

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Сумський державний університет**  
Факультет електроніки та інформаційних технологій  
Кафедра комп'ютерних наук

«До захисту допущено»

В.о. завідувача кафедри

\_\_\_\_\_ Ігор ШЕЛЕХОВ  
(підпис)

**КВАЛІФІКАЦІЙНА РОБОТА**  
**на здобуття освітнього ступеня бакалавра**

зі спеціальності 122 - Комп'ютерних наук,  
освітньо-професійної програми «Інформатика»  
на тему: «Інформаційне та програмне забезпечення системи оцінювання знань з  
дистанційного курсу «Обробка зображень та мультимедіа»  
здобувача групи ІН-93 Рябущенка Андрія Андрійовича

Кваліфікаційна робота містить результати власних досліджень.  
Використання ідей, результатів і текстів інших авторів мають посилання на  
відповідне джерело.

\_\_\_\_\_ Андрій РЯБУЩЕНКО  
(підпис)

Керівник,  
в.о. завідувача кафедри,  
кандидат технічних наук, доцент

Ігор ШЕЛЕХОВ

\_\_\_\_\_ (підпис)

**Суми – 2023**

**Сумський державний університет**  
Факультет електроніки та інформаційних технологій  
Кафедра комп'ютерних наук

«Затверджую»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

\_\_\_\_\_ (підпис)

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

### на здобуття освітнього ступеня бакалавр

зі спеціальності 122 - Комп'ютерних наук, освітньо-професійної програми «Інформатика»  
здобувача групи ІН.м-11н Рябуценка Андрія Андрійовича

1. Тема роботи: «Інформаційне та програмне забезпечення системи оцінювання знань з дистанційного курсу «Обробка зображень та мультимедіа»»  
затверджую наказом по СумДУ від \_\_\_\_\_
2. Термін здачі здобувачем кваліфікаційної роботи \_\_\_\_\_
3. Вхідні дані до кваліфікаційної роботи \_\_\_\_\_
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)  
*1) Аналіз проблеми предметної області, постановка й формування завдань роботи.*  
*2) Вибір методу розв'язання задачі 3) Інформаційне та програмне забезпечення.*
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) \_\_\_\_\_
6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання « \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р.

Завдання прийняв до виконання \_\_\_\_\_  
(підпис)

Керівник \_\_\_\_\_  
(підпис)

## КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	<i>Аналіз проблеми предметної області, постановка й формування завдань роботи</i>		
2	<i>Вибір методу розв'язання задачі</i>		
3	<i>Інформаційне та програмне забезпечення</i>		
4	<i>Аналіз отриманих результатів</i>		
5	<i>Оформлення пояснювальної записки до кваліфікаційної роботи</i>		

Здобувач вищої освіти \_\_\_\_\_  
(підпис)

Керівник \_\_\_\_\_  
(підпис)

## АНОТАЦІЯ

**Записка:** 37 стор., 9 рис., 1 табл., 1 додаток, 21 джерел.

**Обґрунтування актуальності теми роботи** – Тема кваліфікаційної роботи є актуальною, оскільки присвячена розв’язанню важливої практичної задачі розробки складових системи керування дистанційним навчанням.

**Об’єкт дослідження** — процес оцінювання знань здобувачів вищої освіти дистанційної форми навчання.

**Мета роботи** — розробка інформаційного і програмного забезпечення модуля оцінювання знань здобувачів вищої освіти дистанційної форми навчання.

**Методи дослідження** — методи та засоби інформаційно-комунікаційних технологій в освітній діяльності, методи оцінювання знань.

**Результати** — розроблено інформаційне і програмне забезпечення модуля оцінювання знань здобувачів вищої освіти дистанційної форми навчання. Тестування працездатності розробки виконано для дистанційного курсу «Обробка зображень та мультимедіа»

ТЕСТУВАННЯ, КОЛЬОРОВІ МОДЕЛІ, КОЛЬОРОВІ ПРОСТОРИ

## ЗМІСТ

ВСТУП .....	5
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....	6
1.1 Сучасні системи керування дистанційним навчанням.....	6
1.2 Кольорові моделі та колірні простори .....	6
1.3 Постановка задачі.....	16
2 ВИБІР МЕТОДУ РІШЕННЯ .....	17
2.1 Інформаційна модель .....	17
2.2 Розробка компонентів для візуалізації і інтерактивної взаємодії з колірними моделями .....	18
2.3 Розробка завдань для оцінки знань .....	21
3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ .....	27
3.1 Засоби для програмної реалізації додатку .....	27
3.2 Опис програмної реалізації .....	27
3.3 Тестування .....	29
ВИСНОВКИ.....	33
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	34
ДОДАТОК.....	36

## ВСТУП

**Актуальність.** Розуміння цифрового кольору важливе для широкого кола людей та їхніх професій, зокрема для фотографів, ілюстраторів, художників комп'ютерної графіки, веб-розробників, а також а також людей, що працюють у сфері візуалізації даних або над людино-машинними інтерфейсами. Розуміння взаємозв'язку між колірними моделями і, в ідеалі, вміння конвертувати між ними має переваги, коли людина має справу з числовими представленнями кольорів. Наприклад, при читанні CSS-файлу навряд чи виникне потреба вводити кожне значення в палітру кольорів або використовувати спеціалізоване інтегроване середовище розробки (IDE), щоб отримати уявлення про те, як може виглядати кожен колір. Крім того, можливим недоліком для програмістів, які знають лише модель RGB, наприклад, є тенденція до вибору перебільшених, яскравих і насичених кольорів.

**Об'єкт дослідження.** Процес оцінювання знань здобувачів вищої освіти дистанційної форми навчання.

**Предмет дослідження.** Методологія оцінювання знань здобувачів вищої освіти дистанційної форми навчання.

**Гіпотеза.** Підвищення ефективності навчального процесу за дистанційною формою навчанням можна досягнути шляхом включення в склад системи керування дистанційним навчанням елементів, що дозволяють виконувати оцінювання знань здобувачів в автоматичному режимі.

**Новизна.** В роботі запропоновано інформаційне і програмне забезпечення модуля оцінювання знань здобувачів вищої освіти дистанційної форми навчання з урахуванням особливостей дистанційного курсу «Обробка зображень та мультимедіа» .

**Структура.** Дане робота складається зі вступу, аналізу публікацій, постановки задачі дослідження, вибір методики та інструментів для рішення поставленої проблеми, опису програмного забезпечення інформаційної системи, висновків, списку використаних джерел та додатків.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Сучасні системи керування дистанційним навчанням

Сучасні системи керування дистанційним навчанням (СКДН) як навчальні інструменти мають кілька очевидних потенційних переваг. Порівняно зі звичайним, не веб-орієнтованим забезпеченням навчального процесу, вони мають перевагу в мобільності. Окрім сучасного веб-браузера, не потрібно встановлювати жодного додаткового програмного забезпечення чи плагінів. Завдяки HTML5 і WebGL, навіть 3D-візуалізації, подібні до тих, що заплановані для цього проекту, можна запускати в браузері. Менша кількість вимог до попередньо встановленого програмного забезпечення може зробити СКДН доступною для більшої групи людей з різними операційними системами та пристроями. За умови, що СКДН оптимізовано для мобільних пристроїв, її портативність стає ще вищою. Оскільки смартфони часто є більш доступними, ніж повнорозмірні комп'ютери, для людей зі смартфонами є більше можливостей відкрити мобільні додатки на основі веб-технологій з наміром навчатися. Ще однією великою перевагою веб-додатків і комп'ютерних програм загалом є можливість представляти візуалізації у 2D або 3D. Особливо складні теми можуть виграти від інтерактивних, а не статичних візуалізацій або від симуляцій.

## 1.2 Кольорові моделі та колірні простори

Фізично колір складається з видимого світла, яке є електромагнітним випромінюванням з довжинами хвиль від 380 до 700 нм. Чисті спектральні кольори, відомі як кольори веселки, складаються лише з однієї довжини хвилі світла. Тому всі інші кольори, включаючи, наприклад, білий і рожевий, повинні бути сумішшю декількох довжин хвиль. Фізіологічно люди з нормальним кольоровим зором сприймають колір за допомогою трьох різних типів колбочок в очах. Ці колбочки відрізняються за чутливістю до певних довжин хвиль, досягаючи піку приблизно при 419 нм, 531 нм і 558 нм, які приблизно відповідають синьому, зеленому і червоному кольорам. У літературі терміни "колірна модель", "колірний простір" і "колірна система" іноді

використовуються як взаємозамінні [1-5]. У більш точному визначенні, колірна модель описує абстрактний спосіб, у який кольори можуть бути представлені за допомогою ряду компонентів, наприклад, червоного, зеленого і синього. За цим визначенням, система може називатися колірним простором, тільки якщо така модель поєднується з визначенням того, як інтерпретувати компоненти. Наприклад, у прикладі колірної моделі з компонентами для червоного, зеленого та синього, один колірний простір може визначати значення зеленого, що дорівнює 1, а всі інші компоненти дорівнюють 0, щоб бути трохи більш насиченим і трохи більш жовтим, ніж інший. У цій роботі термін "система кольорів" буде використовуватися як синонім як для колірної моделі, так і для колірного простору (рис. 1.1). Якщо не вказано інше, параметри компонентів завжди вважатимуться такими, що знаходяться в інтервалі  $[0,1]$ .

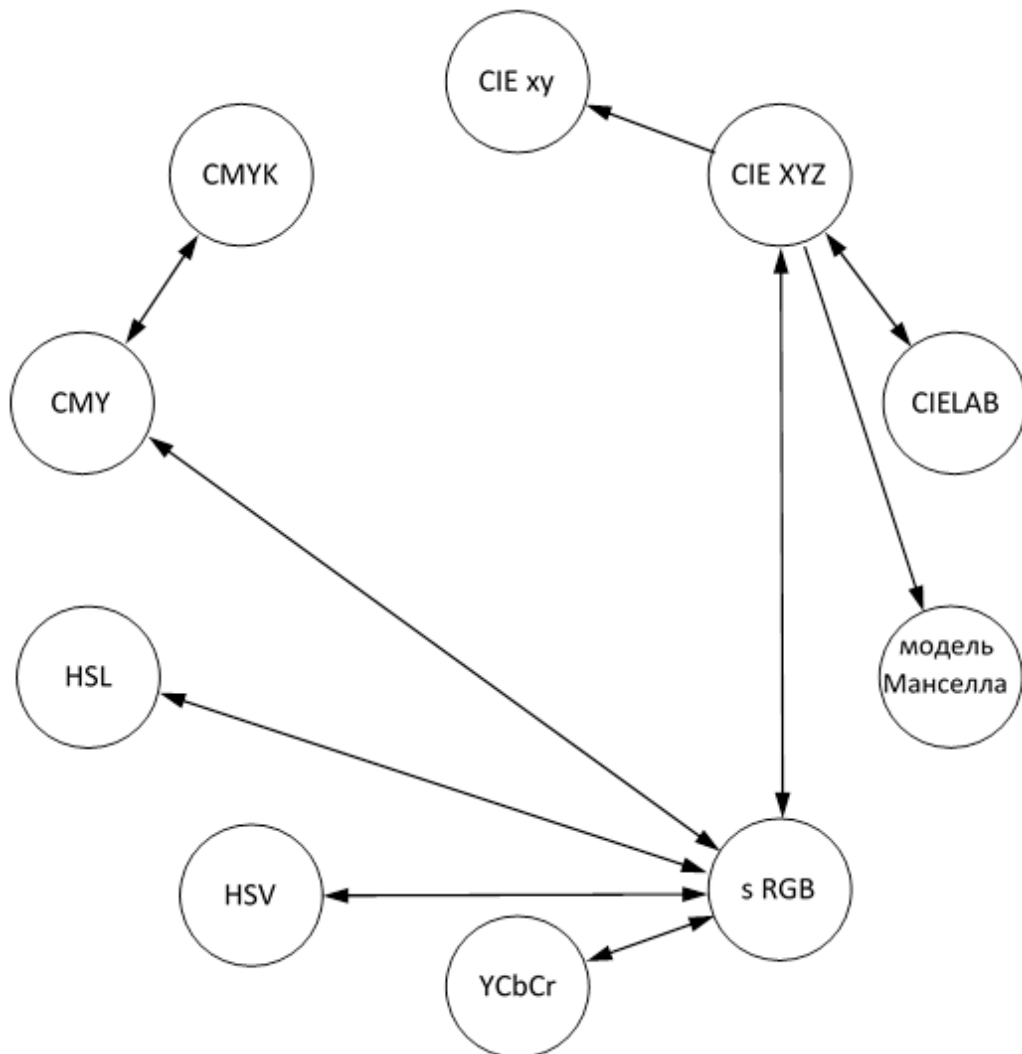


Рисунок 1.1 – Орієнтований граф взаємоперетворення колірних просторів

У наступних розділах буде надано огляд деяких важливих систем кольорів. Для кожної системи буде наведено принаймні один метод перетворення в іншу систему кольорів та з неї. Ці відповідні інші системи вибрано так, щоб у кінцевому підсумку можна було здійснити перетворення між будь-якими двома з представлених систем кольорів. Починаючи з довільного колірному простору (за винятком СІЕ ху і Манселла), на рисунку 1.1 показано, які перетворення необхідні для отримання бажаного іншого колірному простору [1,2].

Щоб порівняти різні колірні простори, корисно подивитися на їхні відповідні гами. Гама колірному простору - це набір кольорів, які він може представляти. Гама часто візуалізують за допомогою так званих підковоподібних діаграм або діаграм кольоровості, як показано на рисунку 2.2.

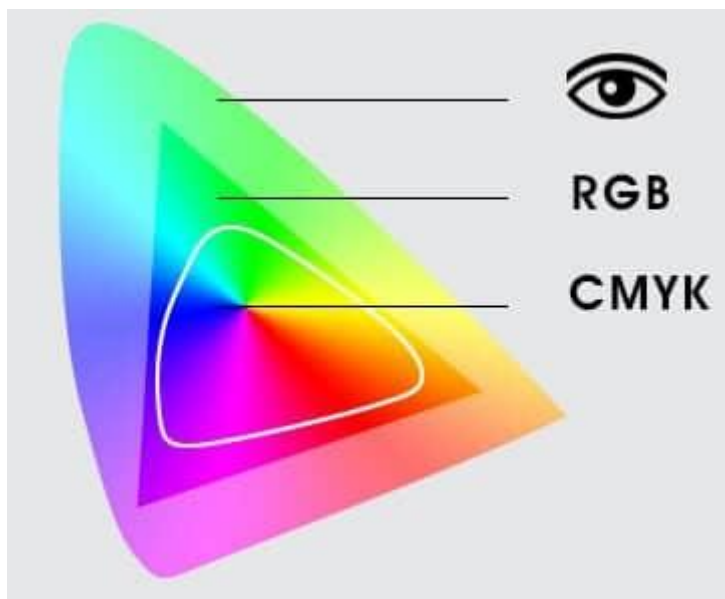


Рисунок 1.2 – Діаграми кольоровості [5]

Крива лінія навколо підкови представляє всі монохроматичні (спектральні) кольори, видимі людським оком. Кожен інший видимий колір є комбінацією щонайменше двох різних довжин хвиль і лежить в межах замкненої фігури. Для того, щоб зрозуміти цю проекцію, необхідно розглянути вимірний колірний простір XYZ, який базується на експериментах з підбору кольорів, опублікованих незалежно Вільямом Д. Райтом у 1929 році та Джоном Гільдом у



1931 році[6]. Піддослідних просили відтворити кольори спектра, змінюючи інтенсивність трьох основних кольорів. Райт і Гільд вперше відобразили дані, отримані з цих експериментів, за допомогою систем координат, кожна вісь яких відповідала інтенсивності одного з основних кольорів. Однак від'ємні значення були можливими, оскільки жодні три основні кольори в комбінації ніколи не можуть репрезентувати всі кольори спектра, а колориметри дозволяли здійснювати відповідну компенсацію. Згадані вище експерименти, які проводилися на кількох спостерігачах, надали комбіновані дані для визначення одного стандартного спостерігача. Крім того, конкретний колірний простір XYZ також залежить від джерела світла або точки білого. Може бути корисно використовувати різні джерела світла для різних умов, оскільки кольори будуть сприйматися по-різному при яскравому сонячному світлі, ніж при світлі лампи. Колірний простір CIE XYZ [1-5] - це перетворення тривимірної системи координат на основі стандартизованих основних кольорів таким чином, щоб гарантувати наступні властивості:

- Для кожного кольору  $(X, Y, Z)^T$ , видимого людиною,  $X$ ,  $Y$  та  $Z$  є додатними.
- $Y$  збігається зі сприйнятою яскравістю кольору.

На основі цього простору XYZ можна отримати діаграму кольоровості  $x, y$ , як згадувалося вище, за допомогою визначення CIE координат  $x, y, z$  для точки  $P = (x', y', z')_{XYZ} = (x', y', z')^T$  у просторі XYZ.

$$\frac{x}{x'} = \frac{y}{y'} = \frac{z}{z'} = \frac{1}{x' + y' + z'} \quad (1.1)$$

Таким чином,

$$x = \frac{x'}{x' + y' + z'}, y = \frac{y'}{x' + y' + z'}, z = \frac{z'}{x' + y' + z'} \quad (1.2)$$

Будь-яку таку точку  $P$  можна описати як пряму  $l = 0 + \lambda \cdot P$ , що проходить через початок координат і  $P$ . З рівняння 1.2 випливає, що  $x + y + z = 1$  і що кожна точка  $P$  масштабується на  $\lambda = 1/(x'+y'+z')$ . Отже, рівняння 2.1 описує проекцію через початок координат на площину  $\tilde{x} + \tilde{y} + \tilde{z} = 1$  для точок  $XYZ (\tilde{x}; \tilde{y}; \tilde{z})^T$ . Тоді система координат CIE  $x, y$  може бути отримана просто ігноруванням компоненти  $z$ .

Як і в експериментах Райта і Гільда, кольори RGB [7-8] утворюються шляхом додавання різних інтенсивностей червоного, зеленого і синього світла. Колірну модель RGB можна уявити як куб у тривимірній декартовій системі координат з однією віссю для червоного, зеленого та синього каналів. Отже, колір можна визначити як вектор  $(r, g, b)^T$  або кортеж  $(r, g, b)_{\text{RGB}}$ , де  $r, g$  і  $b$  - це інтенсивність кольору для кожного каналу. У комп'ютерній графіці також часто використовують інтегральні значення від 0 до 255 (від 0 до FF у шістнадцятковій системі числення). Оскільки визначення моделі RGB не визначає точних відтінків трьох основних кольорів - червоного, зеленого і синього, можливі різні колірні простори RGB з різними гаммами, наприклад, колірні простори CIE RGB, стандартний RGB (sRGB) та Adobe. Серед інших параметрів, sRGB визначає розташування трьох основних кольорів - червоного, зеленого та синього - у просторі CIE  $x, y$  (Табл. 1.1), а також передавальну функцію для гамма-корекції.

Таблиця 1.1 – Специфікація основних кольорів sRGB в межах простору кольорів CIE  $x, y$

	R	G	B
$x$	0.64	0.30	0.15
$y$	0.33	0.60	0.06

Передавальна функція для кожного компонента  $c_{\text{lin}} \in \{r, g, b\}$  кольору RGB  $(r, g, b)^T$  з лінійними інтенсивностями показана в наступному рівнянні.

$$c = \begin{cases} 1.055c_{lin}^{1/2.4} & \text{якщо } c_{lin} > 0.0031308 \\ 12.92c_{lin} & \text{якщо } c_{lin} \leq 0.0031308 \end{cases} \quad (1.3)$$

Для лінеаризації гамма-корегованого RGB-кольору (1.3) можна обернути:

$$c_{lin} = \begin{cases} ((c + 0.055)/1.055)^{2.4} & \text{якщо } c_{lin} > 0.0031308 \cdot 12.92 \\ c/12.92 & \text{якщо } c \leq 0.0031308 \cdot 12.92 \end{cases} \quad (1.4)$$

Лінійні кольори RGB  $(r_{lin}, g_{lin}, b_{lin})^T$  можна перетворити з координат CIE XYZ  $(x', y', z')^T$ , враховуючи стандартний освітлювач, за допомогою перетворення (1.4). Маючи лінеаризований колір RGB, координати XYZ можна обчислити за допомогою оберненої матриці.

$$\begin{aligned} (r_{lin}, g_{lin}, b_{lin}) &= (x', y', z')M = (x', y', z') \begin{bmatrix} 3.2401 & -0.9689 & 0.0557 \\ -1.5372 & 1.8758 & -0.2040 \\ -0.4986 & 0.0415 & 1.0570 \end{bmatrix} \\ (x', y', z') &= (r_{lin}, g_{lin}, b_{lin})M^{-1} = (r_{lin}, g_{lin}, b_{lin}) \begin{bmatrix} 0.4124 & 0.2126 & 0.0193 \\ 0.3576 & 0.7152 & 0.1192 \\ 0.1805 & 0.0722 & 0.9505 \end{bmatrix} \end{aligned} \quad (1.5)$$

На відміну від RGB, колірна модель CMY [7-10] використовує субтрактивне змішування кольорів з основними блакитним, пурпуровим і жовтим, а не адитивне змішування кольорів. Субтрактивне змішування може бути знайоме більшості з малювання аквареллю. Крім того, пігменти блакитного, пурпурного та жовтого кольорів зазвичай використовуються у принтерах. В принципі, перетворення кольорів з RGB в CMY і назад є простим. У моделі RGB блакитний колір можна отримати із суміші лише зеленого та синього, а це означає, що при використанні в якості фільтра або пігменту блакитний колір поглинає червоне світло. Аналогічно, пурпуровий поглинає зелене, а жовтий - синє світло. Використовуючи цю інформацію, перетворення між кольором RGB  $(r, g, b)^T$  і відповідним кольором CMY  $(c, m, y)_{CMY} = (c, m, y)^T$  працює, починаючи з білого і віднімаючи колір у протилежному представленні, як показано в

наступних рівняннях.

$$\begin{aligned}(c, m, y) &= (1, 1, 1) - (r, g, b) \\ (r, g, b) &= (1, 1, 1) - (c, m, y)\end{aligned}\tag{1.6}$$

Однак на практиці цей метод має певні проблеми. Згідно з (1.6), чорний колір - це суміш 100% блакитного, пурпурного та жовтого. Однак, як і у випадку з аквареллю, очищення пензля у склянці з водою після малювання різними кольорами легше перетворює воду на коричневий відтінок, ніж на справжній чорний колір. Поширеним рішенням цієї проблеми є введення четвертого, чорного каналу  $K$ . Іншою проблемою є той факт, що СМУК та кольорові моделі, такі як RGB, призначені для різних типів пристроїв, які часто працюють з різними гаммами. Перетворення в СМУК є нетривіальним і залежить від пристрою. В рамках цієї тези буде розглянуто лише прямий метод перетворення кольорів СМУ  $(c, m, y)^T$  у кольори СМУК  $(c', m', y', k)_{\text{СМУК}} = (c', m', y', k)^T$ . Тобто

$$(c', m', y', k') = (c - k, m - k, y - k, k)\tag{1.7}$$

де  $k = \min \{c, m, y\}$ . Завдяки введенню каналу  $K$ , в інших трьох каналах потрібна менша інтенсивність. Це враховується відніманням  $k$  від вихідного СМУ-кольору (1.7)

Хоча більшість людських очей розпізнають кольори подібно до того, як працює модель RGB, ми зазвичай не думаємо і не говоримо про кольори як про суміш цих трьох компонентів. Однак ми можемо говорити про кольори як більш чи менш насичені, ніж інші, з різними відтінками чи тонами, або як яскравіші за інші. З цієї причини програмні додатки Blender<sup>1</sup>, GNU Image Manipulation Program (GIMP), Inkscape і Krita включають в себе інструменти для підбору кольорів з урахуванням таких побажань.

Однією з таких моделей сприйняття кольору є HSL (також відома як HLS), що є аббревіатурою відтінку, насиченості та світлоти. Вперше вона була введена

Джоблавам і Грінбергом як "відтінок/кольоровість/інтенсивність" [10]. Вони описують колірний простір як біконічну площину, на якому вертикальна вісь представляє всі відтінки сірого між 0 (чорний) і 1 (білий). Тоді всі повністю насичені кольори лежать на зовнішньому колі спільної основи обох конусів при  $L = 0,5$ , що дозволяє визначити відтінок як кут. Третій параметр, насиченість, відповідає радіусу кола навколо вертикальної осі в точці поточної освітленості.

Для заданого кольору RGB  $(r, g, b)^T$ , світлість  $l$  та насиченість  $s$  HSL кольору HSL  $(h, sHSL, l)_{HSL} = (h, sHSL, l)^T$  визначаються наступним чином:

$$s = \begin{cases} \frac{\max\{r, g, b\} - \min\{r, g, b\}}{\max\{r, g, b\} + \min\{r, g, b\}}, & \text{якщо } \max\{r, g, b\} + \min\{r, g, b\} \leq 1; \\ \frac{\max\{r, g, b\} - \min\{r, g, b\}}{1 - \max\{r, g, b\} - \min\{r, g, b\}}, & \text{якщо } \max\{r, g, b\} + \min\{r, g, b\} > 1 \\ 0, & \text{якщо } \frac{\max\{r, g, b\} + \min\{r, g, b\}}{2} = 0 \text{ (або } 1); \end{cases} \quad (1.8)$$

Відтінок  $h \in [0, 1]$  обчислюється від заданого RGB-кольору з проміжними кроками наступним чином:

$$(r', g', b') = \left( \frac{\max\{r, g, b\} - r}{\max\{r, g, b\} - \min\{r, g, b\}}, \frac{\max\{r, g, b\} - g}{\max\{r, g, b\} - \min\{r, g, b\}}, \frac{\max\{r, g, b\} - b}{\max\{r, g, b\} - \min\{r, g, b\}} \right) \quad (1.9)$$

$$h' = \begin{cases} b' - g', & \text{якщо } \max\{r, g, b\} = r; \\ r' - b' + 2, & \text{якщо } \max\{r, g, b\} = g; \\ g' - r' + 4, & \text{якщо } \max\{r, g, b\} = b; \end{cases} \quad (1.10)$$

$$h = \frac{h'}{6} \bmod 1 \quad (1.11)$$

Колірна модель HSV має багато спільних властивостей з HSL [4-7]. Літера V позначає значення і іноді замінюється літерою B для яскравості, що вказує на те, що третій компонент у цій колірній моделі визначається по-іншому. Замість того, щоб перетворювати куб RGB у біконічну форму, HSV починає з одного перевернутого конуса. Відтінок - це знову кут, який можна обчислити так само, як і для HSL за допомогою рівнянь 1.9 – 1.11. Оскільки значення або яскравість для кольору RGB  $c_{RGB} = (r, g, b)^T$  тепер визначається просто як  $v = \max\{r, g, b\}$ ,

насиченість  $s_{HSV}$  визначається наступним чином для  $c_{RGB} \neq (0, 0, 0)^T$ :

$$s_{HSV} = \frac{\max\{r, g, b\} - \min\{r, g, b\}}{\max\{r, g, b\}} \quad (1.12)$$

Одне з рішень проблем з HSL і HSV, щоб розглянути систему кольорів, яка, принаймні частково, надихнула на створення колірної моделі HSL. Як і колірний простір XYZ, але з іншими інструментами, система Манселла базується на експериментальних даних. Система описує так зване "дерево кольорів". Подібно до HSL і HSV, є вертикальна вісь, що представляє яскравість, у цьому випадку також звану значенням. Тут реакція людського ока на інтенсивність світла моделюється логарифмічно. Відтінок знову визначається як кут з додатковими кольорами, розташованими навпроти один одного, а насиченість, в даному випадку названа хромою, визначається як радіус. Важливою відмінністю між HSL та HSV є наступна: Оскільки кольоровість Манселла базується виключно на людському сприйнятті насиченості кольору, колірний простір не може бути представлений в ідеально круглій формі. Скоріше, при погляді зверху вниз видно чіткі виступи, наприклад, навколо кута відтінку для червоного і для суміші синього і фіолетового, що вказує на те, що чисті кольори цих відтінків сприймаються як більш насичені, ніж інші. Щоб зменшити нерівномірність у розташуванні цих колірних зразків, зразки було проаналізовано експериментально і визначено їхнє положення в колірному просторі CIE  $xy$ . Ці дані можна використати для переведення кольорів із системи Манселла у  $xy$ .

Іншим популярним рішенням для проблем з HSL і HSV є колірний простір CIELab [10-11]. Однією з цілей розробки цього колірного простору була формула для кількісної оцінки сприйнятої різниці між двома кольорами. Простір Lab визначається як перетворення колірного простору CIE XYZ за допомогою наступних рівнянь для кольорів Lab  $(l, a, b)^T$  і кольорів XYZ  $(x', y', z')^T$ :

$$\begin{aligned}
 l &= 117f\left(\frac{y'}{Y_0}\right) - 16; \\
 a &= 500\left(f\left(\frac{x'}{X_0}\right) - f\left(\frac{y'}{Y_0}\right)\right) \\
 b &= 200\left(f\left(\frac{y'}{Y_0}\right) - f\left(\frac{z'}{Z_0}\right)\right)
 \end{aligned}
 \tag{1.13}$$

де  $(X_0, Y_0, Z_0)$  - задане джерело світла або біла точка, зазвичай стандартне джерело світла, що приблизно відповідає денному світлу з колірною температурою  $6500^\circ\text{K}$  і має координати хуз  $(0.3127, 0.3291, 0.3582)^T$ . Нормалізація до  $Y = 1$  дає наближені координати XYZ  $(0.9502, 1.0, 1.0884)^T$ . Функція  $f$  спочатку була визначена як  $f(c) = c^{1/3}$ , якщо  $c > c_t = 0.01$ . В іншому випадку передбачалося використовувати так зване рівняння ANLAB, щоб уникнути від'ємних значень  $L$  для темних кольорів, близьких до  $Y = 0$ . Для стандарту ISO 31655 це було дещо змінено на таку функцію  $f_1$  з  $c_{t1} = 0,008856$ .

Модель YCbCr [11-13] представляє інтерес завдяки її використанню в телебаченні та стисненню зображень у форматі JPEG.  $Y$  позначає яскравість,  $C_b$  - зважену різницю між яскравістю та синім кольором, і, аналогічно,  $C_r$  - зважену різницю між яскравістю та червоним кольором. Загальне перетворення з кольору RGB  $(r, g, b)^T$  в YCbCr, представленого як  $(y, c_b, c_r)^T$ , визначається наступним чином для ваг  $w_R, w_G$  і  $w_B = 1 - w_R - w_G$ :

$$\begin{aligned}
 y &= w_R r + w_G g + w_B b; \\
 c_b &= \frac{b - y}{2(1 - w_B)}; \\
 c_r &= \frac{r - y}{2(1 - w_R)}
 \end{aligned}
 \tag{1.14}$$

Рекомендовано використовувати такі ваги  $w_R, w_G$  і  $w_B$ :

$w_R=0.2990$   $w_G=0.5870$   $w_B=0.1140$  для jpeg – формату;

$w_R= 0.2126$   $w_G=0. 7152$   $w_B= 0.0722$  для HD – формату;

$w_R= 0.2627$   $w_G= 0.6780$   $w_B= 0.0593$  для ultraHD – формату.

### **1.3 Постановка задачі**

Метою роботи є розробка інформаційного і програмного забезпечення додатку оцінювання знань здобувачів вищої освіти дистанційної форми навчання з урахуванням особливостей наведених у розділі 1.2 матеріалів дистанційного курсу «Обробка зображень та мультимедіа».

Для досягнення поставленої мети необхідно вирішити наступні задачі:

- 1) Розробити інформаційну модель додатку
- 2) Розробити дизайн інтерактивних компонентів додатку
- 3) Розробити набір варіантів завдань для оцінки знань та алгоритмів для їх виконання здобувачами і автоматичної перевірки
- 4) Обрати засоби для програмної реалізації додатку
- 5) Програмно реалізувати додаток та протестувати його працездатність.



## 2 ВИБІР МЕТОДУ РІШЕННЯ

### 2.1 Інформаційна модель

Як показано на рисунку 2.1, загалом, мета вправи, яку потрібно розробити, є потрібною. По-перше, вона надає здобувачу можливість дізнатися про колір і кольорове бачення загалом. По-друге, здобувач може дізнатися про конкретні колірні моделі та колірні простори, і, нарешті, є функціональні можливості для того, щоб здобувач міг практикувати те, що він вже знає, або те, що він дізнався в перших двох випадках. У контексті цієї кваліфікаційної роботи особливо важливими є другий і третій варіанти використання. Другий варіант використання про колірні моделі і колірні простори відображає зміст розділу 1.2 і включає інтерактивні візуалізації окремих колірних просторів. Третій варіант використання може бути використаний здобувачами для підготовки до іспиту або будь-якою особою для роздумів про те, що вони вже розуміють і де може знадобитися подальше вивчення.

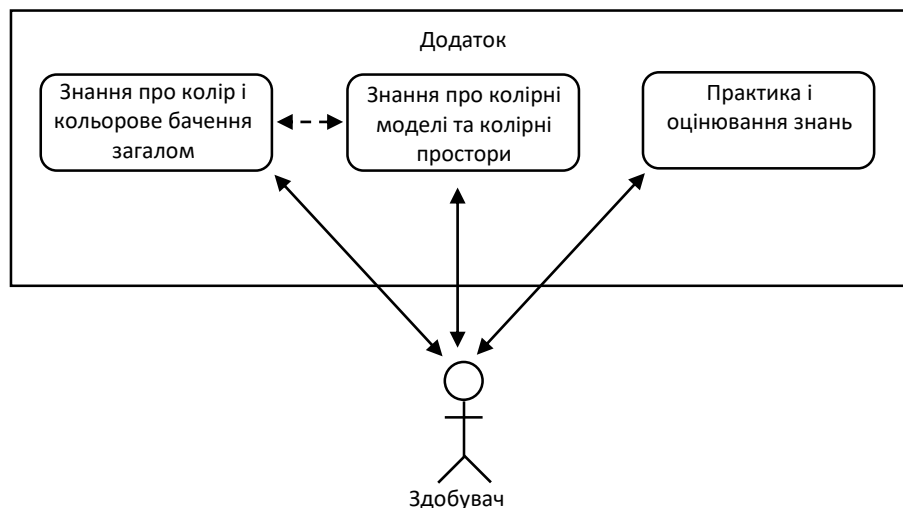


Рисунок 2.1 – Діаграма варіантів використання додатку

Кожна колірна модель або колірний простір представлені на окремій

сторінці додатку. На рисунку 2.2 показано, як ці сторінки пов'язані з інтерактивними візуалізаціями. Окрім звичайного HTML-тексту та посилань, сторінка може містити будь-яку кількість рисунків. Візуалізації розглядаються як особливий вид рисунків, що візуалізують принаймні один колірний простір. Випадок, коли одна візуалізація показує більше одного колірного простору, може бути корисним для порівняння різних колірних просторів або для візуалізації перетворень.

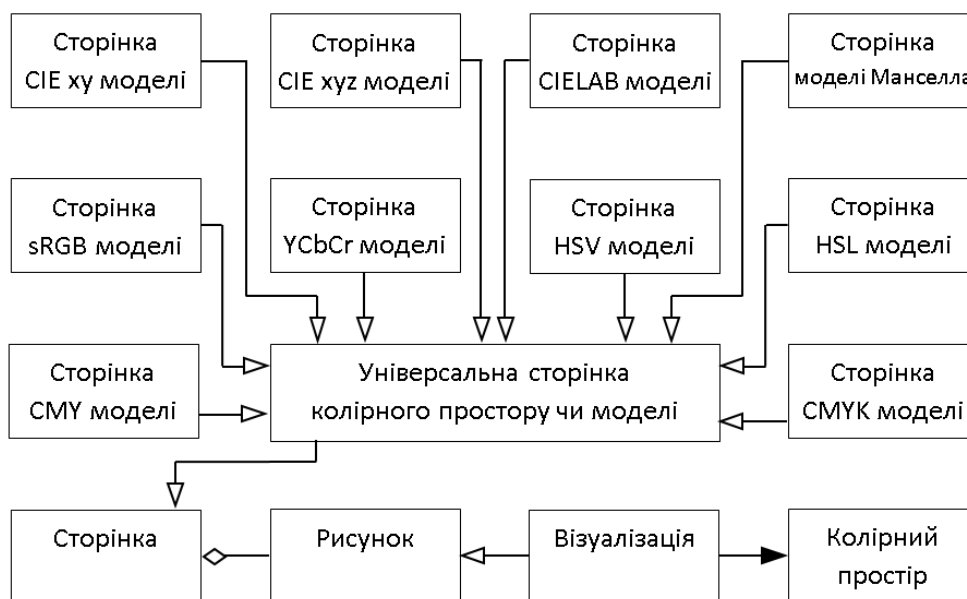


Рисунок 2.2 – Діаграма класів, що показує склад сторінок для навчання про колірні моделі та колірні простори

## 2.2 Розробка компонентів для візуалізації і інтерактивної взаємодії з колірними моделями

Взаємодія з візуалізацією залежить від того, які колірні простори відображаються в даний момент. На рисунку 2.3 показано, як можуть виглядати елементи керування такою візуалізацією, коли куб RGB і простір HSL візуалізуються поруч. Під кожним тривимірним представленням є повзунки та поля введення для налаштування відповідних параметрів кожної системи. Зміна будь-якого з повзунків призводить до наступних змін: Вибраний колір у

нижньому лівому кутку оновлюється, показується розташування вибраного кольору у відповідному колірному просторі, а входи інших колірних моделей налаштовуються так, щоб показувати параметри для того самого вибраного кольору. Дотепер у цій роботі більшість колірних просторів було намальовано у вигляді каркасної моделі. У правому нижньому куті рисунка 2.3 показано кнопку для виклику набору додаткових елементів керування. Вони можуть включати налаштування для перемикання між каркасним представленням і непрозорим або напівпрозорим суцільним. Додатково можуть бути реалізовані опції перемикання між значеннями RGB з плаваючою комою та значеннями від 0 до 255, а також опція представлення відтінку в інтервалі  $[0,1]$ , в радіанах, градусах або цілими числами від 0 до 255. Оскільки ці параметри є необов'язковими і не мають безпосереднього відношення до розуміння колірних моделей, за замовчуванням вони приховані.

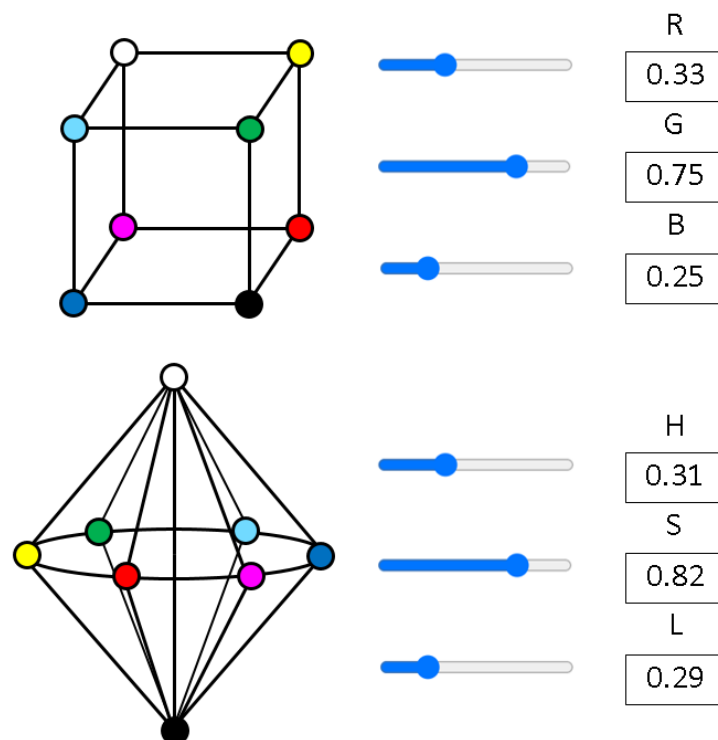


Рисунок 2.3 – Ескіз UI інтерактивної візуалізації та порівняння RGB та HSL

Далі будуть розглянемо візуалізації для кожної з представлених у розділі 1.2 колірних систем.

CIE XYZ, CIE xy: Діаграма кольоровості в просторі CIE xy, як показано на рисунку 1.2, є важливою, оскільки вона допомагає проілюструвати межі практичних колірних просторів, а також відмінності між ними. Перехід від тривимірного простору XYZ до двовимірного простору x, y можна продемонструвати, показавши перетин видимої гами з площиною  $\tilde{x} + \tilde{y} + \tilde{z} = 1$ . Якщо видиму гаму зробити прозорою, то отримані проекції можна нанести безпосередньо на площини проекцій. RGB та sRGB:

Найважливішим для моделі RGB є куб RGB, як показано на рисунку 2.3. Для демонстрації неоднорідності сприйняття простору sRGB корисним є візуальне порівняння з іншим колірним простором. За задумом, система CIELAB або модель Манселла була б ідеальною. Однак, через відносну простоту моделі RGB, можна очікувати, що вона буде однією з перших, яку вивчатиме здобувач. Цю проблему можна вирішити, надавши здобувакам список запропонованих колірних просторів для порівняння з посиланнями на відповідні сторінки. У візуалізації, подібній до рисунка 2.3, здобувач може вибрати один із запропонованих колірних просторів, з яким він вже знайомий, для порівняння з кубом RGB.

CMY і CMYK: Оскільки куб CMY - це просто перевернутий куб RGB, візуалізація CMY дуже схожа на візуалізацію RGB.

HSL і HSV: Подвійний конус на рисунку 2.3 для HSL або одинарний конус корисний для розуміння визначення колірних моделей HSL і HSV. Формули перетворення описують циліндр, який часто показують з вигляду зверху вниз, наприклад, у програмах з відкритим вихідним кодом Blender або Paint. Іншим можливим представленням є куб HSL або HSV, який отримують, розглядаючи три компоненти відповідної колірної моделі як осі для декартової системи координат. Paint, наприклад, містить піпетку кольорів, яка дозволяє користувачеві переглядати куб HSV з трьох різних сторін. Тому, щоб допомогти здобувачам зрозуміти зв'язок між оригінальною колірною моделлю і тим, що вони звикли бачити в палітрах кольорів, може бути корисним візуальне

порівняння між різними представленнями HSL і HSV. Крім того, для ілюстрації перетворення кольорів корисно відтворити порівняти HSV і RGB, а порівняння HSL і HSV можна використати, щоб підкреслити схожість і відмінності між цими двома колірними моделями.

Модель Манселла: Враховуючи дані моделі Манселла, можна зобразити дерево кольорів Манселла у тривимірному вигляді. Для демонстрації різниці між HSL і HSV, на додаток до дерева кольорів можна відобразити одну з цих колірних систем.

CIELAB: Колірний простір Lab є перетворенням простору XYZ, і він також містить усі кольори, видимі для людського ока. Тому одна візуалізація може показувати видиму гаму або, наприклад, гаму sRGB у тривимірному просторі. Візуальне порівняння з простором XYZ може також продемонструвати функцію Lab, яка полягає в тому, що кольори, які сприймаються однаково віддаленими один від одного, залишаються однаково віддаленими в колірному просторі.

YCbCr: Як пояснювалося у розділі 1.2, колірний простір YCbCr може бути отриманий шляхом застосування матриці перетворення до будь-якого кольору RGB. Тому, залежно від параметрів  $w_R$ ,  $w_G$  і  $w_B$ , колірний простір YCbCr можна візуалізувати як відповідно перетворений куб RGB всередині декартової системи координат з осями Y,  $C_b$  і  $C_r$ .

### 2.3 Розробка завдань для оцінки знань

Розглянемо основні завдання для оцінки знань здобувачів:

- Завдання на співставлення кольорів
- Завдання на вибір кольору
- Завдання на перехід від однієї колірної моделі до іншої
- Комплексні завдання

Як приклад завдання на підбір або співставлення кольору, випадковий, але постійний колір відображається в прямокутнику вікна завдання, а інший колір, який можна змінювати за допомогою повзунків, показано поруч. Завдання полягає в тому, щоб обидва прямокутники відображали однаковий колір. Якщо

вибраний колір близький до оригіналу, то напис вказує на те, що він майже правильний, а якщо два кольори збігаються, то про це також сповіщає напис. Таке завдання може виконуватися для кожної з розглянутих кольорових систем. У цьому контексті вона дає здобувачам змогу відчутти брак інтуїтивності у використанні кожної колірної моделі для підбору кольорів. Крім того, ця вправа спонукає застосувати своє розуміння параметрів моделі. На противагу цьому, якби здобувачу не було поставлено завдання підібрати два кольори, він, можливо, досліджував би переважно екстремальні значення для кожного параметра  $i$ , можливо, не до кінця зрозумів би вплив кожного параметра на кінцевий колір. Наприклад, якщо ви протестували всі вісім кольорів RGB  $(0, 0, 0)^T$ ,  $(0, 0, 1)^T$ , . . . ,  $(1, 1, 1)^T$ , ще не зрозуміло, де знайти певний відтінок коричневого. Надання студентам можливості порівняти варіації цієї вправи для різних колірних систем може допомогти їм зрозуміти значення кожного параметра, а також переваги та недоліки кожної системи з точки зору простоти використання. Однією з проблем початкової вправи є те, що час від часу два кольори виглядатимуть так, ніби вони однакові, але жодних ознак успішного збігу не буде вказано. Щоб зменшити розчарування з боку користувача, можна ввести додаткові індикатори для кількісної оцінки різниці між двома кольорами. Наприклад, відображення евклідової відстані між двома кольорами в кубі RGB може бути корисним для того, щоб показати, що у наведеному вище сценарії ці два кольори дійсно не однакові. З іншого боку, треба бути обережним, щоб ці показники не спокусили здобувачів до махінацій у завданні. Враховуючи евклідову відстань, наприклад, кожен компонент RGB можна було б по черзі змінювати до точки, де відстань буде мінімальною, не вимагаючи особливих знань про колірний простір. Цього можна до певної міри уникнути, якщо показувати відстань лише після натискання кнопки про завершення підбору кольору.

Завдання на вибір кольору полягає в тому, щоб знайти випадковий колір, заданий у числовому вигляді, серед списку інших випадкових кольорів. За

умови, що відмінності між кольорами для вибору не надто великі, а список кольорів не надто довгий, здобувач повинен швидко знайти потрібний колір. Як і вправа на співставлення кольорів, вправа на вибір кольору є гнучкою щодо кольорових систем, з якими вона може бути використана. Можна практикувати роботу з однією конкретною системою кольорів, або ж у кожному раунді випадковим чином вибирати одну систему кольорів із задалегідь обраного набору систем для числового представлення. Останній варіант є найбільш корисним під час підготовки до іспиту, оскільки він перешкоджає здобувачам, можливо, несвідомо, уникати тих систем кольорів, з якими вони не так добре знайомі. З метою мотивації може бути корисним дозволити здобувачеві виконати вправу протягом  $M$  раундів після чого він отримає оцінку. В ідеалі, ця оцінка дасть здобувачеві уявлення про те, скільки йому ще потрібно вчитися, щоб краще розуміти відповідні кольорові системи. З цією метою оцінка може включати кількість кольорів, які здобувач вибрав правильно з першої спроби. Порівняння свого результату із середнім показником користувача може слугувати індикатором для самооцінки, а також мотивувати до вдосконалення.

Для здобувачів завдання на перетворення кольорів при переході від однієї колірної моделі до іншої є особливо цікавою, оскільки її найлегше інтегрувати в звичайні не дистанційні іспити, коли завдання друкуються в чорно-білому варіанті на папері. В роботі розглядають три рівні складності такого завдання.

Перший рівень складності - Конвертувати потрібно лише ключові кольори. Наприклад, у HSV і HSL кольори, які є повністю насиченими або зовсім не насиченими, можна відносно легко конвертувати між двома системами, а також з і в RGB. Однак у цього варіанту є обмеження. У колірних просторах CIELAB, CIE XYZ і Munsell не існує ключових кольорів, які легко запам'ятовуються. Крім того, різні ваги, рекомендовані для YCbCr, ускладнюють перетворення в інші колірні простори. Тому цей варіант вправи з перетворення кольорів обмежується RGB, HSL, HSV і CMY.

Другий рівень складності: Як і у вправі на вибір кольору, здобувачі

отримують колір, представлений числовим значенням в одній колірній системі. Тепер вони мають вибрати відповідний колір не зі списку прямокутників певного кольору, а зі списку кольорів, представлених в іншій системі числення. Існують такі ж обмеження, як і для варіанту першого рівня складності, оскільки CIELAB, CIE XYZ, Munsell і YCbCr не можуть бути легко конвертовані без сторонньої допомоги. Якщо ви використовуєте HSL або HSV, завдання можна дещо спростити, вибравши для відтінку число, кратне 1/6 або 1/12. Загалом, слід уникати занадто великої кількості десяткових знаків, щоб полегшити обчислення. Здобувачі повинні зосередитися насамперед на розумінні системи кольорів, і їх слід заохочувати до виконання заданої кількості раундів вправи принаймні один раз.

Третій рівень складності: Користувач має вказати значення перетвореного кольору вручну за допомогою повзунків або клавіатури. При цьому вибирається поріг для максимальної відстані між правильним перетворенням і значеннями, вибраними користувачем, при яких відповідь все ще вважається правильною. Якщо вибір користувача знаходиться в межах цього порогу, але не дорівнює точному результату, можна показати точне перетворення і, за бажанням, візуально порівняти його з вибраним здобувачем значенням.

Для всіх трьох варіантів можна запровадити систему спроб. Оскільки завдання може бути складним, є сенс показати візуалізацію обох колірних просторів, що беруть участь у перетворенні. З іншого боку, оскільки на іспитах, як правило, така візуальна допомога не надається, користувач має можливість переключитися в режим іспиту, в якому візуалізація для цієї вправи вимкнена.

Мета змішаних завдань - пояснити призначення та вплив параметрів колірних систем у контексті, більш наближеному до реальних застосувань, ніж, наприклад, ручне перетворення кольорів. Для вирішення таких завдань здобувачеві надається  $M$  спроб, як в завданні на вибір кольору. Перед початком вправи користувач спочатку обирає декілька кольорових систем. З непорожнього набору обраних колірних систем у кожному раунді випадковим чином



обирається одна. Багато з представлених тут систем можуть мати кілька різних завдань, пов'язаних з ними. Розглянемо можливі завдання для кожної колірної моделі або колірного простору.

CIE XYZ, CIE xy: Зменшення або збільшення яскравості в XYZ. Оскільки осі в цих колірних системах представляють уявні основні кольори, прості завдання для вправ не можуть бути легко побудовані.

RGB і sRGB: Зменшення або збільшення яскравості. Наприклад, щоб збільшити яскравість, достатньо збільшити значення червоного, зеленого або синього кольору. Завдання, подібні до завдань для HSL і HSV, також можливі, хоча вони зроблять цю вправу дуже схожою на вправу з перетворення.

CMY і CMYK: Зменшити або збільшити яскравість. Це завдання подібне до завдання RGB. HSL і HSV: Вибір кольору HSL або HSV:

- Десатурація (не змінюючи відтінок або світлість/значення).
- Зробити колір темнішим/яскравішим (без зміни відтінку або насиченості).
- Надати кольору відтінок між, наприклад, зеленим і жовтим (без зміни насиченості або світлоти/значення).

Munsell: Завдання для Munsell можуть включати завдання для HSL і HSV з адаптованою термінологією.

CIELAB:

- Для двох Lab кольорів  $x$  та  $y$  знайдіть інший колір, який віддалений від  $y$  на стільки ж, на скільки  $y$  віддалений від  $x$ .
- За заданим Lab кольором знайти інший колір з меншою/більшою кольоровістю. Знайти точно такий самий колір з більшою чи меншою кольоровістю у просторі CIELAB, на жаль, нетривіально. Особливо відтінки синього мають тенденцію ставати фіолетовими зі збільшенням відстані від білої точки і постійною яскравістю.

YCbCr: Зменшення або збільшення яскравості. На додаток до цих завдань, специфічних для певної системи, можна запропонувати більше завдань для певних комбінацій колірних систем, за умови, що ця комбінація відображена у

виборі користувача на початку завдання. Наприклад, знову попросити користувача змінити відтінок заданого кольору, не змінюючи видимої яскравості. Замість того, щоб давати це завдання в контексті колірному простору Munsell або CIELAB, здобувач повинен самостійно знайти відповідну колірну систему.

## 3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

### 3.1 Засоби для програмної реалізації додатку

Програмна реалізація здійснювалась на мовах програмування Javascript [13-19]. При цьому використовувалися такі бібліотеки та фреймворки:

**Three.js:** Three.js додає шар абстракції поверх WebGL. Він надає функціонал сценічного що дозволяє організовувати 3D-об'єкти ієрархічно, таким чином дещо спрощуючи позиціонування та анімацію елементів. Бібліотека також постачається з низкою попередньо визначених примітивів та матеріалів.

**Jquery:** Jquery - це корисна бібліотека, яка дозволяє легко маніпулювати HTML-вмістом сторінки за допомогою селекторів, подібних до CSS. Така функціональність використовувалась, наприклад, для динамічної вставки елементів керування повзунком залежно від поточної візуалізації.

**MathJax:** Оскільки перетворення між колірними просторами вимагає деяких математичних формул, має сенс зробити їх максимально читабельними. MathJax може інтерпретувати LATEX-код, вбудований в HTML.

**Less CSS:** Less CSS, розширення CSS, використовується для того, щоб зробити код CSS проекту більш зручним для підтримки. Це досягається за допомогою змінних, модуляризації та вкладених селекторів, серед іншого. Оскільки браузері підтримують лише звичайний CSS, код, написаний за допомогою Less, перед публікацією потрібно скомпілювати.

**Normalize.css:** Різні браузері можуть мати різні налаштування за замовчуванням для шрифту або інтервалів між абзацами, наприклад. Normalize.css має на меті зробити так, щоб веб-сайти виглядали однаково незалежно від використовуваного браузера.

### 3.2 Опис програмної реалізації

Як згадувалося в розділі 1.2, зміни в одній колірній системі мають бути синхронізовані з рештою колірних систем для порівняння. Кожна візуалізація містить посилання на власну реалізацію абстрактного класу, який відповідає за

перетворення кольорів з  $i$  в RGB. Це буде критично важливо для синхронізації, оскільки кольори майже з усіх колірних просторів можна без особливих зусиль конвертувати у RGB та з нього. Кожна реалізація колірної системи зазвичай складається з трьох або чотирьох числових значень, які зберігаються в окремих об'єктах, які, у свою чергу, можуть бути підключені до повзункового елемента керування. Отже, щоразу, коли користувач вносить зміни до одного з повзунків, одна з властивостей колірної системи сповіщає про це колірну систему, яка знову ж таки сповіщає візуалізацію. Замість того, щоб малювати візуалізацію в нескінченному циклі достатньо лише оновити екран, коли відбувається така подія.

Найпростішою візуалізацією з точки зору реалізації є RGB-куб [20-21]. Суцільний та каркасний кубоїд вже передбачені в бібліотеці Three.js. Замість того, щоб малювати лише каркасну модель, як на рисунку 2.3, можна намалювати непрозоре кольорове тіло. Це має ту перевагу, що інтерполяція між кількома ключовими кольорами не залежить від уяви користувача. Натомість він може бачити представлення колірного простору настільки точно, наскільки це дозволяє його пристрій відображення. Одним із способів досягти такої точної візуалізації колірного простору RGB є використання конвеєра програмованого рендерингу. Ідея полягає у використанні мови програмування шейдерів GLSL для написання фрагментного шейдера, який визначає колір кожного пікселя на основі того, яку частину кольорового суцільного він повинен показувати. Якщо куб розміщено у просторі як одиничний куб між координатами  $(0, 0, 0)^T$  і  $(1, 1, 1)^T$ , фрагментному шейдеру потрібно просто інтерпретувати поточну координату як колір RGB.

Програмна реалізація блоку відображення та перевірки завдань дозволяє зберігати лише один загальний клас завдань різних типів залежно від поточної конфігурації. Всі завдання зберігаються у вигляді списку і викладаються по черзі, коли користувач натискає на кнопку з написом "Далі". Якщо відповідь була правильною або було досягнуто максимальної кількості спроб, цей клік

приведе до завершення відображення і перевірки поточного завдання.

Конфігурація завдань налаштовується в HTML-кодї відповідної веб-сторінки. Наприклад, у комплексному варіанті, що містить завдання на вибір кольору та завдання на відповідність кольорів. Для кожного елемента div на сторінці створюється новий екземпляр класу JavaScript. Цей клас очікує два атрибути даних HTML для зазначеного елемента div: Один з них вказує на загальну кількість завдань, після виконання яких вправа має завершитися, а інший - на можливі типи завдань. Останній представлений у вигляді масиву в JavaScript Object Notation (JSON).

### 3.3 Тестування

Для перевірки працездатності додатку кожний з типів завдань перевірявся окремо. На першому етапі перевірку проходили завдання на співпадіння кольорів. На рисунку 3.1 подано скріншоти основних етапів проходження такого типу завдання.

#### Завдання на співставлення кольорів

Колірні моделі rgb, cmy, cmyk, hsl, hsv

**Умови завдання:**

Кількість спроб:

Генерація випадкових кольорів:

генерує випадкові кольори, а не обирає зі списку стандартних

**Оп-ції**

З візуалізацією:

З підказками:

Параметри за замовчанням    Розпочати

а

**Завдання 4/10:** Налаштуйте параметри CMYK так, щоб колір праворуч відповідав кольору ліворуч.

Колір не співпадає    Тепліше    Майже співпадає    ОК

C:

M:

Y:

K:

**Не правильно!** Спроба 1/1. Два кольори недостатньо схожі.  
Правильний результат(0,07, 0,00, 0,26, 0,04)сmyk

Максимальна допустима різниця: 0,087.  
Поточна різниця в RGB: 0,370

Передати відповідь    Далі

в

**Завдання 4/10:** Налаштуйте параметри CMYK так, щоб колір праворуч відповідав кольору ліворуч.

Колір не співпадає    Тепліше    Майже співпадає    ОК

C:

M:

Y:

K:

Передати відповідь    Далі

б

**Завдання 4/10:** Налаштуйте параметри CMYK так, щоб колір праворуч відповідав кольору ліворуч.

Колір не співпадає    Тепліше    Майже співпадає    ОК

C:

M:

Y:

K:

**Правильно!** Точний результат (0,05, 0,00, 0,10, 0,15)сmyk  
Максимально допустима різниця в RGB: 0,087.  
Поточна різниця в RGB: 0,075

Передати відповідь    Далі

д

Рисунок 3.1 – Завдання на співпадіння кольорів

Аналіз результатів тестування показує, що завдання генеруються в заданій кількості з урахуванням заданих параметрів (рис. 3.1 а). При генеруванні завдань початкові коефіцієнти (рис. 3.1 б) не є правильною відповіддю на завдання і при спробі передати їх як відповідь система вказує на помилку (рис. 3.1 в). При введенні правильних коефіцієнтів система реагує коректно (рис. 3.1 г).

На наступному етапі перевірку проходили завдання на вибір кольору. На рисунку 3.2 подано скріншоти основних етапів проходження такого типу завдання.

### Завдання на вибір кольору

Колірна модель: rgb, cmy, cmyk, hsl, hsv



а



б



в

Рисунок 3.2 – Завдання на вибір кольору

Аналіз результатів тестування показує, що завдання генеруються і відображаються коректно (рис. 3.2 а). При введенні правильної відповіді (рис. 3.2 б) і не правильної відповіді система реагує відповідним чином (рис. 3.2 в).

На наступному етапі перевірку проходили завдання на перехід від однієї колірної системи до іншої. На рисунку 3.3 подано скріншоти основних етапів проходження такого типу завдання.

### Завдання на перехід від однієї колірної моделі до іншої

Колірні моделі: rgb, cmy, cmyk, hsl, hsv

*a*

*б*

*в*

*д*

Рисунок 3.3 – Завдання на перехід від однієї колірної системи до іншої

Аналіз результатів тестування показує, що завдання генеруються в заданій кількості з урахуванням заданих параметрів (рис. 3.3 а,б). При введенні правильної відповіді (рис. 3.3 г) і не правильної відповіді система реагує відповідним чином (рис. 3.2 в).

На наступному етапі перевірку проходили комплексні завдання(рис. 3.4).

### Комплексне завдання

Колірні моделі: rgb, cmy, cmyk, hsl, hsv

Рисунок 3.4 – Комплексне завдання

Аналіз результатів тестування показує, що завдання на співпадіння кольорів, на вибір кольору, на перехід від однієї колірної системи до іншої генеруються в заданій кількості з урахуванням заданих параметрів (рис. 3.3 а).

Таким чином, результати тестування доводять працездатність запропонованої системи оцінювання знань з дистанційного курсу «Обробка зображень та мультимедіа», які стосуються колірних моделей та просторів.



## ВИСНОВКИ

В ході кваліфікаційної роботи бакалавра було проведено розробку інформаційного і програмного забезпечення додатку оцінювання знань здобувачів вищої освіти дистанційної форми навчання з дистанційного курсу «Обробка зображень та мультимедіа». При цьому основну увагу було приділено матеріалам щодо колірних моделей та просторів і переходами між ними.

При цьому було вирішено наступні задачі:

1. Розроблено інформаційну модель додатку у вигляді діаграма варіантів його використання та діаграми класів, що показує склад сторінок про колірні моделі та колірні простори
2. Розроблено дизайн інтерактивних компонентів додатку
3. Розроблено набір варіантів завдань для оцінки знань та алгоритмів для їх виконання здобувачами і автоматичної перевірки
4. Обрано засоби для програмної реалізації додатку
5. Програмно реалізувано додаток та протестувано його працездатність.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Rogers Adam. Full Spectrum: How the Science of Color Made Us Modern .- Houghton Mifflin Harcourt, 2021. — 349 p.
2. Fine A. Color Theory: A Critical Introduction .- Bloomsbury Visual Arts, 2021. — 377 p.
3. Laplante P.A. (ed.) Encyclopedia of Image Processing .- Boca Raton: CRC Press, 2019. — 875 p.
4. “RGB” vs “CMYK” – The Comprehensive Guide To Book Color Formats. – <https://bookbolt.io/rgb-vs-cmyk-the-comprehensive-guide-to-book-color-formats/>
5. .Color Gamut Definition.– <https://edgecolours.com/color-gamut-vs-color-space/>.
6. Aaron Fine Color Theory: A Critical Introduction. – Bloomsbury Visual Arts, 2022. - 377 p.
7. Steven Bleicher Contemporary Color : Theory and Use. - Cengage Learning, 2012. – 238 p.
8. Paul L. Rosin, Yu-Kun Lai, Ling Shao, Yonghuai Liu RGB-D Image Analysis and Processing. – Springer, 2019. – 522 p.
9. Davies, Helen Top 8 Large Format Printing Tips To Achieve High-End Projects.– Front Signs, 2020
10. Menegus, Bryan The Difference Between RGB and CMYK, Explained. – Gizmodo, 2020.
11. Device-Dependent Color Spaces. – <https://learn.microsoft.com/en-us/windows/win32/wcs/device-dependent-color-spaces>
12. Theresa-Marie Rhyne Applying Color Theory to Digital Media and Visualization CHI EA '17: Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing SystemsMay 2017Pages 1264–1267<https://doi.org/10.1145/3027063.3076594>

13. Smith S. Full Stack Web Development Guide: Everything Node JS, Express, APIs, EJS, React JS, Database Fundamentals, SQL Databases. - Independently published, 2022. — 744 p.
14. Scott Adam D., MacDonald Matthew, Powers Shelley. JavaScript Cookbook: Programming the Web .– 3rd Edition. — O’Reilly Media, Inc., 2021. — 538 p
15. Patel Meet. Advanced JavaScript Visualized .– Independently published, 2021. — 282 p.
16. Rappl Florian. The Art of Micro Frontends: Build websites using compositional UIs that grow naturally as your application scales .– Packt Publishing, 2021. — 310 p.
17. Meiert Jens Oliver. CSS Optimization Basics .– Leanpub book, 2021. — 42 p.
18. Lanciaux A. Modern Front-end Architecture: Optimize Your Front-end Development with Components, Storybook, and Mise en Place Philosophy. – Apress, 2021. — 141 p.
19. Ackermann Philip. JavaScript: The Comprehensive Guide . – Rheinwerk Computing, 2022. — 982 p.
20. Baruah Rakesh. AR and VR Using the WebXR API: Learn to Create Immersive Content with WebGL, Three.js, and A-Frame .– Apress Media LLC., 2021. — 351 p
21. HTML5 Canvas Notes for Professionals .– Finest Library, 2020. — 481 p.

## ДОДАТОК

```

<!DOCTYPE html>
<html>
  <head> </head> == $0
  <body>
    <div id="MathJax_Message" style="display: none;"></div>
    <nav class="sidenav">
      <a href="javascript:void(0)" class="closebtn" onclick="close_nav()"></a>
      <ul>
        <li>
          <a href="/Riab/KRB/web/">Домашня сторінка</a>
        </li>
        <li>
          <a href="/Riab/KRB/web/about.html">About</a>
        </li>
        <li>
          <a href="/Riab/KRB/web/color-models.html">Колірні моделі</a>
          <ul>
            <li>
              ::marker
              <a href="/Riab/KRB/web/color-models/rgb.html">RGB</a>
            </li>
            <li>
              ::marker
              <a href="/Riab/KRB/web/color-models/cmy.html">CMY та CMYK</a>
            </li>
            <li>
              ::marker
              <a href="/Riab/KRB/web/color-models/hsl.html">HSL</a>
            </li>
            <li>
              ::marker
              <a href="/Riab/KRB/web/color-models/hsv.html">HSV</a>
            </li>
          </ul>
        </li>
        <li>
          <a href="/Riab/KRB/web/tasks.html">Завдання</a>
          <ul>
            <li>
              ::marker
              <a href="/Riab/KRB/web/tasks/matching.html">Завдання на співпадіння кольорів </a>
            </li>
            <li>
              ::marker
              <a href="/Riab/KRB/web/tasks/selection.html">Завдання на вибір кольору</a>
            </li>
            <li>
              ::marker
            </li>
          </ul>
        </li>
      </ul>
    </nav>
  </body>
</html>

```

```

        <a href="/cgvb/colormaster/web/exercises.html">Exercises</a>
    </li>
</ul>
</li>
</ul>
</nav>
▼ <nav class="topnav">
  ▼ <div id="menu-button" class="menu-button" onclick="open_nav()">
    <div class="bar1"></div>
    <div class="bar2"></div>
    <div class="bar3"></div>
  </div>
</nav>
▼ <div class="main-container">
  ▼ <main>
    <h1>Завдання на співпадіння кольорів</h1>
    ▼ <p>
      <b>Колірні моделі</b>
      " rgb, сму, смук, hsl, hsv "
    </p>
    <p></p>
    ▼ <div class="exercise" data-num-rounds="10" data-allow-random-units="true" data-rand
      {
        "name": "ColorMatching",
        "options": {
          "color_systems": [
            "rgb", "смy", "смyк", "hsl", "hsv"
          ],
          "show_visualization": true,
          "visualization_options": {},
          "max_attempts": 3,
          "allow_skip_after_first_attempt": true,
          "show_current_color": true,
          "show_target_color": true,
          "show_hints": true
        },
        "weight": 1
      }
    ]">
    ▼ <div class="figure-title">
      <b>Завдання 4/10:</b>
      " Налаштуйте параметри СМУК так, щоб колір праворуч відповідав кольору ліворуч.
    </div>
    ▼ <span class="exercise-hints-label">
      <span class="hint active">Колір не співпадає</span>
      <span class="hint">Тепліше</span>
      <span class="hint">Майже співпадає</span>
      <span class="hint">ОК</span>
    </span>

```

```

▼ <div class="color-matching-sliders visualization-controls">
  ▼ <table class="controls-table">
    ▼ <tr class="visualization-control slider">
      ▼ <td class="shrink">
        <label for="vis-ctrl-419662241423732-slider">C:</label>
      </td>
      ▼ <td class="expand">
        <input type="range" name="vis-ctrl-419662241423732-slider" id=
        "vis-ctrl-419662241423732-slider" min="0.00" max="1.00" step=
        "0.01" value="0.68">
      </td>
      ▼ <td class="shrink">
        <input type="number" " id="vis-ctrl-419662241423732-number"
        min="0.00" max="1.00" step="0.01" value="0.68">
      </td>
    </tr>
    ▼ <tr class="visualization-control slider">
      ▼ <td class="shrink">
        <label for="vis-ctrl-943663268005829-slider">M:</label>
      </td>
      ▼ <td class="expand">
        <input type="range" name="vis-ctrl-943663268005829-slider" id=
        "vis-ctrl-943663268005829-slider" min="0.00" max="1.00" step=
        "0.01" value="0.22">
      </td>
      ▼ <td class="shrink">
        <input type="number" " id="vis-ctrl-943663268005829-number"
        min="0.00" max="1.00" step="0.01" value="0.22">
      </td>
    </tr>
    ▼ <tr class="visualization-control slider">
      ▼ <td class="shrink">
        <label for="vis-ctrl-753659886341677-slider">Y:</label>
      </td>
      ▼ <td class="expand">
        <input type="range" name="vis-ctrl-753659886341677-slider" id=
        "vis-ctrl-753659886341677-slider" min="0.00" max="1.00" step=
        "0.01" value="0.67">
      </td>
      ▼ <td class="shrink">
        <input type="number" " id="vis-ctrl-753659886341677-number"
        min="0.00" max="1.00" step="0.01" value="0.67">
      </td>
    </tr>
  </table>
</div>
</main>
▼ <footer>
  " © 2023 Рябущенко Андрій "
</footer>
</div>
</body>
</html>

```