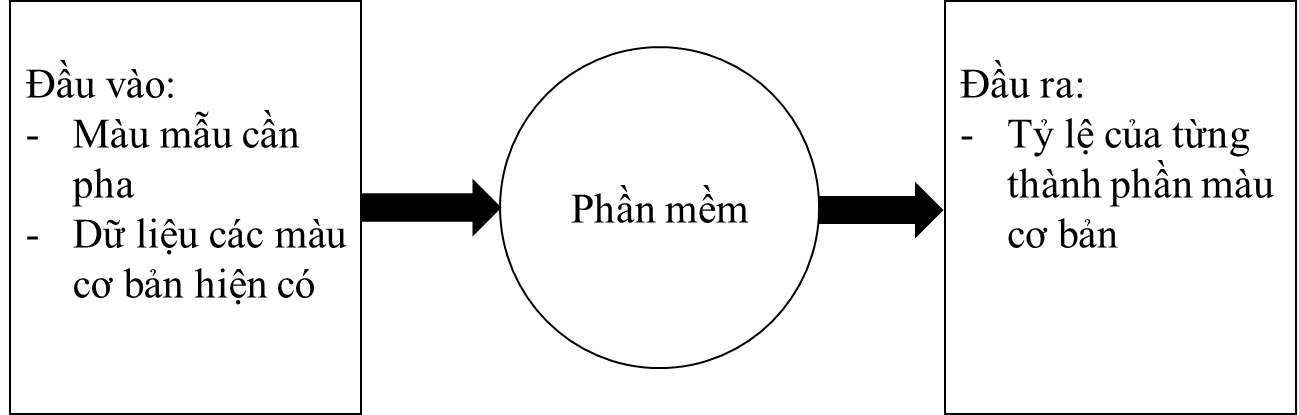
# NỘI DUNG

## PHÂN TÍCH, THIẾT KẾ VÀ TRIỂN KHAI

### Xác định yêu cầu

Thiết kế một phần mềm cho phép thực hiện tính toán tỷ lệ của các màu thành phần cơ bản sao cho khi pha theo đúng tỷ lệ đó thì sẽ thu được màu giống với màu mẫu cần pha.



Hình 1

#### Dữ liệu đầu vào

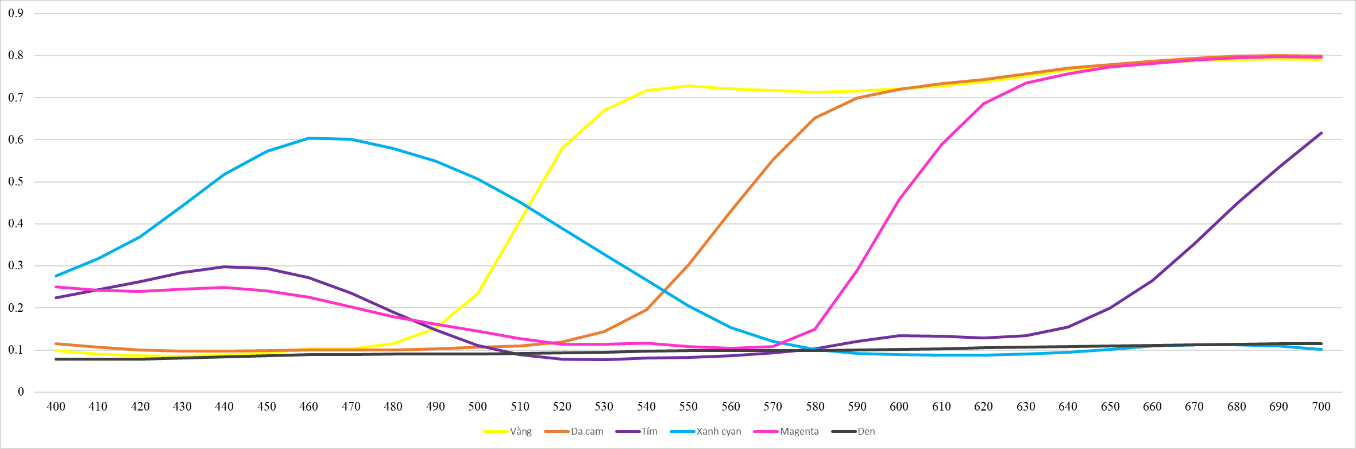
*\* 6 màu cơ bản:*

Đầu vào của phần mềm gồm dữ liệu của 6 màu cơ bản dùng để pha trộn tạo ra các màu mong muốn. Dữ liệu này là hệ số phản xạ tương ứng của màu trong dải bước sóng từ 400 *nm* đến 700 *nm*. 6 màu cơ bản gồm có: magenta, da cam, vàng, xanh cyan, tím và đen. Dữ liệu của 6 màu được cho trong Bảng 2 dưới đây.

Bảng 2 Hệ số phản xạ của 6 màu cơ bản

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Bước sóng (*nm*)** | **Hệ số phản xạ** | | | | | |
| Magenta | Da cam | Vàng | Xanh cyan | Tím | Đen |
| 400 | 0.2502 | 0.1154 | 0.0989 | 0.2763 | 0.2238 | 0.0782 |
| 410 | 0.2427 | 0.1066 | 0.0911 | 0.3170 | 0.2439 | 0.0777 |
| 420 | 0.2393 | 0.1004 | 0.0863 | 0.3692 | 0.2628 | 0.0784 |
| 430 | 0.2443 | 0.0978 | 0.0850 | 0.4431 | 0.2845 | 0.0809 |
| 440 | 0.2484 | 0.0974 | 0.0872 | 0.5174 | 0.2985 | 0.0840 |
| 450 | 0.2410 | 0.0984 | 0.0939 | 0.5722 | 0.2936 | 0.0866 |
| 460 | 0.2254 | 0.0998 | 0.1022 | 0.6036 | 0.2724 | 0.0886 |
| 470 | 0.2031 | 0.1000 | 0.1022 | 0.6016 | 0.2349 | 0.0897 |
| 480 | 0.1794 | 0.1005 | 0.1146 | 0.5794 | 0.1906 | 0.0903 |
| 490 | 0.1613 | 0.1030 | 0.1526 | 0.5491 | 0.1477 | 0.0905 |
| 500 | 0.1450 | 0.1069 | 0.2335 | 0.5072 | 0.1109 | 0.0908 |
| 510 | 0.1271 | 0.1091 | 0.4041 | 0.4520 | 0.0891 | 0.0915 |
| 520 | 0.1139 | 0.1193 | 0.5792 | 0.3901 | 0.0780 | 0.0928 |
| 530 | 0.1138 | 0.1438 | 0.6701 | 0.3278 | 0.0770 | 0.0950 |
| 540 | 0.1166 | 0.1954 | 0.7171 | 0.2662 | 0.0808 | 0.0972 |
| 550 | 0.1084 | 0.3022 | 0.7286 | 0.2055 | 0.0827 | 0.0982 |
| 560 | 0.1043 | 0.4305 | 0.7214 | 0.1534 | 0.0859 | 0.0986 |
| 570 | 0.1081 | 0.5523 | 0.7168 | 0.1206 | 0.0929 | 0.0988 |
| 580 | 0.1493 | 0.6524 | 0.7138 | 0.1015 | 0.1033 | 0.0991 |
| 590 | 0.2896 | 0.6992 | 0.7157 | 0.0920 | 0.1200 | 0.1001 |
| 600 | 0.4572 | 0.7195 | 0.7208 | 0.0888 | 0.1342 | 0.1015 |
| 610 | 0.5883 | 0.7337 | 0.7277 | 0.0875 | 0.1325 | 0.1032 |
| 620 | 0.6864 | 0.7438 | 0.7373 | 0.0883 | 0.1290 | 0.1050 |
| 630 | 0.7348 | 0.7574 | 0.7516 | 0.0902 | 0.1349 | 0.1066 |
| 640 | 0.7570 | 0.7702 | 0.7662 | 0.0941 | 0.1553 | 0.1080 |
| 650 | 0.7727 | 0.7794 | 0.7758 | 0.1018 | 0.2001 | 0.1094 |
| 660 | 0.7816 | 0.7868 | 0.7829 | 0.1094 | 0.2647 | 0.1109 |
| 670 | 0.7900 | 0.7936 | 0.7882 | 0.1129 | 0.3528 | 0.1124 |
| 680 | 0.7955 | 0.7986 | 0.7913 | 0.1130 | 0.4487 | 0.1139 |
| 690 | 0.7975 | 0.8004 | 0.7921 | 0.1093 | 0.5347 | 0.1148 |
| 700 | 0.7960 | 0.7994 | 0.7906 | 0.1019 | 0.6161 | 0.1154 |

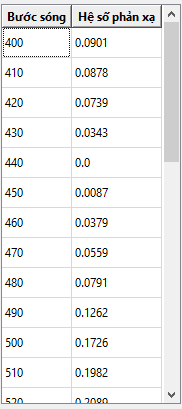
Hình 2 biểu diễn trực quan đường phổ của 6 màu cơ bản.



Hình 2 Đường phổ của 6 màu cơ bản

*\* Màu cần pha:*

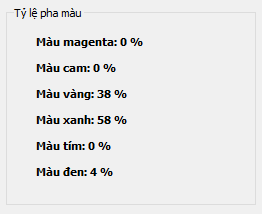
Dữ liệu về màu cần pha cũng được cho ở dạng hệ số phản xạ của màu đó tương ứng với các bước sóng trong dải 400 *nm* đến 700 *nm*.



#### Dữ liệu đầu ra

Dữ liệu đầu ra của phần mềm là một bộ số *a1, a­2, a3, a4, a5, a6* thỏa mãn:

Và kết quả khi thực hiện các phép tính toán giữa các hệ số này và phổ của 6 màu cơ bản đầu vào thu được là một phổ sao cho màu sắc thu được giống với màu mẫu cần pha ban đầu nhất.



Hình 3

### Triển khai chi tiết

Phần mềm trong đề tài này được thiết kế, lập trình bằng ngôn ngữ Java.

#### Ngôn ngữ lập trình Java

##### Giới thiệu

Java được tạo ra vào năm 1991 do một số kỹ sư ở Sun, bao gồm ông James Gosling, một phần của dự án Green Project. Java được phát hành vào năm 1994 và thực sự trở nên nổi tiếng khi Netscape tuyên bố tại hội thảo SunWorld năm 1995 là trình duyệt của họ sẽ hỗ trợ Java.

Java là một trong những ngôn ngữ lập trình hướng đối tượng, được sử dụng trong hầu hết các lĩnh vực của phát triển phần mềm gồm app máy tính, app di động, website, game,…

##### Các đặc điểm của Java

* Hướng đối tượng hoàn toàn:

Nhằm mục đích tạo ra một ngôn ngữ mới dễ học và quen thuộc với đa số người lập trình, các kỹ sư của Sun MicroSystem đã sử dụng lại các cú pháp của C và C++. Tuy nhiên, khác với C và C++, các thao tác với con trỏ đã bị lược bỏ hoàn toàn để đảm bảo tính an toàn và dễ sử dụng.

* Độc lập phần cứng và hệ điều hành:

Một chương trình viết bằng Java không bị giới hạn bởi phần cứng và hệ điều hành, nó có thể chạy được ở nhiều môi trường khác nhau, được gọi là khả năng “cross-platform”. Khả năng độc lập được thể hiện ở 2 cấp độ là cấp độ mã nguồn và cấp độ nhị phân.

Ở cấp độ mã nguồn: các kiểu dữ liệu trong Java được thống nhất chung cho tất cả các hệ điều hành và phần cứng khác nhau. Java có riêng một bộ thư viện để hỗ trợ điều này, do đó chương trình viết bằng Java có thể biên dịch trên nhiều loại máy khác nhau mà không gặp lỗi.

Ở cấp độ nhị phân: một mã biên dịch của Java có thể chạy trên nhiều nền tảng khác nhau mà không cần biên dịch lại mã nguồn.

* Ngôn ngữ lập trình thông dịch:

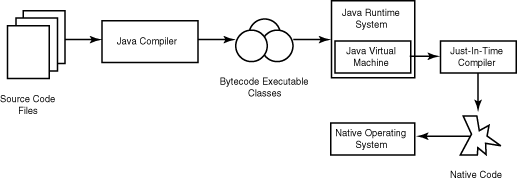
Ngôn ngữ lập trình thường được chia ra làm hai loại: thông dịch và biên dịch

Thông dịch (Interpreter): thực hiện dịch từng lệnh rồi chạy từng lệnh, mỗi lần chạy thì cần phải dịch lại

Biên dịch (compiler): mã nguồn sau khi được biên dịch sẽ tạo ra một file thực thi (ví dụ trên Windows là .exe), và file thực thi này có thể sử dụng lại mà không cần biên dịch nữa.

Java là ngôn ngữ vừa biên dịch, vừa thông dịch.

Khi viết mã nguồn, hệ thống tạo ra một file .java. Khi biên dịch mã nguồn của chương trình sẽ biên dịch ra mã byte code. Máy ảo Java sẽ thông dịch mã byte code này thành mã máy tương ứng với phần cứng và hệ điều hành khi nhận được yêu cầu chạy chương trình.



Hình 4 Quá trình thực thi chương trình Java

* Cơ chế thu gom rác tự động:

Khi tạo ra các đối tượng, JRE (Java Runtime Environment) tự động cấp phát không gian bộ nhớ cho các đối tượng trên heap.

Với ngôn ngữ C/C++, sau khi cấp phát bộ nhớ, ta cần phải thực hiện yêu cầu giải phóng vùng nhớ đã cấp phát để đảm bảo không lãng phí vùng nhớ và đảm bảm hiệu năng của chương trình.

Trong Java, việc giải phóng các vùng nhớ sau khi sử dụng xong được thực hiện tự động. Bộ dọn rác của Java sẽ theo vết các tài nguyên đã cấp phát, khi không có tham chiếu nào đến vùng nhớ, bộ thu dọn rác sẽ tự động tiến hành thu hồi các vùng nhớ đó.

* Đa luồng:

Java hỗ trợ lập trình đa tiến trình để thực thi nhiều công việc đồng thời. Bên cạnh đó, nó cũng cung cấp các giải pháp cho phép đồng bộ giữa các tiến trình.

* Tính an toàn và bảo mật:

Tính an toàn:

* + Ngôn ngữ Java yêu cầu chặt chẽ về các kiểu dữ liệu
  + Dữ liệu phải được khai báo tường minh
  + Không sử dụng con trỏ và các phép toán với con trỏ
  + Java kiểm soát chặt chẽ việc truy cập đến mảng, chuỗi
  + Tự động cấp phát và giải phóng bộ nhớ

Tính bảo mật: gồm 4 mức

* + Mức 1: chỉ có thể truy xuất dữ liệu cũng như phương thức thông qua giao diện mà lớp cung cấp
  + Mức 2: Trình biên dịch kiểm soát các đoạn mã sao cho tuân thủ các quy tắc của ngôn ngữ Java trước khi thông dịch
  + Mức 3: Trình thông dịch sẽ kiểm tra byte code đảm bảo các quy tắc trước khi thực thi
  + Mức 4: Java kiểm soát việc nạp các lớp vào bộ nhớ để giám sát việc vi phạm giới hạn truy xuất trước khi nạp vào hệ thống

##### Máy ảo Java (JVM)

Máy ảo Java (Java Virtual Machine) là một phần mềm được triển khai trên máy tính giúp thực hiện các chương trình giống như một máy thật. Máy ảo Java được viết cho các hệ điều hành khác nhau để đảm bảo khả năng đa nền tảng ngôn ngữ lập trình Java.



Hình 5 Kiến trúc của một chương trình Java

Hình 5 mô tả các thành phần chính của một hệ điều hành chạy được chương trình viết bằng Java. Các chương trình sau khi chuyển sang định dạng byte code sẽ được máy ảo Java dịch thành mã máy để hệ điều hành có thể xử lý, kết hợp cùng phần cứng để thực hiện chạy chương trình.

#### Thiết kế giao diện phần mềm

Ta sử dụng framework Java Swing của ngôn ngữ lập trình Java để lập trình giao diện cho phần mềm.

Java Swing là một phần của Java Foundation Classes (JFC) được sử dụng để tạo các ứng dụng window-based. Nó được xây dựng trên API AWT (Abstract Windowing Toolkit) và được viết hoàn toàn bằng Java.

Không giống như AWT, Java Swing cung cấp các thành phần không phụ thuộc vào nền tảng và nhẹ hơn.

##### Giao diện chọn màu cơ bản

Graphical user interface, application

Description automatically generated

Giao diện này gồm 4 checkbox tương ứng với 4 màu cơ bản. Có 4 màu cơ bản là: vàng, xanh cyan, hồng và đen. Người dùng tick vào các checkbox tương ứng để chọn/bỏ chọn màu sẽ sử dụng để pha.

##### Giao diện nhập dữ liệu màu cần pha

Table

Description automatically generated

Giao diện này cho phép người dùng nhập dữ liệu của màu cần pha từ một file data bên ngoài. Dữ liệu nhập vào là hệ số phản xạ của màu trên dải bước sóng 400 – 700 *nm*.

##### Giao diện hiển thị dữ liệu đầu ra

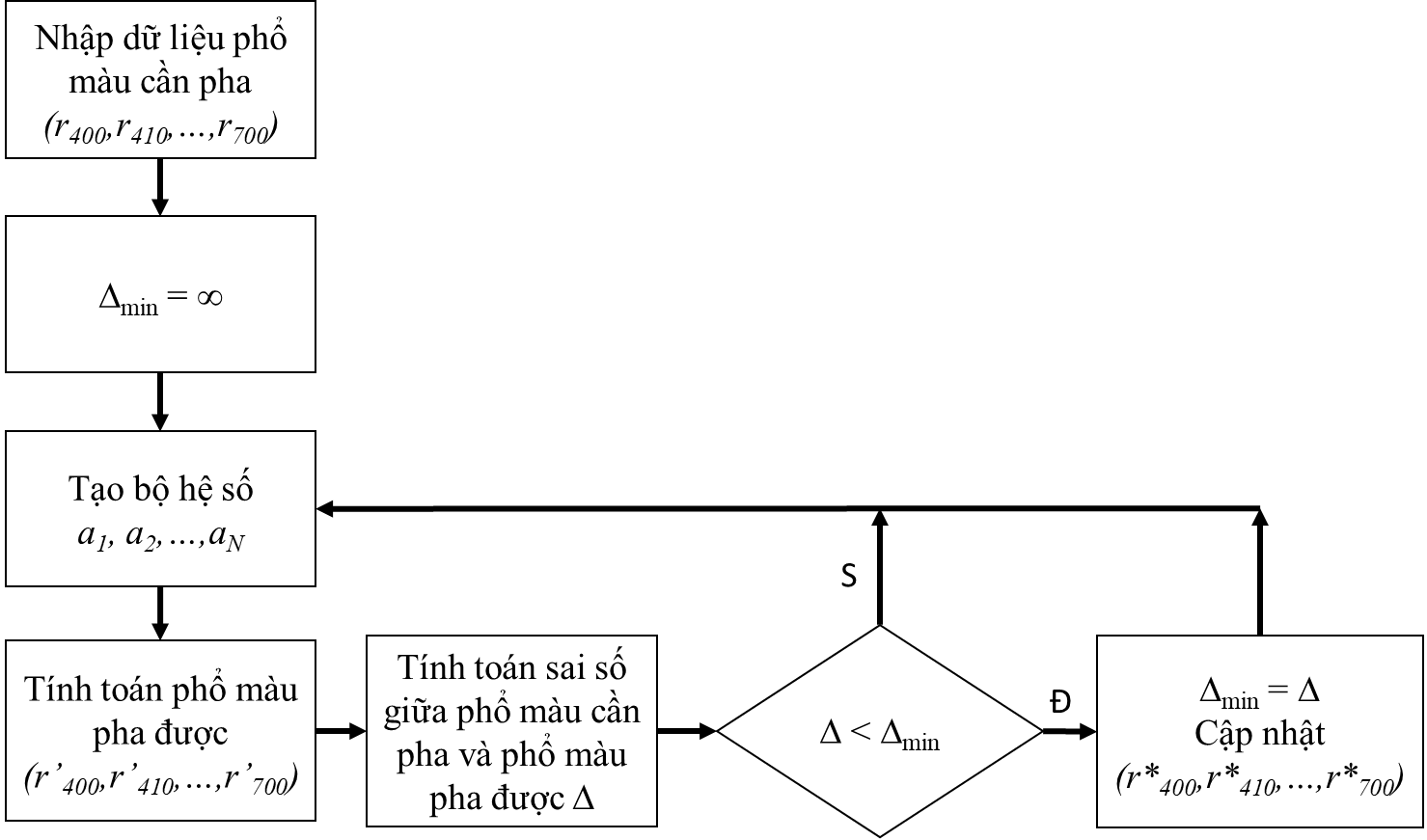
Chart

Description automatically generated

Giao diện này hiển thị kết quả đầu ra sau khi phần mềm thực hiện các phép xử lý, tính toán. Những nội dung được hiển thị gồm có tỷ lệ của các màu cơ bản thành phần, biểu đồ so sánh phổ của màu cần pha và màu pha được, màu sắc mà mắt ta sẽ nhìn thấy của 2 màu được tính toán từ phổ.

#### Xử lý, tính toán

##### Lưu đồ thuật toán



Hình 6

##### Tạo bộ hệ số tỷ lệ

Ở bước này, ta sử dụng thuật toán đệ quy quay lui (back-tracking) để tìm tất cả các bộ hệ số *a1, a2, a3, a4* sao cho thỏa mãn:

Với mỗi bộ số *a1, a2, a3, a4*  tìm được, kết hợp với phổ của 4 màu cơ bản, ta tính được phổ của màu pha được.

##### Tính toán phổ màu pha được

Ta biểu diễn phổ của 4 màu cơ bản dưới dạng ma trận có kích thước 31x4, trong đó 31 hàng tương ứng với các bước sóng 400, 410, 420, …, 700 *nm*. 4 cột tương ứng với 4 màu cơ bản.

Các hệ số *a1, a2, a3, a4* được biểu diễn dưới dạng ma trận kích thước 4x1.

Phổ của màu pha được là ma trận A kích thước 31x1 và được tính bởi công thức

##### Chuyển đổi sang hệ màu RGB

*\* Không gian CIE XYZ:*

Từ phổ màu thu được, ta thực hiện tính toán 3 giá trị X, Y, Z tương ứng theo công thức:

Trong đó:

là hệ số chuẩn hóa sao cho *Y* = 1

 là hệ số phản xạ của màu tại bước sóng  tương ứng

, ,  là các hàm Observer tương ứng với X, Y, Z

 là hàm Illuminant đặc trưng cho màu đang được nhìn dưới ánh sáng nào, ví dụ D65 Illuminant là ánh sáng ban ngày.

Bảng sau cho biết giá trị của các hàm Observer và hàm D65 Illuminant.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bước sóng (*nm*)** |  |  |  |  |
| 400 | 0.0143 | 0.0004 | 0.0679 | 82.75 |
| 410 | 0.0435 | 0.0012 | 0.2074 | 91.49 |
| 420 | 0.1344 | 0.0040 | 0.6456 | 93.43 |
| 430 | 0.2839 | 0.0116 | 1.3856 | 86.68 |
| 440 | 0.3483 | 0.0230 | 1.7471 | 104.86 |
| 450 | 0.3362 | 0.0380 | 1.7721 | 117.01 |
| 460 | 0.2908 | 0.0600 | 1.6692 | 117.81 |
| 470 | 0.1954 | 0.0910 | 1.2876 | 114.86 |
| 480 | 0.0956 | 0.1390 | 0.8132 | 115.92 |
| 490 | 0.0320 | 0.2080 | 0.4652 | 108.81 |
| 500 | 0.0049 | 0.3230 | 0.2720 | 109.35 |
| 510 | 0.0093 | 0.5030 | 0.1582 | 107.8 |
| 520 | 0.0633 | 0.7100 | 0.0782 | 104.79 |
| 530 | 0.1655 | 0.8620 | 0.0422 | 107.69 |
| 540 | 0.2904 | 0.9540 | 0.0203 | 104.41 |
| 550 | 0.4334 | 0.9950 | 0.0087 | 104.05 |
| 560 | 0.5945 | 0.9950 | 0.0039 | 100 |
| 570 | 0.7621 | 0.9520 | 0.0021 | 96.33 |
| 580 | 0.9163 | 0.8700 | 0.0017 | 95.79 |
| 590 | 1.0263 | 0.7570 | 0.0011 | 88.69 |
| 600 | 1.0622 | 0.6310 | 0.0008 | 90.01 |
| 610 | 1.0026 | 0.5030 | 0.0003 | 89.6 |
| 620 | 0.8544 | 0.3810 | 0.0002 | 87.7 |
| 630 | 0.6424 | 0.2650 | 0 | 83.29 |
| 640 | 0.4479 | 0.1750 | 0 | 83.7 |
| 650 | 0.2835 | 0.1070 | 0 | 80.03 |
| 660 | 0.1649 | 0.0610 | 0 | 80.21 |
| 670 | 0.0874 | 0.0320 | 0 | 82.28 |
| 680 | 0.0468 | 0.0170 | 0 | 78.28 |
| 690 | 0.0227 | 0.0082 | 0 | 69.72 |
| 700 | 0.0114 | 0.0041 | 0 | 71.61 |

*\* Không gian CIE LAB:*

Sau khi tính ra 3 giá trị *X, Y, Z* của không gian CIE XYZ, ta tính được 3 giá trị *L\*, a\*, b\** của không gian CIE LAB như sau:

Trong đó:

Hàm

, ,  là 3 giá trị tham chiếu tương ứng với hàm Illuminant.

Với hàm D65 Illuminant, ra có (; ; ) = (0,9505; 1; 1.0888)

*\* Không gian RGB:*

Đầu tiên, ta tính 3 giá trị *r, g, b* thông qua giá trị *X, Y, Z* của không gian CIE XYZ như sau:

Có một điều kiện cần lưu ý là . Do đó, trong trường hợp ta tính ra hệ số nhỏ hơn 0 thì ta sẽ làm tròn bằng 0 và nếu lớn hơn 1 thì sẽ làm tròn bằng 1.

Sau khi tính *r, g, b*, ta có thể dễ dàng tính được giá trị của RGB (dải 0 – 255) với

Trong đó hàm

Lúc này, dựa vào 3 giá trị *R, G, B* ta đã có thể xác định được màu sắc cụ thể.

##### Lựa chọn bộ hệ số tỷ lệ thích hợp

Ở mục 1.2.3.2, ta đã đề cập đến việc chọn và thử tất cả các bộ số *a1, a2, a3, a4*thỏa mãn các điều kiện ràng buộc. Mỗi khi chọn được một bộ số thỏa mãn. Ta thực hiện các phép tính để xác định được phổ của màu pha được, từ đó tính toán chuyển đổi ra hệ màu RGB và so sánh với màu cần pha ban đầu. Tiếp tục chọn và thử cho đến khi tìm được các hệ số đảm bảo màu pha được giống với màu cần pha.

## KẾT QUẢ THỰC HIỆN

### Giao diện phần mềm

Graphical user interface, chart

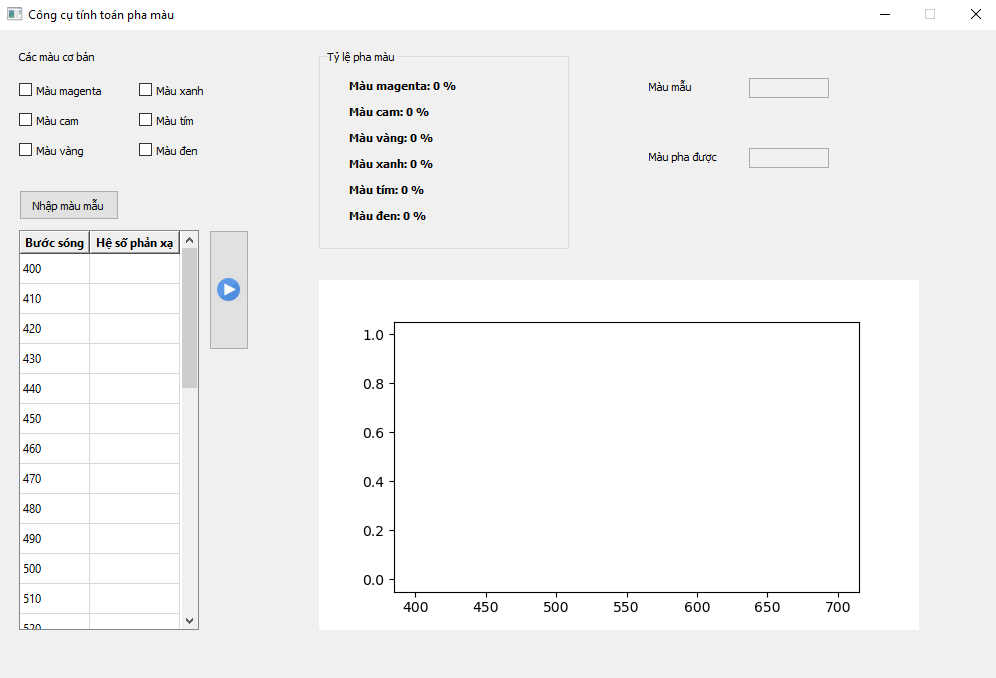
Description automatically generated

Hình 7

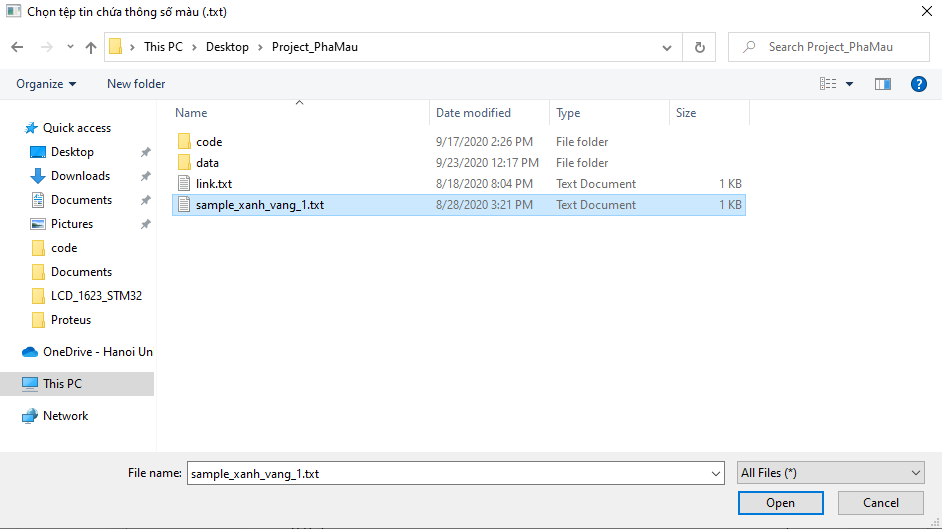
Sau quá trình phân tích, thiết kế và triển khai viết chương trình, ta có được một phiên bản phần mềm tính toán pha màu có giao diện như Hình 7.

### Hướng dẫn sử dụng

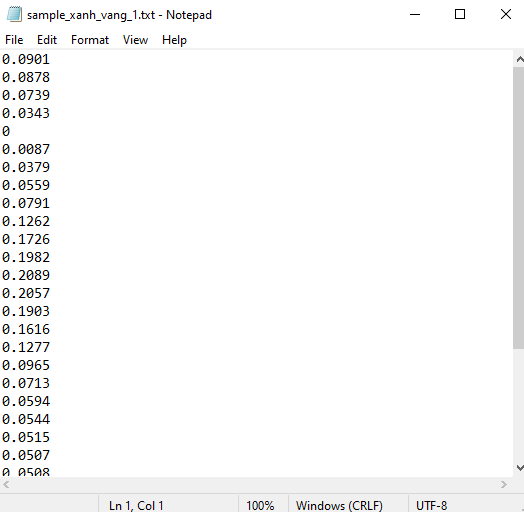
Sau khi khởi động, giao diện ban đầu của phần mềm như sau:



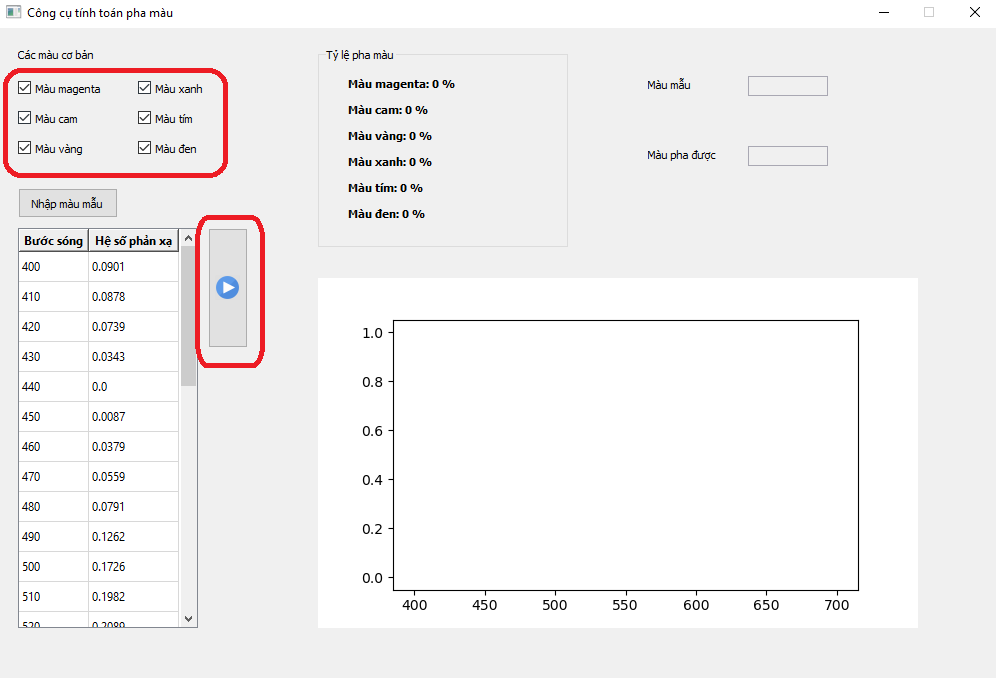
Nhấn vào nút “Nhập màu mẫu”, một hộp thoại xuất hiện. Trong hộp thoại này, ta chọn file chứa dữ liệu về phổ của màu cần pha.



File dữ liệu có dạng một file văn bản gồm các số là hệ số phản xạ tại các bước sóng trong dải 400 – 700 *nm* của màu cần pha.



Sau khi nhập dữ liệu phổ của màu cần pha, ta tick vào các checkbox để chọn những màu cơ bản sẽ dùng để pha rồi nhấn nút có biểu tượng “Run” để phần mềm thực hiện xử lý và tính toán.



Sau một khoảng thời gian xử lý tính toán, kết quả đầu ra sẽ hiện lên giao diện của phần mềm gồm có: hệ số tỷ lệ của các màu cơ bản thành phần, biểu đồ phổ so sánh giữa màu cần pha và màu pha được, màu sắc mà mắt ta sẽ nhìn thấy của 2 màu.

