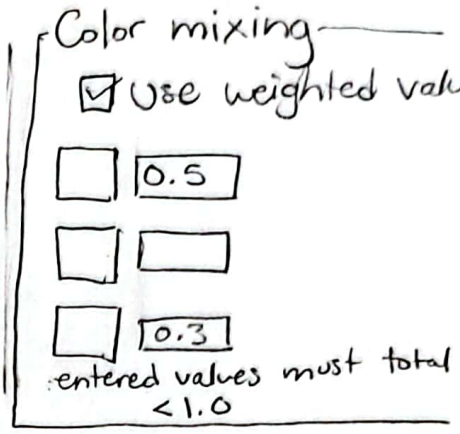
When is it significant to use this.method() as opposed to method() inside of a class? is it only needed when accessing attributes?

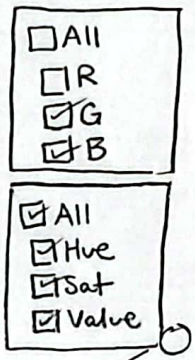
12/10/24

⊕ int activeColor

○ int[] selectedColors
  ○ select all ○ deselect buttons
⊖ value modification (direct)
  ⊕ RGB — ⊕ set, ⊕ shift, ⊕ scale
  ○ HSB — ○ set, ○ shift, ○ scale (SB only)
  ⊖ Calculate changes to update
○ Color mixing
  ○ RGB
    ○ average — mathematical average for R,G,B values $\left(\frac{a+x+y+z}{n}\right)$
    ○ weighted combinations
    ○ toggle R,G,B fields
  ○ HSB
    ○ average
    ○ weighted combinations
    ○ toggle H,S,B fields
○ Center palette on page
  ↳ ○ Allow multiple palette sizes







Direct modification
- ◉ Only active
- ○ Active & Selected

○ add JRadioButtons to HashMap based on palette size
  — should be able to access based on index

○ Create separate listeners for buttons (probably good idea to create new classes)
  ○ Access activeColor using getter (non-static)

CamScanner