

Київський національний університет імені Тараса Шевченка
Механіко-математичний факультет

Звіт на тему:
«COLOR»

Виконав:
студент 3 курсу, 1-ша групи комп'ютерної математики
Решетник Віталій Владиславович
Київ 2023

Вступ

Метою цього звіту є детальний розбір роботи класу Color та його нащадків/допоміжних класів. Ця реалізація полегшує роботу з кольорами, особливо з переведенням кольорів з однієї системи в іншу і одразу візуалізація. А також можливість робити це зі списками кольорів.

Кольорові системи

RGBA – Red Green Blue Alpha, змінні набувають значень від 0 до 255, три кольори а також прозорість 0 – прозорий, 255 – максимально насичений.

CMYK - Cyan Magenta Yellow Key Black, змінні набувають раціональних значень від 0 до 100 інтерпретують відсоток кольору в кольорі.

XYZ – математична модель кольору, кольорові координати, змінні набувають раціональних значень.

HSB – Hue Saturation Brightness, змінні набувають раціональних значень, інтерпретація кольору через циклічний градієнт і відповідний кут в ньому.

Усі 4 системи кольорів є нащадками Color, в якому визначені методи.

Побітові операції **&(bitwiseIntersection) ^ (bitwiseXOR) | (bitwiseUnion)**, але так як це бітові операції вони можуть бути виконані тільки для цілих чисел, тобто для RGBA.

Для бітових операцій, а також для операції змішування передається і ще один колір такої ж системи, і виконуються відповідні дії: бітові операції зі змінними або ж береться середнє(**mixColors**).

Для методів отримання цілого та раціонального значення довелося придумувати власні підходи так як не зустрічав відповідних функцій в інтернет мережі.

Також, для кожного класу є аналогічні функції введення виведення. Введення з консолі **inputColorHSB**, зі стрічки **setFromFormattedString** для зчитування по рядку файлу та передача стрічок цьому методу. Виведення в консоль **toString** та в стрічку **toStringFormatted**, знову ж для запису в файл.

Робота з файлами передбачає порядкове зчитування та запис в список **List<ColorRGBA> colorList**, де потім за допомогою циклу можна окремо опрацьовувати кожен елемент.

За для спрощення коду і його не нагромадження усі перетворення між кольорами відбуваються через RGBA так як він є найбільш вживаним та простим в розумінні. Тому наприклад для перетворення з CMYK в XYZ треба спочатку CMYK перетворити в RGBA, а потім RGBA в XYZ.

Перетворення CMYK в RGBA і навпаки не є доволі складним так як системи майже ідентичні в механізмі, CMYK походить від RGBA. Перетворення HSB в RGBA і навпаки доволі ускладнена задача так як представлення HSB є конусом де рух по висоті змінює яскравість, від центру до краю – насиченість, а по колу – колір. Не прості перетворення й в XYZ так як це математична модель людського сприйняття кольору і треба рухатись по трьох-вимірному графіку, заданим суто імперичними значеннями, тому при перетворенні є багато коефіцієнтів.

Image/Mask

Image – матриця, елементами якої є об’єкти саме **ColorRGBA**, так як саме ця система використовується для графіки в Java. Коли для Mask – це не кольори а цілі значення де 0-чорний, 255-білий. В обох випадках створюється клас нащадок від **JFrame** для побудови відповідних картинок. Так як показ матриць малого розміру, не був би достатньо наглядним, додав параметр **cellSize** для збільшення розміру пікселя. Присутнє введення з консолі **inputImage**, а також виведення в неї **outputImage**. І введення по-елементно через параметри методу за для записування кольорів в матрицю зі списку, що був раніше прочитаний з файлу. Для маски, щоб отримати чорно-біле представлення кольору використовував вираз

(int) Math.floor(colorList.get(i).getValueFloat()*255)

getValueInt() в даному випадку не підходить так як виходить за межі допустимих значень.

Run files

В роботі два файли з main, run_test_file.java повністю автоматичний і проходиться по всім функціям, коли run_test_console.java вимагає введення з консолі, але тут теж можна використати значення з файлу.

Примітка: в run_test_file.java викликається візуалізація і Mask і Image, тому вікна закривають одне-одне, тому щоб побачити варто відсунути в сторону.

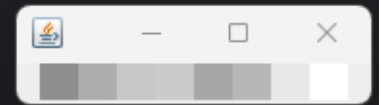
Висновок

Ця реалізація пропонує доволі широкий спектр дій з кольоровими системами та зображеннями так як є робота з файлами, але використання вже існуючих бібліотек та програм буде набагато доцільнішим при щоденному використанні.

Але за одиничною потребою це є прекрасним варіантом.

Скріншоти роботи run_test_file.java

```
0 ColorRGBA{red=255, green=0, blue=255, alpha=128}
1 ColorRGBA{red=35, green=123, blue=25, alpha=216}
2 ColorRGBA{red=5, green=0, blue=255, alpha=78}
3 ColorRGBA{red=15, green=57, blue=135, alpha=23}
4 ColorRGBA{red=25, green=142, blue=52, alpha=178}
5 ColorRGBA{red=35, green=123, blue=25, alpha=126}
6 ColorRGBA{red=5, green=0, blue=20, alpha=72}
7 ColorRGBA{red=0, green=0, blue=0, alpha=0}
```



Perform bitwise operations for every element of list with first element
XOR

```
ColorRGBA{red=0, green=0, blue=0, alpha=0}
ColorRGBA{red=220, green=123, blue=230, alpha=88}
ColorRGBA{red=250, green=0, blue=0, alpha=206}
ColorRGBA{red=240, green=57, blue=120, alpha=151}
ColorRGBA{red=230, green=142, blue=203, alpha=50}
ColorRGBA{red=220, green=123, blue=230, alpha=254}
ColorRGBA{red=250, green=0, blue=235, alpha=200}
ColorRGBA{red=255, green=0, blue=255, alpha=128}
```



XOR list was written to fileoutputs/output1.txt

Intersection

```
ColorRGBA{red=255, green=0, blue=255, alpha=128}
ColorRGBA{red=35, green=0, blue=25, alpha=128}
ColorRGBA{red=5, green=0, blue=255, alpha=0}
ColorRGBA{red=15, green=0, blue=135, alpha=0}
ColorRGBA{red=25, green=0, blue=52, alpha=128}
ColorRGBA{red=35, green=0, blue=25, alpha=0}
ColorRGBA{red=5, green=0, blue=20, alpha=0}
ColorRGBA{red=0, green=0, blue=0, alpha=0}
```

Union

```
ColorRGBA{red=255, green=0, blue=255, alpha=128}  
ColorRGBA{red=255, green=123, blue=255, alpha=216}  
ColorRGBA{red=255, green=0, blue=255, alpha=206}  
ColorRGBA{red=255, green=57, blue=255, alpha=151}  
ColorRGBA{red=255, green=142, blue=255, alpha=178}  
ColorRGBA{red=255, green=123, blue=255, alpha=254}  
ColorRGBA{red=255, green=0, blue=255, alpha=200}  
ColorRGBA{red=255, green=0, blue=255, alpha=128}
```

Perform mix/mean operations for every element of list with first element

```
ColorRGBA{red=255, green=0, blue=255, alpha=128}  
ColorRGBA{red=145, green=61, blue=140, alpha=172}  
ColorRGBA{red=130, green=0, blue=255, alpha=103}  
ColorRGBA{red=135, green=28, blue=195, alpha=75}  
ColorRGBA{red=140, green=71, blue=153, alpha=153}  
ColorRGBA{red=145, green=61, blue=140, alpha=127}  
ColorRGBA{red=130, green=0, blue=137, alpha=100}  
ColorRGBA{red=127, green=0, blue=127, alpha=64}
```

Transform to XYZ

```
ColorXYZ{x=0.5928939, y=0.28484792, z=0.969638}  
ColorXYZ{x=0.07951123, y=0.1459258, z=0.03317134}  
ColorXYZ{x=0.18106346, y=0.07249776, z=0.9503334}  
ColorXYZ{x=0.060317196, y=0.04776317, z=0.23520993}  
ColorXYZ{x=0.10692933, y=0.19799307, z=0.06506239}  
ColorXYZ{x=0.07951123, y=0.1459258, z=0.03317134}  
ColorXYZ{x=0.0018881927, y=8.2765357E-4, z=0.006677109}  
ColorXYZ{x=0.0, y=0.0, z=0.0}
```

Transform to CMYK

```
ColorCMYK{cyan=0.0, magenta=1.0, yellow=0.0, black=0.4980392}  
ColorCMYK{cyan=0.8627451, magenta=0.517647, yellow=0.9019608, black=0.15294117}  
ColorCMYK{cyan=0.98039216, magenta=1.0, yellow=0.0, black=0.69411767}  
ColorCMYK{cyan=0.9411765, magenta=0.7764706, yellow=0.4705882, black=0.9098039}  
ColorCMYK{cyan=0.9019608, magenta=0.44313723, yellow=0.79607844, black=0.30196077}  
ColorCMYK{cyan=0.8627451, magenta=0.517647, yellow=0.9019608, black=0.5058824}  
ColorCMYK{cyan=0.98039216, magenta=1.0, yellow=0.92156863, black=0.7176471}  
ColorCMYK{cyan=1.0, magenta=1.0, yellow=1.0, black=1.0}
```

```
Transform to HSB
ColorHSB{hue=300.0, saturation=100.0, brightness=100.0}
ColorHSB{hue=113.87755, saturation=79.6748, brightness=48.235294}
ColorHSB{hue=241.17648, saturation=100.0, brightness=100.0}
ColorHSB{hue=219.0, saturation=88.8889, brightness=52.941177}
ColorHSB{hue=133.84615, saturation=82.39437, brightness=55.686275}
ColorHSB{hue=113.87755, saturation=79.6748, brightness=48.235294}
ColorHSB{hue=255.0, saturation=100.0, brightness=7.8431377}
ColorHSB{hue=0.0, saturation=0.0, brightness=0.0}

Visualise in colors
Image dimensions: 1 x 8
Pixel values (RGBA format):
RGBA 255 0 255 128 RGBA 35 123 25 216 RGBA 5 0 255 78 RGBA 15 57 135 23 RGBA 25 142 52 178 RGBA 35 123 25 126 RGBA 5 0 20 72 RGBA 0 0 0 0

Visualise in gray
Mask dimensions: 1 x 8
Pixel int values:
142 173 199 203 167 183 232 255
```