Оглавление

Введение

Дальтонизм, цветовая слепота, — наследственная, реже приобретённая, особенность зрения человека и приматов, выражающаяся в сниженной способности или полной неспособности видеть или различать все или некоторые цвета.

Дальтонизм неизлечимая болезнь, но сегодняшние технологии могут помочь людям, страдающим этой болезнью, видеть изображение и различать цвета.

Дальтонизмом страдают примерно 1 из 12 мужчин и 1 из 200 женщин. Большинство людей с дальтонизмом имеют нормальное зрение, но не различают какие-то конкретные цвета. Очень редки случаи полной цветовой слепоты.

У человека в центральной части сетчатки расположены светочувствительные рецепторы — нервные клетки, которые называются колбочками. Каждый из трёх видов колбочек имеет свой тип светочувствительного пигмента, характеризующийся определённым спектром поглощения. Первый тип пигмента, условно называемый «красным», имеет максимум чувствительности к спектру с максимумом 560 нм; другой, «зелёный» — с максимумом 530 нм; третий, «синий» — с максимумом 430 нм.

Виды дальтонизма:

- Монохромия – полная цветовая слепота.
- Дихроматия – способность распознавать только два цвета.
  - Протанопия – отсутствие пигмента, ответственного за красный цвет.
  - Дейтеранопия – отсутствие пигмента, ответственного за зеленый цвет.

– Тританопия – отсутствие пигмента, ответственного за синий цвет.

Цель работы – реализовать программное обеспечение для изменения цветовой палитры изображения на дисплее с учетом конкретного вида дальтонизма в реальном времени.

Чтобы достигнуть поставленной цели, требуется решить следующие задачи:

- Проанализировать существующие виды дальтонизма;
- Проанализировать существующие алгоритмы преобразования цветов;
- Проанализировать существующие цветовые модели, чтобы выбрать подходящие для решения задачи;
- Спроектировать и реализовать алгоритм изменения цветов изображения с учетом заболевания пользователя;
- Проанализировать существующие симуляторы дальтонизма;
- Сравнить результат с учетом симуляции существующих видов дальтонизма;

Исходя из приведенных выше фактов, стоит *задача* разработать такую программу, которая будет в реальном времени изменять цвета так, чтобы человек страдающий конкретным видом дальтонизма смог различать цвета на дисплее своего персонального компьютера.

1 Аналитическая часть

В данном разделе описаны цветовые модели, пригодные для решения задачи, виды дальтонизма, выбор цветовой модели, анализ существующих решений и возможных решений.

1.1 Формализация задачи

Изображение на дисплее состоит из пикселей. Пиксели состоят из субпикселей: красного, зеленого, синего.

Пиксель – это наименьший логический элемент двумерного цифрового изображения в растровой графике, или физический элемент матрицы дисплеев, формирующих изображение.

Для преобразования цветов на дисплее используются средства операционной системы, таких как: Windows, Mac OS, Linux, Android, IOS.

При работе с цветами в операционных системах используется цветовая модель RGB, RGBA. Однако, для решения поставленной проблемы будут использоваться и другие модели. Общее решение задачи можно представить в виде формулы 1.1

$$CM_{\text{converted}} = F(CM_{\text{initial}}), \tag{1.1}$$

где F - функция преобразования цветовой модели, $CM_{\text{converted}}$ и CM_{initial} - начальная и преобразованная цветовые модели соответственно.

1.2 Дальтонизм

Дальтонизм, цветовая слепота, — наследственная, реже приобретённая, особенность зрения человека и приматов, выражающаяся в сниженной способности или полной неспособности видеть или различать все или некоторые цвета.

1.2.1 Виды дальтонизма

1. Монохроматия — полная цветовая слепота
2. Дихроматия
    - Протанопия — Отсутствие L типа колбочек (Невозможность видеть красный цвет)
    - Дейтеранопия — Отсутствие M типа колбочек (Невозможность видеть зеленый цвет)
    - Тританопия — Отсутствие S типа колбочек (Невозможность видеть синий цвет)
3. Аномальная трихроматия
    - Протаномалия — Проблемы с L типом колбочек (ухудшение видимости красного цвета)
    - Дейтераномалия — Проблемы с M типом колбочек (ухудшение видимости зеленого цвета)
    - Тританомалия — Проблемы с S типом колбочек (ухудшение видимости синего цвета)

1.3 Алгоритмы преобразования цветов

В данном разделе представлены алгоритмы преобразования цветов для коррекции конкретных типов дальтонизма.

1.3.1 Коррекция цвета для дихроматии

Существует множество исследований, направленных на коррекцию цвета для дихроматии. Эти алгоритмы различаются по типам дихроматии. Алгоритмы для коррекции цвета изображения будут представлены далее.

Для коррекции цвета используются следующие алгоритмы:

1. The LMS Daltonization алгоритм.
2. Алгоритм улучшения контрастности цвета.
3. LAB корректировка цвета.
4. Цветовой сдвиг (Color shifting)
5. CBFS алгоритм

K. Эрдоган и H. Ильмаз предложили метод сдвига цвета в цветовой модели HSV [1]. Главная идея состоит в том, чтобы сдвинуть насыщенность цвета (Hue) для всего спектра на конкретное значение $v \in [0.1, 0.9]$.

CBFS алгоритм — алгоритм, с помощью которого проводится коррекция цвета в зависимости от 6 параметров, введенных пользователем. Алгоритм рассчитан на людей с красным и зеленым типом цветовой слепоты. Алгоритм основан на преобразовании цвета из цветовой модели RGB в HSL [2].

C. L. Lai разработал портативную систему для людей с плохим зрением или дальтонизмом. Его система имеет три этапа обработки:

1. Составление модели видимости.
2. Компенсация восприятия.
3. Распознавание шаблонов.

Замеченные выше алгоритмы могут быть использованы для всех типов дихроматии. Следовательно, алгоритмы, описанные в данной работе будут реализованы для конкретных типов цветовой слепоты, но это не исключает их использования для других типов дальтонизма.

[1] HSV (или HSB) - цветовая модель, в которой координатами цвета являются: тон, насыщенность, значение цвета (яркость).

[2] Данные цветовые модели (RGB, HSL) будут рассмотрены далее.

1.3.2 Алгоритмы корректировки цвета для дихроматии

В таблице 1.1 представлены алгоритмы корректировки цвета для различных типов дихроматии.

Таблица 1.1: Алгоритмы корректировки цвета

Тип	Алгоритм
Протанопия	LMS Daltonization & CBFS
Деитеранопия	LMS Daltonization & LAB корректировка
Тританопия	LMS Daltonization & Цветовой сдвиг

1.3.3 LMS Daltonization алгоритм

Данный алгоритм один из самых известных алгоритмов для корректировки цвета для людей с цветовой слепотой. Главная идея заключается в том, чтобы использовать информацию, потерянную в ходе симуляции цветовой слепоты и использовать цветовое пространство LMS, чтобы компенсировать отсутствующие цвета в каждой группе (типе) колбочек: длинные (L), средние (M) и короткие (S).

LMS Daltonization состоит из следующих шагов:

1. Преобразование цветовой модели из RGB в LMS

$$\begin{pmatrix} L \\ M \\ S \end{pmatrix} = \begin{pmatrix} 17.8824 & 43.5161 & 4.11935 \\ 3.45565 & 27.1554 & 3.86714 \\ 0.0299566 & 0.184309 & 1.46709 \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{1.2}$$

2. Симуляция цветовой слепоты, используя формулы: 1.3 для Протанопии, 1.4 для Дейтеранопии и 1.5 для Тританопии:

- $$\begin{pmatrix} L_P \\ M_P \\ S_P \end{pmatrix} = \begin{pmatrix} 0 & 2.02344 & -2.52581 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} L \\ M \\ S \end{pmatrix} \tag{1.3}$$
- $$\begin{pmatrix} L_D \\ M_D \\ S_D \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0.49421 & 0 & 1.24827 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} L \\ M \\ S \end{pmatrix} \tag{1.4}$$

- $$\begin{pmatrix} L_T \\ M_T \\ S_T \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -0.395913 & 0.801109 & 0 \end{pmatrix} \cdot \begin{pmatrix} L \\ M \\ S \end{pmatrix} \tag{1.5}$$

3. Конвертация $L_iM_iS_i$ обратно в $R_iG_iB_i$, используя формулу 1.6, $i = \{P, D, T\}$

$$\begin{pmatrix} R_i \\ G_i \\ B_i \end{pmatrix} = \begin{pmatrix} 0.0809444479 & -0.130504409 & 0.116721066 \\ 0.113614708 & -0.0102485335 & 0.0540193266 \\ -0.000365296938 & -0.00412161469 & 0.693511405 \end{pmatrix} \cdot \begin{pmatrix} L_i \\ M_i \\ S_i \end{pmatrix} \tag{1.6}$$

4. Найти разность между исходным изображением и преобразованным.
   $D_{R(i)} = R - R_i$,
   $D_{G(i)} = G - G_i$,
   $D_{B(i)} = B - B_i$

5. Сдвинуть цвета в сторону видимого спектра с помощью умножения на матрицу ошибок: 1.7 для Протанопии, 1.8 для Дейтеранопии и 1.9 для Тританопии:

   - $$\begin{pmatrix} R_{map(P)} \\ G_{map(P)} \\ B_{map(P)} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0.7 & 1 & 0 \\ 0.7 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} D_{R(P)} \\ D_{G(P)} \\ D_{B(P)} \end{pmatrix} \tag{1.7}$$

   - $$\begin{pmatrix} R_{map(D)} \\ G_{map(D)} \\ B_{map(D)} \end{pmatrix} = \begin{pmatrix} 1 & 0.7 & 0 \\ 0 & 0 & 0 \\ 0 & 0.7 & 1 \end{pmatrix} \cdot \begin{pmatrix} D_{R(D)} \\ D_{G(D)} \\ D_{B(D)} \end{pmatrix} \tag{1.8}$$

   - $$\begin{pmatrix} R_{map(T)} \\ G_{map(T)} \\ B_{map(T)} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0.7 \\ 0 & 1 & 0.7 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} D_{R(T)} \\ D_{G(T)} \\ D_{B(T)} \end{pmatrix} \tag{1.9}$$

6. Сложить исходное изображение со сдвинутыми цветами:
   $R_{F(i)} = R + R_{map(i)}$

$$G_{F(i)} = G + G_{map(i)}$$
$$B_{F(i)} = B + B_{map(i)}$$

1.3.4 CBFS алгоритм

Данный алгоритм использует цветовое пространство HSL. В шаге 2 алгоритма фигурирует параметр близости цвета и определяет насколько близок цвет пикселя в HSL к преобладающему цвету пикселя в RGB и равен абсолютной разнице между ними.

CBFS алгоритм состоит из следующих шагов:

1. Конвертация изображения из цветового пространства RGB в HSL [3]

2. Для каждого пикселя, **если** цвет пикселя достаточно близок к преобладающему цвету изображения (красный/зеленый) **тогда**
   Тон ← Тон - 30%
   Насыщенность ← Насыщенность - 10%
   Светлота ← Светлота + 25%
   иначе
   Насыщенность ← Насыщенность + 10%
   Светлота ← Светлота - 10%

3. Конвертация из HSL в RGB. [4]

1.3.5 Алгоритм корректировки цвета LAB

Данный алгоритм модифицирует значение цвета в зависимости от типа цветовой слепоты следующим образом: алгоритм повышает контрастность необходимых цветов. В втором шаге необходимо отрегулировать значение параметра A. Положительное его значения обозначает, что он ближе к красному, а отрицательный — к зеленому. Значение цвета в LAB принадлежит [-100, 100].

В данном алгоритме не хватает теоретических данных. Так, большинство значений были получены в ходе проб и ошибок в эксперименте с дальтоником.

[3] Преобразование из RGB в HSL описано в главе 1.6
[4] Преобразование из HSL в RGB описано в главе 1.6

Наиболее подходящие преобразования для параметров L, B, A были найдены для дейтеранопии.

Алгоритм корректировки цвета LAB состоит из следующих шагов:

1. Конвертация изображения из цветовой модели RGB в LAB.
2. Регулировка параметра A относительно его максимума: немного увеличить положительные значения и уменьшить отрицательные.
3. Регулировка параметра B в зависимости от того, насколько зеленым или красным он является
4. Изменение параметра L пикселей, который представляет из себя яркость пикселя относительно значений параметра A пикселей.
5. Изображение конвертируется обратно в RGB и складывается с исходным, чтобы гарантировать, что значения цветов пикселей находится между 0 и 1.

1.3.6 Алгоритм цветового сдвига

В данном алгоритме параметр h — коэффициент сдвига и зависит от типа дальтонизма. Для тританопии $h \in [0.1, 0.9]$. Наиболее подходящее значения для h при тританопии — это 0.3.

Данный алгоритм состоит из следующих шагов:

1. Конвертация изображения из RGB в HSV
2. Сдвиг цвета для каждого пикселя (x, y) с помощью уравнения 1.10

$$H_{xy} = H_{xy} + h \tag{1.10}$$

3. Проверка значения параметра H на принадлежность [0, 0.9].
   если $(H_{xy} > 1)$ **то**
   $H_{xy} = 1 - H_{xy}$
   иначе
   $H_{xy} = 0$
4. Конвертация изображения из HSV в RGB

1.4 Цветовая модель

Цветовая модель – термин, обозначающий абстрактную модель описания представления цветов в виде кортежей чисел, называемых цветовыми компонентами или цветовыми координатами. Вместе с методом интерпретации этих данных (например, определение условий воспроизведения или просмотра – то есть задание способа реализации), цвета цветовой модели определяет цветовое пространство.

1.4.1 Цветовая модель RGB

Цветовая модель RGB описывает излучаемые цвета. Она основана на трёх базовых цветах: красный (Red), зелёный (Green) и синий (Blue). Остальные цвета получаются сочетанием базовых. Цвета такого типа называются аддитивными. В компьютерах для представления каждой из координат традиционно используется один октет, значения которого обозначаются для удобства целыми числами от 0 до 255 включительно. Она применяется в приборах, излучающих свет, таких, например, как мониторы, прожекторы, фильтры.

1.4.2 Цветовая модель HSL

Цветовая модель HSL, HLS или HSI (от англ. Hue, Saturation, Lightness (Intensity)) – цветовая модель, в которой цветовыми координатами являются тон, насыщенность и светлота.

- Hue – цветовой тон (например, красный, зелёный или сине-голубой). Варьируется в пределах 0–360°, однако иногда приводится к диапазону 0–100 или 0–1.

- Saturation – насыщенность. Варьируется в пределах 0–100 или 0–1. Чем больше этот параметр, тем «чище» цвет, поэтому этот параметр иногда называют чистотой цвета. А чем ближе этот параметр к нулю, тем ближе цвет к нейтральному серому.

- Lightness (Intensity) – светлота (яркость). Постоянный оттенок (d, h) приводит к вертикальному поперечному сечению. Также задаётся в пределах 0–100 и 0–1.

1.4.3 Цветовая модель HSLuv

Цветовая модель HSLuv является альтернативной версией модели HSL, которая позволяет с большей точностью (градацией) задать контраст цвета.

1.4.4 Цветовая модель CIELAB

Цветовая модель CIELAB – цветовая модель, в которой цветовыми координатами являются светлота от черного до белого, от зеленого до красного и от синего до желтого цветов.

- L – светлота от черного (0) до белого (100).
- A – светлота от зеленого (-) до красного (+).
- B – светлота от синего (-) до желтого (+).

Данная цветовая модель спроектирована так, что любое изменение числовых параметров компонент соответствующе влияет на визуальное восприятие.

1.4.5 Цветовая модель LCH

Цветовая модель LCH – "цилиндрическая"версия цветовой модели CIELAB, что означает, что вместо 6 цветовых параметров, доступных в CIELAB, используется только 4.

- L – светлота от черного (0) до белого (100).
- C – расстояние от серого (0 – 100).
- H – направление цвета (0 – 360) (0 – красный, 90 – желтый, 180 – зеленый, 270 – синий).

1.4.6 Цветовая модель LMS

LMS — цветовое пространство, представляющее собой отклики трёх типов колбочек. В зависимости от спектральной чувствительности существуют L- (long wavelength), M- (middle wavelength) и S- (short wavelength) колбочки.

Человек является трихроматом — сетчатка глаза имеет три вида рецепторов (колбочек), ответственных за цветное зрение. Можно считать, что каждый вид колбочек даёт свой отклик на определённую длину волны видимого спектра.

1.4.7 Цветовая модель HSB (HSV)

HSB (HSV) — цветовая модель, в которой координатами цвета являются:

- Hue — цветовой тон, (например, красный, зелёный или сине-голубой). Варьируется в пределах 0—360°, однако иногда приводится к диапазону 0—100 или 0—1.
- Saturation — насыщенность. Варьируется в пределах 0—100 или 0—1. Чем больше этот параметр, тем «чище» цвет, поэтому этот параметр иногда называют чистотой цвета. А чем ближе этот параметр к нулю, тем ближе цвет к нейтральному серому.
- Value (значение цвета) или Brightness — яркость. Также задаётся в пределах 0—100 или 0—1.

Следует отметить, что HSV (HSB) и HSL — две *разные* цветовые модели.

1.5 Выбор цветовой модели для решения задачи

Для реализации описанных выше алгоритмов потребуются цветовые модели RGB, CIELAB, HSV, LMS.

1.6 Конвертация цветов между цветовыми моделями

В формулах 1.11 – 1.14 представлены шаги для получения компонент цвета цветовой модели HSL из RGB.

$$H = \begin{cases} \text{undefined} & \text{if } MAX = MIN \\ 60° \times \frac{G-B}{MAX-MIN} + 0°, & \text{if } MAX = R \\ & \text{and } G \geq B \\ 60° \times \frac{G-B}{MAX-MIN} + 360°, & \text{if } MAX = R \\ & \text{and } G < B \\ 60° \times \frac{B-R}{MAX-MIN} + 120°, & \text{if } MAX = G \\ 60° \times \frac{R-G}{MAX-MIN} + 240°, & \text{if } MAX = B \end{cases}, \tag{1.11}$$

$$S = \begin{cases} 0 & \text{if } L = 0 \text{ or } MAX = MIN \\ \frac{MAX-MIN}{MAX+MIN} = \frac{MAX-MIN}{2L}, & \text{if } 0 < L \leq \frac{1}{2} \\ \frac{MAX-MIN}{2-(MAX+MIN)} = \frac{MAX-MIN}{2-2L}, & \text{if } \frac{1}{2} < L < 1 \end{cases}, \tag{1.12}$$

или, в общем случае

$$S = \frac{MAX - MIN}{1 - |1 - (MAX + MIN)|}, \tag{1.13}$$

$$L = \frac{1}{2}(MAX + MIN), \tag{1.14}$$

где:

- R, G, B – значения цвета в цветовой модели RGB, значения в диапазоне [0; 1] (R - красный, G - зелёный, B - синий);
- MAX – максимум из трёх значений (R, G, B);
- MIN – минимум из трёх значений (R, G, B);
- H – тон [0; 360];
- S – насыщенность [0; 1];

- L – светлота [0; 1].

В формулах 1.15 – 1.23 представлены шаги для получения компонент цвета цветовой модели RGB из HSL.

$$Q = \begin{cases} L \times (1.0 + S), & \text{if } L < 0.5 \\ L + S - (L \times S), & \text{if } L \geq 0.5 \end{cases} \tag{1.15}$$

$$P = 2.0 \times L - Q \tag{1.16}$$

$$H_k = \frac{H}{360} \tag{1.17}$$

$$T_R = H_k + \frac{1}{3} \tag{1.18}$$

$$T_G = H_k \tag{1.19}$$

$$T_B = H_k - \frac{1}{3} \tag{1.20}$$

$$\text{if } T_c < 0 \rightarrow T_c = T_c + 1.0 \quad \text{for each } c = R, G, B \tag{1.21}$$

$$\text{if } T_c > 1 \rightarrow T_c = T_c - 1.0 \quad \text{for each } c = R, G, B \tag{1.22}$$

Для каждого цвета $c = R, G, B$:

$$\text{color}_c = \begin{cases} P + ((Q - P) \times 6.0 \times T_c), & \text{if } T_c < \frac{1}{6} \\ Q, & \text{if } \frac{1}{6} \leq T_c < \frac{1}{2} \\ P + \left((Q - P) \times (\frac{2}{3} - T_c) \times 6.0\right), & \text{if } \frac{1}{2} \leq T_c < \frac{2}{3} \\ P, & \text{otherwise} \end{cases} \tag{1.23}$$

1.7 Анализ существующих решений

В ходе поисков во всемирной системе компьютерной сети была найдена лишь одна программа Visolve.

1.8 Анализ существующих симуляторов цветовой слепоты

Во всемирной сети интернет существует множество симуляторов цветовой слепоты, например:

1. Vischeck
2. Coblis — Color Blindness Simulator

В дальнейшем для проведения экспериментов со сравнением результатов разработанной программы будут использоваться именно эти симуляторы цветовой слепоты.

Вывод

В данном разделе были рассмотрены виды дальтонизма, существующие алгоритмы, цветовые модели и симуляторы цветовой слепоты. В качестве решения задачи были выбраны алгоритмы, указанные в таблице 1.1.

2 Конструкторская часть

В данном разделе представлены требования к программному обеспечению, схема выбранного для решения поставленной задачи алгоритма.

2.1 Требования к программному обеспечению

Программа должна предоставлять доступ к следующим возможностям:

1. Преобразования цвета с учетом заболевания пользователя.
2. Сохранение визуальной структуры изображения.

К программе предъявляются следующие требования:

1. Программа должна осуществлять изменение цвета в реальном времени.

2.1.1 Общий алгоритм решения задачи

1. Выбор заболевания пользователем.
2. Выбор оптимального алгоритма для этого заболевания.
3. Преобразование текущего изображения, используя выбранный алгоритм.
4. Применить преобразованное изображение.

Вывод

В данном разделе были представлены требования к программному обеспечению и разработана схема реализуемого алгоритма.

3 Технологическая часть

В данном разделе представлены средства разработки программного обеспечения, детали реализации и тестирование функций.

3.1 Средства реализации

В качестве языка программирования, на котором будет реализовано программное обеспечение, выбран язык C++. Выбор языка обусловлен тем, что к разрабатываемому программному обеспечению предъявляются высокие требования к быстродействию и потреблению памяти, а также тем, что в ходе выполнения потребуется иметь доступ к низкоуровненым интерфейсам операционной системы, что делается просто именно на этом языке программирования.

Для тестирования программного обеспечения будет использоваться библиотека Google test (gtest).

Для обеспечения качества кода будут использоваться различные статические и динамические анализаторы, такие как:

- Cppcheck
- Valgrind
- Drmemory
- Heaptracker

Такой богатый выбор анализаторов обусловлен тем, что в языке C++ часто встречаются различные ошибки при работе с памятью.

В качестве компилятора для Unix операционных систем был выбран Clang. Для операционной системы Windows MVSC (Microsoft visual studio compiler).

В качестве среды разработки была выбрана среда CLion, так как она предоставляет большой выбор различных полезных возможностей: автодополнения, статическая проверка кода, компиляция.

В качестве библиотеки для реализации графического интерфейса была выбрана библиотека Qt.

3.2 Детали реализации

В данном разделе представлены листинги реализаций структур данных и контроллеров.

Листинг 3.1: Структура пикселя

```
class pixel {
    public:
    union {
        struct {int  _r, _g, _b;};
        int raw[3];
    };
    pixel(int r, int g, int b) : _r(r), _g(g), _b(g) {}

    int operator [](int i) {
        return raw[i];
    }
};
```

Листинг 3.2: Контроллер корректировки изображения

```
class controller {
    public:

    getPixel(int x, int y);
    setPixel(int x, int y, pixel &p);
};
```

Вывод

В данном разделе были представлены средства реализации программного обеспечения и детали реализации.

Заключение

Во время выполнения поставленной задачи были рассмотрены и проанализированы основные алгоритмы коррекции цвета изображения, виды дальтонизмов, цветовые модели. Были проанализированы достоинства и недостатки рассмотренных алгоритмов, выбраны наиболее подходящие.

В ходе выполнения поставленной задачи были изучены возможности различных библиотек и языка C++, операционных систем и различные факты о заболевании дальтонизм.

Литература

1. Квасова М. Д. Зрение и наследственность. — Москва / Санкт-Петербург: Диля, 2002. — 160 с. — (Здоровое зрение). — 7000 экз.

2. Рабкин Е. Б. Полихроматические таблицы для исследования цветоощущения. — Минск, 1998.

3. Бьёрн Страуструп. Язык программирования C++ = The C++ Programming Language / Пер. с англ. — 3-е изд. — СПб.; М.: Невский диалект — Бином, 1999. — 991 с. — 3000 экз. — ISBN 5-7940-0031-7 (Невский диалект), ISBN 5-7989-0127-0 (Бином), ISBN 0-201-88954-4 (англ.).

4. Visual Studio Code – Code Editing. Redefined [Электронный ресурс]. Режим доступа: https://code.visualstudio.com/ (дата обращения: 26.09.2021).

5. Lamiaa A. Elreaei. Smartphone Based Image Color Correction for Color Blindness // International Journal of Interactive Mobile Technologies. - 2018. - С. 104-119.

6. Color Blindness – learn all about it [Электронный ресурс]. Режим доступа: https://www.color-blindness.com (дата обращения: 26.09.2021).

7. Vischeck [Электронный ресурс]. Режим доступа: http://www.vischeck.com (дата обращения: 26.09.2021)

8. Visolve — The assistive software for people with color blindness [Электронный ресурс]. Режим доступа: https://www.ryobi.co.jp/products/visolve/en/ (дата обращения: 26.09.2021)