# MỞ ĐẦU

**Lý do chọn đề tài**

**Tổng quan đề tài**

**Mục đích nghiên cứu**

**Đối tượng và phạm vi của đề tài**

**Phương pháp nghiên cứu**

**Đóng góp của đề tài**

**Kết cấu của đề tài**

# NỘI DUNG

## CƠ SỞ LÝ THUYẾT

### Hệ màu RGB

RGB (Red-Green-Blue) sử dụng mô hình màu bổ sung trong đó ánh sáng đỏ, xanh lá cây và xanh lam được tổ hợp với nhau theo nhiều phương thức khác nhau để tạo thành các màu sắc khác nhau.

#### Cơ sở sinh học

Các màu gốc có liên quan đến các khái niệm sinh học hơn là vật lý, nó dựa trên cơ sở phản ứng sinh lý học của mắt người đối với ánh sáng. Mắt người có các tế bào cảm quang có hình nón nên còn được gọi là tế bào hình nón, các tế bào này thông thường có phản ứng cực đại với ánh sáng vàng - xanh lá cây (tế bào hình nón L), xanh lá cây (tế bào hình nón M) và xanh lam (tế bào hình nón S) tương ứng với các bước sóng khoảng 564 nm, 534 nm và 420 nm. Ví dụ, màu vàng thấy được khi các tế bào cảm nhận màu xanh ánh vàng được kích thích nhiều hơn một chút so với tế bào cảm nhận màu xanh lá cây và màu đỏ cảm nhận được khi các tế bào cảm nhận màu vàng - xanh lá cây được kích thích nhiều hơn so với tế bào cảm nhận màu xanh lá cây.

Mặc dù biên độ cực đại của các phản xạ của các tế bào cảm quang không diễn ra ở các bước sóng của màu “đỏ”, “xanh lục” và “xanh lam”, ba màu này được mô tả như là các màu gốc vì chúng có thể sử dụng một cách tương đối độc lập để kích thích ba loại tế bào cảm quang.

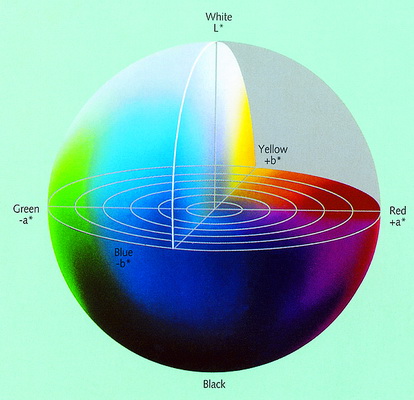
Để sinh ra khoảng màu tối ưu cho các loài động vật khác, các màu gốc khác có thể được sử dụng. Với các loài vật có bốn loại tế bào cảm quang, chẳng hạn như nhiều loại chim, người ta có lẽ phải nói là cần tới bốn màu gốc; cho các loài vật chỉ có hai loại tế bào cảm quang, như phần lớn các loại động vật có vú, thì chỉ cần hai màu gốc.

#### Hiển thị màu RGB

Một trong những ứng dụng phổ biến nhất của mô hình màu RGB là việc hiển thị màu sắc trong các ống tia âm cực, màn hình tinh thể lỏng hay màn hình plasma, chẳng hạn như màn hình máy tính hay ti vi. Mỗi điểm ảnh trên màn hình có thể được thể hiện trong bộ nhớ máy tính như là các giá trị độc lập của màu đỏ, xanh lá cây và xanh lam. Các giá trị này được chuyển đổi thành các cường độ và gửi tới màn hình. Bằng việc sử dụng các tổ hợp thích hợp của các cường độ ánh sáng đỏ, xanh lục và xanh lam, màn hình có thể tái tạo lại phần lớn các màu trong khoảng đen và trắng. Các phần cứng hiển thị điển hình được sử dụng cho các màn hình máy tính trong năm 2003 sử dụng tổng cộng 24 bit thông tin cho mỗi điểm ảnh. Nó tương ứng với mỗi 8 bit cho màu đỏ, xanh lá cây và xanh lam, tạo thành một tổ hợp 256 các giá trị có thể, hay 256 mức cường độ cho mỗi màu. Với hệ thống như thế, khoảng 16,7 triệu màu rời rạc có thể tái tạo.

### Không gian màu CIELAB

Hệ màu CIELAB (L\*a\*b\*) được xây dựng dựa trên khả năng cảm nhận màu của mắt người. Các giá trị Lab mô tả tất cả những màu mà mắt một người bình thường có thể nhìn thấy được. Lab được xem là một mô hình màu độc lập đối với thiết bị và thường được sử dụng như một cơ sở tham chiếu khi chuyển đổi một màu từ một không gian màu này sang một không gian màu khác.



Hình

Theo hệ màu Lab:

* △L có giá trị càng lớn ý nói màu trên mẫu nghiên về màu trắng.
* △L có giá trị càng bé ý nói màu trên mẫu nghiên về màu đen.
* △a có giá trị càng lớn ý nói màu trên mẫu nghiên về màu đỏ.
* △a có giá trị càng bé ý nói màu trên mẫu nghiên về màu xanh.
* △b có giá trị càng lớn ý nói màu trên mẫu nghiên về màu vàng.
* △b có giá trị càng bé ý nói màu trên mẫu nghiên về màu xanh.

### Ngôn ngữ lập trình Python

Python là ngôn ngữ lập trình hướng đối tượng, cấp cao, mạnh mẽ, được tạo ra bởi Guido van Rossum. Nó dễ dàng để tìm hiểu và đang nổi lên như một trong những ngôn ngữ lập trình nhập môn tốt nhất cho người lần đầu tiếp xúc với ngôn ngữ lập trình. Python hoàn toàn tạo kiểu động và sử dụng cơ chế cấp phát bộ nhớ tự động. Python có cấu trúc dữ liệu cấp cao mạnh mẽ và cách tiếp cận đơn giản nhưng hiệu quả đối với lập trình hướng đối tượng. Cú pháp lệnh của Python là điểm cộng vô cùng lớn vì sự rõ ràng, dễ hiểu và cách gõ linh động làm cho nó nhanh chóng trở thành một ngôn ngữ lý tưởng để viết script và phát triển ứng dụng trong nhiều lĩnh vực, ở hầu hết các nền tảng.

#### Lịch sử phát triển của Python

Python được ra đời vào cuối những năm 1980 bởi Guido van Rossum và được phát hành lần đầu tiên vào tháng 2 năm 1991. Vào cuối những năm 1980, Guido van Rossum làm việc trong Amoeba, phân phối một nhóm hệ điều hành. Ông muốn sử dụng một ngôn ngữ thông dịch như ABC để truy cập vào những cuộc gọi hệ thống Amoeba. Vì vậy, ông quyết định tạo ra một ngôn ngữ mở rộng, từ đó một ngôn ngữ lập trình mới được thiết kế và chính là Python sau này.

Từ khi được phát hành đến nay, Python đã trải qua rất nhiều phiên bản và không ngừng được nâng cấp. Bảng 1 là các phiên bản Python đã phát hành cho đến nay.

Bảng Các phiên bản đã phát hành của Python

|  |  |
| --- | --- |
| **Phiên bản** | **Ngày phát hành** |
| Python 1.0 (bản phát hành chuẩn đầu tiên) | 1/1994 |
| Python 1.6 (phiên bản 1.x cuối cùng) | 5/9/2000 |
| Python 2.0 (giới thiệu list comprehension) | 16/10/2000 |
| Python 2.7 (phiên bản 2.x cuối cùng) | 3/7/2010 |
| Python 3.0 (loại bỏ cấu trúc và module trùng lặp) | 3/12/2008 |
| Python 3.8.5 (bản mới nhất hiện nay) | 20/7/2020 |

#### Các đặc điểm chính của Python

* *Ngôn ngữ lập trình đơn giản, dễ học*: Python có cú pháp rất đơn giản, rõ ràng. Nó dễ đọc và viết hơn rất nhiều khi so sánh với những ngôn ngữ lập trình khác như C++, Java, C#. Python làm cho việc lập trình trở nên thú vị, cho phép bạn tập trung vào những giải pháp chứ không phải cú pháp.
* *Miễn phí, mã nguồn mở*: Bạn có thể tự do sử dụng và phân phối Python, thậm chí là dùng nó cho mục đích thương mại. Vì là mã nguồn mở, bạn không những có thể sử dụng các phần mềm, chương trình được viết trong Python mà còn có thể thay đổi mã nguồn của nó. Python có một cộng đồng rộng lớn, không ngừng cải thiện nó mỗi lần cập nhật.
* *Khả năng di chuyển*: Các chương trình Python có thể di chuyển từ nền tảng này sang nền tảng khác và chạy nó mà không có bất kỳ thay đổi nào. Nó chạy liền mạch trên hầu hết tất cả các nền tảng như Windows, macOS, Linux.
* *Khả năng mở rộng và có thể nhúng*: Giả sử một ứng dụng đòi hỏi sự phức tạp rất lớn, bạn có thể dễ dàng kết hợp các phần code bằng C, C++ và những ngôn ngữ khác (có thể gọi được từ C) vào code Python. Điều này sẽ cung cấp cho ứng dụng của bạn những tính năng tốt hơn cũng như khả năng scripting mà những ngôn ngữ lập trình khác khó có thể làm được.
* *Ngôn ngữ thông dịch cấp cao*: Không giống như C/C++, với Python, bạn không phải lo lắng những nhiệm vụ khó khăn như quản lý bộ nhớ, dọn dẹp những dữ liệu vô nghĩa,... Khi chạy code Python, nó sẽ tự động chuyển đổi code sang ngôn ngữ máy tính có thể hiểu. Bạn không cần lo lắng về bất kỳ hoạt động ở cấp thấp nào.
* *Thư viện tiêu chuẩn lớn để giải quyết những tác vụ phổ biến*: Python có một số lượng lớn thư viện tiêu chuẩn giúp cho công việc lập trình của bạn trở nên dễ thở hơn rất nhiều, đơn giản vì không phải tự viết tất cả code. Ví dụ: Bạn cần kết nối cơ sở dữ liệu MySQL trên Web server? Bạn có thể nhập thư viện MySQLdb và sử dụng nó. Những thư viện này được kiểm tra kỹ lưỡng và được sử dụng bởi hàng trăm người. Vì vậy, bạn có thể chắc chắn rằng nó sẽ không làm hỏng code hay ứng dụng của mình.
* *Hướng đối tượng*: Mọi thứ trong Python đều là hướng đối tượng. Lập trình hướng đối tượng (OOP) giúp giải quyết những vấn đề phức tạp một cách trực quan. Với OOP, bạn có thể phân chia những vấn đề phức tạp thành những tập nhỏ hơn bằng cách tạo ra các đối tượng.

#### Ứng dụng của Python

* *Lập trình ứng dụng web:* Ta có thể tạo web app có khả năng mở rộng (scalable) được bằng cách sử dụng framework và CMS (Hệ thống quản trị nội dung) được tích hợp trong Python. Một vài nền tảng phổ biến để tạo web app là: Django, Flask, Pyramid, Plone, Django CMS. Các trang như Mozilla, Reddit, Instagram và PBS đều được viết bằng Python.
* *Khoa học và tính toán:* Có nhiều thư viện trong Python cho khoa học và tính toán số liệu, như SciPy và NumPy, được sử dụng cho những mục đích chung chung trong tính toán. Và, có những thư viện cụ thể như: EarthPy cho khoa học trái đất, AstroPy cho Thiên văn học,... Ngoài ra, Python còn được sử dụng nhiều trong machine learning, khai thác dữ liệu và deep learning.
* *Tạo nguyên mẫu phần mềm:* Python chậm hơn khi so sánh với các ngôn ngữ được biên dịch như C++ và Java. Nó có thể không phải là lựa chọn tốt nếu nguồn lực bị giới hạn và yêu cầu về hiệu quả là bắt buộc. Tuy nhiên, Python là ngôn ngữ tuyệt vời để tạo những nguyên mẫu (bản chạy thử - prototype). Ví dụ, bạn có thể sử dụng Pygame (thư viện viết game) để tạo nguyên mẫu game trước. Nếu thích nguyên mẫu đó có thể dùng C++ để viết game thực sự.

#### Một số cú pháp cơ bản của Python

##### Các kiểu dữ liệu

Sử dụng Python, ta không cần phải khai báo biến. Biến được xem là đã khai báo nếu nó được gán một giá trị lần đầu tiên. Căn cứ vào mỗi lần gán, Python sẽ tự động xác định kiểu dữ liệu của biến. Python có một số kiểu dữ liệu thông dụng sau:

* *int, long*: số nguyên (trong phiên bản 3.x long được nhập vào trong kiểu int). Độ dài của kiểu số nguyên là tùy ý, chỉ bị giới hạn bởi bộ nhớ máy tính.
* *float*: số thực
* *complex*: số phức, chẳng hạn 5+4j
* *list*: dãy trong đó các phần tử của nó có thể được thay đổi, chẳng hạn [8, 2, 'b', -1.5]. Kiểu dãy khác với kiểu mảng (array) thường gặp trong các ngôn ngữ lập trình ở chỗ các phần tử của dãy không nhất thiết có kiểu giống nhau. Ngoài ra phần tử của dãy còn có thể là một dãy khác.
* *tuple*: dãy trong đó các phần tử của nó không thể thay đổi.
* *str*: chuỗi ký tự. Từng ký tự trong chuỗi không thể thay đổi. Chuỗi ký tự được đặt trong dấu nháy đơn, hoặc nháy kép.
* *dict*: từ điển, còn gọi là “hashtable”: là một cặp các dữ liệu được gắn theo kiểu {*từ khóa: giá trị*}, trong đó các từ khóa trong một từ điển nhất thiết phải khác nhau. Chẳng hạn {1: "Python", 2: "Pascal"}
* *set*: một tập không xếp theo thứ tự, ở đó, mỗi phần tử chỉ xuất hiện một lần.

##### Cấu trúc rẽ nhánh

Có 3 dạng cấu trúc rẽ nhánh thường được sử dụng.

* Dạng 1:

|  |
| --- |
| if điều\_kiện:  # lệnh... |

* Dạng 2:

|  |
| --- |
| if điều\_kiện:  # lệnh...  else:  # lệnh... |

* Dạng 3:

|  |
| --- |
| if điều\_kiện\_1:  # lệnh... (được thực hiện nếu điều\_kiện\_1 là đúng)  elif điều\_kiện\_2:  # lệnh... (được thực hiện nếu điều\_kiện\_1 là sai, nhưng điều\_kiện\_2 là đúng)  else:  # lệnh... (được thực hiện nếu tất cả các biểu thức điều kiện đi kèm if và elif đều sai) |

##### Cấu trúc lặp

##### Có 2 dạng vòng lặp thường được sử dụng.

##### Dạng 1: lặp với số lần chưa biết trước

|  |
| --- |
| while biểu\_thức\_đúng:  # lệnh... |

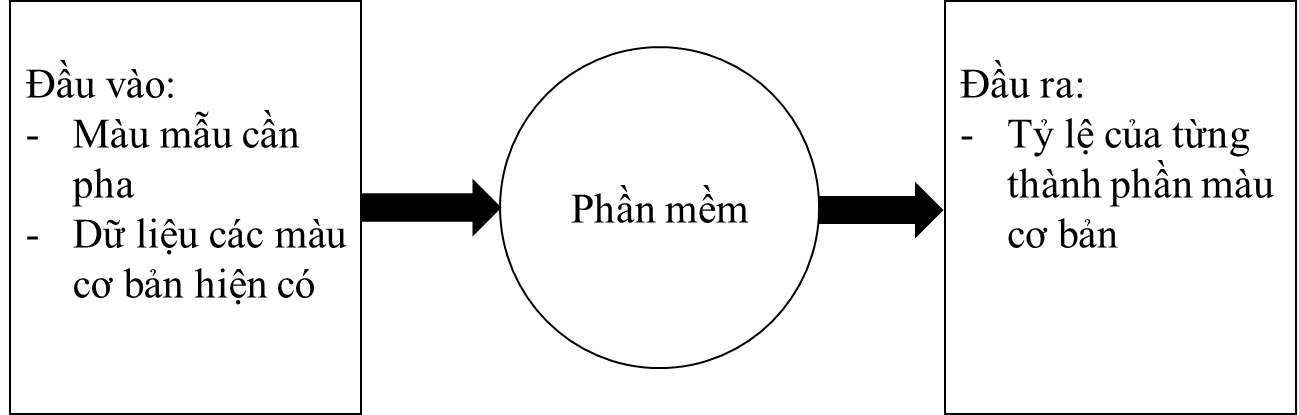
* Dạng 2: lặp với số lần đã biết

|  |
| --- |
| for phần\_tử in dãy:  # lệnh... |

## 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.



#### Dữ liệu đầu vào

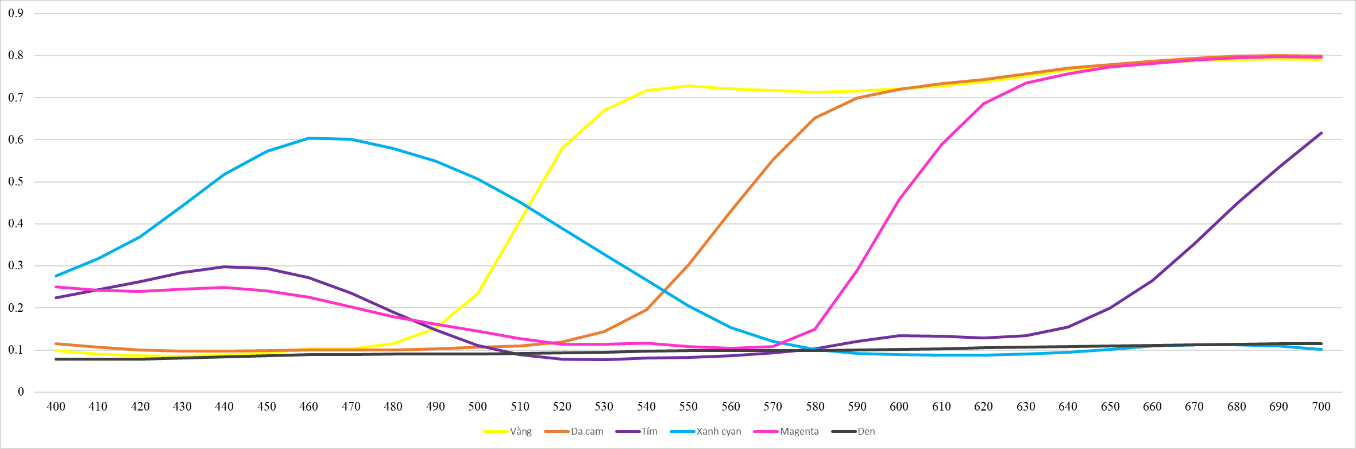
*\* 6 màu cơ bản:*

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 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 |

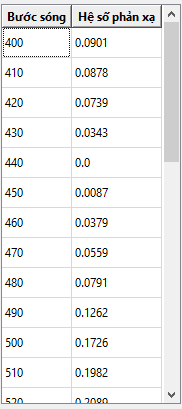
Hình 2 biểu diễn trực quan đường phổ của 6 màu cơ bản.



Hình Đường phổ của 6 màu cơ bản

*\* Màu cần pha:*

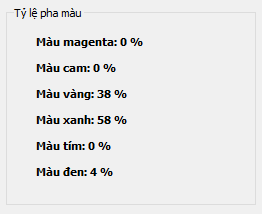
Dữ liệu về màu cần pha cũng được cho ở dạng hệ số phản xạ của màu đó tương ứng với các bước sóng trong dải 400 *nm* đến 700 *nm*.



#### 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

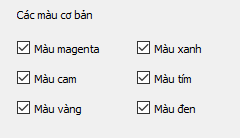
### Triển khai chi tiết

#### Thiết kế giao diện phần mềm

Ta sử dụng thư viện PyQT của ngôn ngữ lập trình Python để lập trình giao diện cho phần mềm.

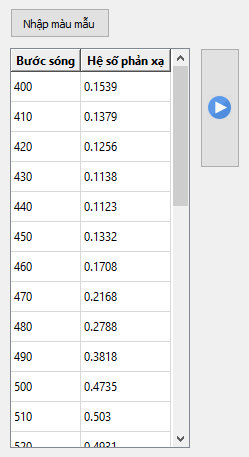
QT là một thư viện mã nguồn mở cho phép lập trình viên nhanh chóng tạo ra giao diện phần mềm với các thành phần cơ bản như form, button, edit text, label text,… Trong đó PyQT là thư viện dành cho ngôn ngữ lập trình Python.

##### Giao diện chọn màu cơ bản



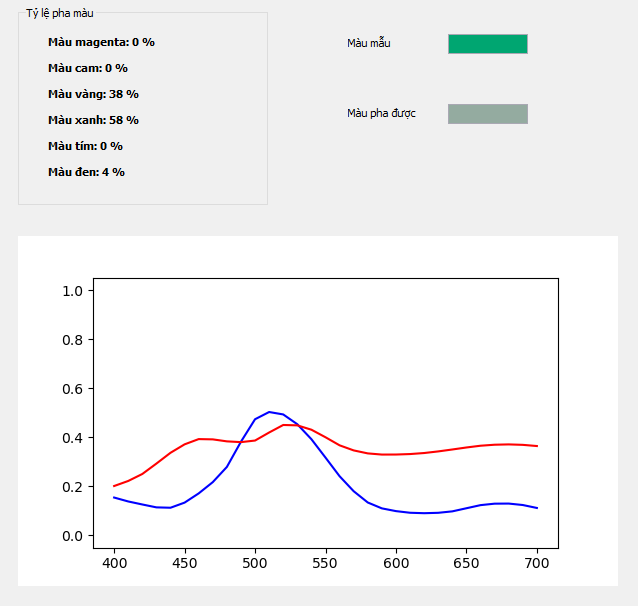
Giao diện này gồm 6 checkbox tương ứng với 6 màu cơ bản. Người dùng tick vào các checkbox tương ứng để chọn/bỏ chọn màu sẽ sử dụng để pha.

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



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*.

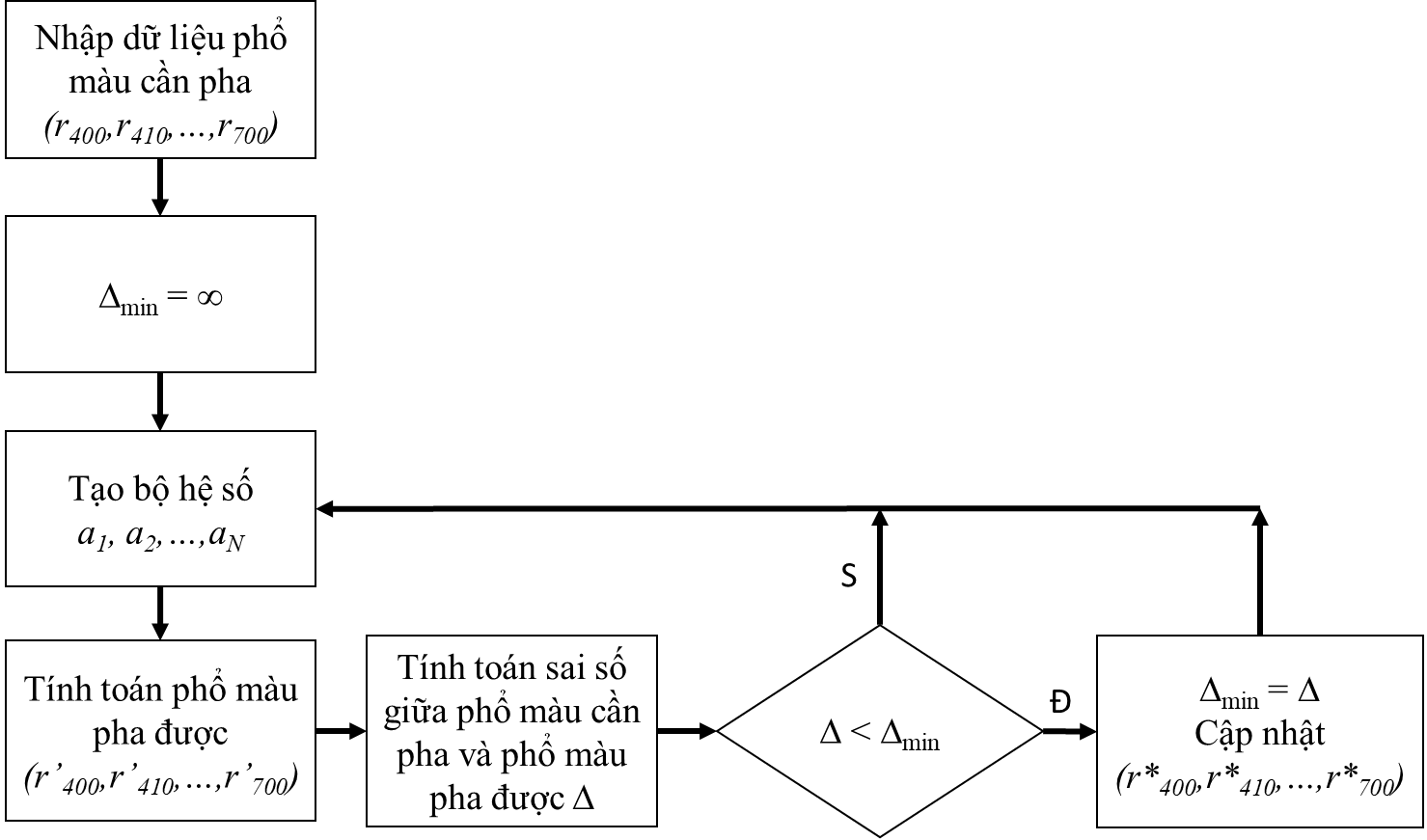
##### Giao diện hiển thị dữ liệu đầu ra



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

##### Tạo bộ hệ số tỷ lệ

Ở bước này, ta sử dụng thuật toán đệ quy quay lui (vét cạn) để tìm tất cả các bộ hệ số *a1, a2,…, a6* sao cho thỏa mãn:

Với mỗi bộ số *a1, a2,…, a6*  tìm được, kết hợp với phổ của 6 màu cơ bản, ta tính được phổ của màu pha được.

##### Tính toán phổ màu pha được

Ta biểu diễn phổ của 6 màu cơ bản dưới dạng ma trận có kích thước 31x6, trong đó 31 hàng tương ứng với các bước sóng 400, 410, 420, …, 700 *nm*. 6 cột tương ứng với 6 màu cơ bản.



Các hệ số *a1, a2,…, a6* được biểu diễn dưới dạng ma trận kích thước 6x1.



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 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 

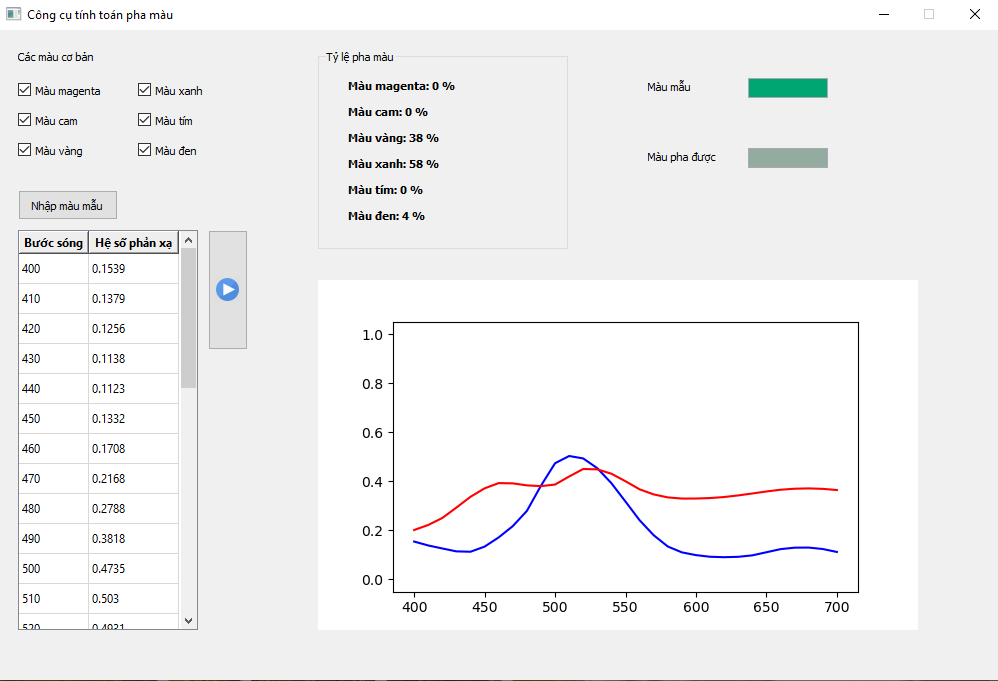
Lúc này, dựa vào 3 giá trị *R, G, B* ta đã có thể xác định được màu sắc cụ thể.

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

Ở mục 2.2.2.2, ta đã đề cập đến việc chọn và thử tất cả các bộ số *a1, a2,…, a6*thỏa mãn 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

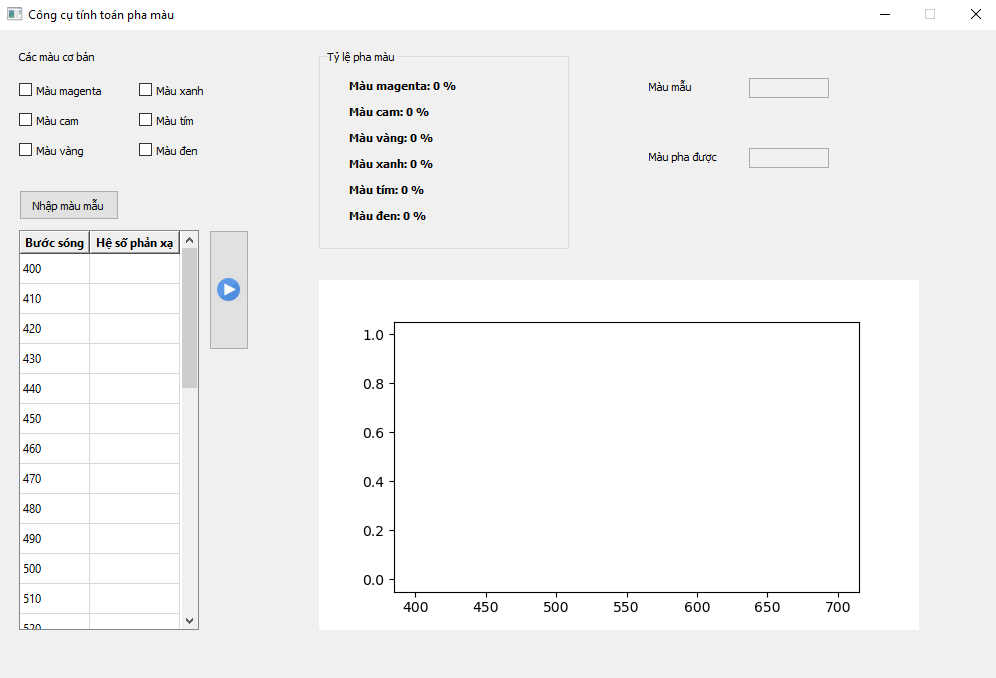


Hình

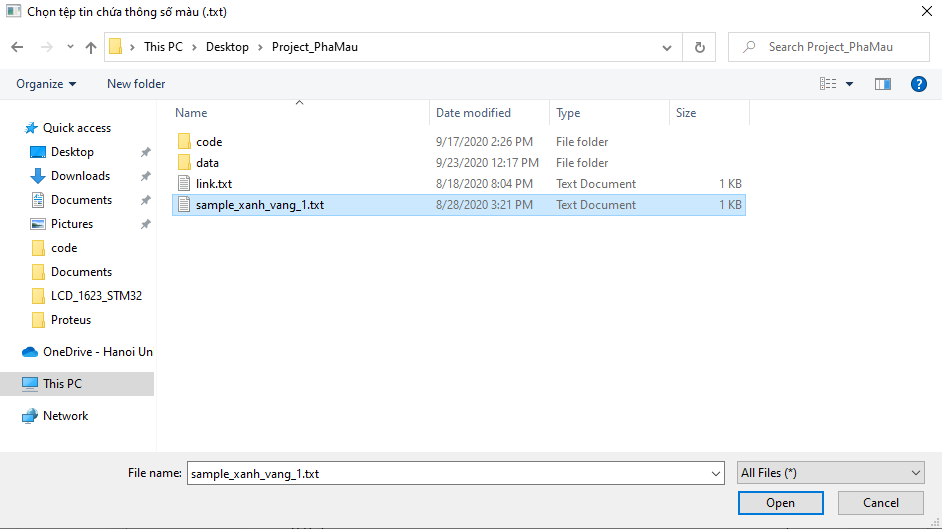
Sau quá trình phân tích, thiết kế và triển khai, sản phẩm thu được là phần mềm tính toán pha màu có giao diện như Hình 5.

### Hướng dẫn sử dụng

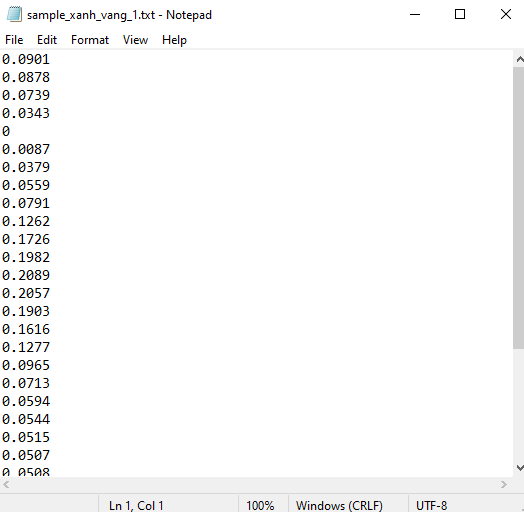
Sau khi khởi động, giao diện ban đầu của phần mềm như sau:



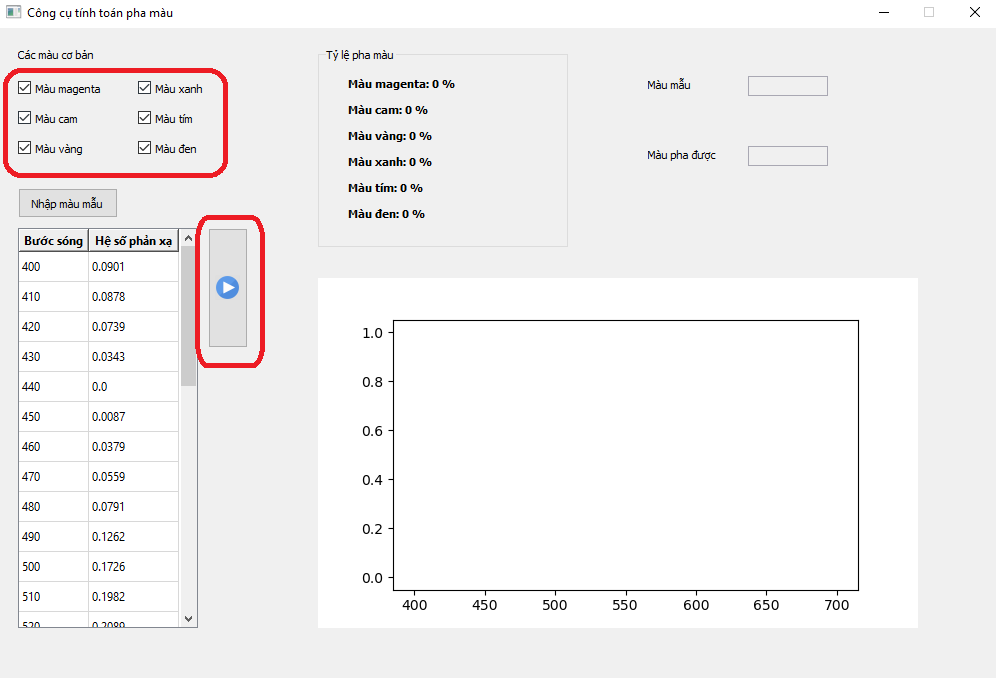
Nhấn vào nút “Nhập màu mẫu”, một hộp thoại xuất hiện. Trong hộp thoại này, ta chọn file chứa dữ liệu về phổ của màu cần pha.



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.

