p03 Final deliverable

Published

Getting Started

This deliverable will improve upon the previous deliverable so find your colorvisionproject on Replit and start updating it **within this new v3 Replit (https://replit.com/team/csci111-2023-fall/ColorVisionProject-v3)**

Note that extensions cannot be granted for p03. If you have a compelling reason why you need an extension, contact the instructor, but accommodating an extension probably include receiving an Incomplete for your grade this semester.

Assignment

Your final deliverable should experiment with techniques for addressing color combinations that are safe for people with colorblindness.

To be innovative, we need to make incremental progress and experiment with new solutions. Build upon your previous work to meet your goal:

1. Analyzer: Provided a palette of colors, evaluate to what degree its color combinations are problematic for people with a particular form of color blindness
2. Optimizer: Provided a palette of colors, adjust the colors when necessary to optimize the percentage of people who will be able to easily detect the color differences
3. Designer: Provided a base color and a number of additional colors needed, generate a palette that incorporates a the base color with others without problematic color combinations

Develop your program to reach a unique approach to your goal and then **update the README.md file to replace all text that begins and ends with !!! to document your particular program**.

This deliverable should STILL demonstrate use of "C++ Fundamentals" including:

- Variables - more than one variable and use of more than one data type
- Console Input and Output - appropriate use of both standard input and output

- Decisions - more than one use of decisions, including use of more than one of the following: if, else if, else, switch
- Iteration - appropriate use of at least one loop
- File Input and Output - appropriate use of file input and/or output
- Arrays/Vectors - appropriate use of at least one array or vector
- Functions - appropriate use of at least two functions. Each function should adopt proper prototype, definition, and call(s). You should also demonstrate appropriate use of **both** pass-by-value AND pass-by-reference parameters as well as more than one return type. void can be considered one of the return types, if appropriate to your function.

In addition, it should demonstrate the following concepts:

- Classes - appropriate use of at least one class that you have created yourself. The class should demonstrate good practices in Object-Oriented Programming design.
- Style - code should be easy to read, adopt best practices, and use style as demonstrated in the style guide (below).

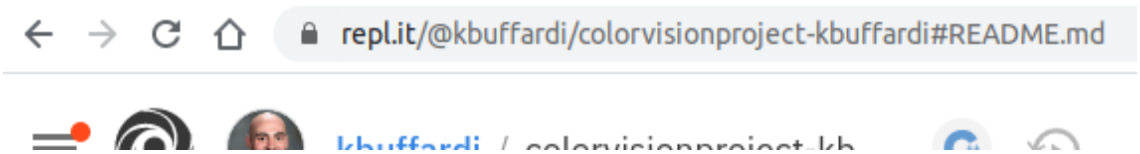
Submitting

Before submitting, don't forget to update your documentation in README.md!

1. Under "v3 Updates" describe what is new to your program (including if any bugs have been fixed) and how new functionality works
2. Under "When the program runs, do the following:" provide updated, detailed instructions
3. Update the documentation of any of the previous requirements if you have made changes/corrections
4. Under "Classes" describe what you have used and why it shows the appropriate use of object-oriented design

Then click on this assignment title and select to **submit**, where you can **paste a URL**

Then return to your Replit project and **copy the URL** from your browser:



Then **paste the URL in the Canvas submission** you already started

Press **Submit** and wait for a confirmation it has been submitted.

Grading
You will be graded on the following:

- **Useful Progress (20%)** - The program has clear use, no known faults, and the documentation is clear and makes the program easy to use
- **C++ Fundamentals (20%)** - Both the use and documentation show clear understanding and appropriate use of variables, console IO, decisions, iteration, file IO, arrays/vectors, and functions
- **Classes (50%)** - Appropriate use of at least one newly-created class. The class should have dedicated .h and .cpp files, adopt good practices of encapsulation and information hiding, and be adopted as the data type for at least one variable in the main program.
- **Style (10%)** - Demonstration of best practices and style that makes the program easy to read, debug, and maintain

Points	100
Submitting	a website url

Due	For	Available from	Until
Dec 8, 2023	Everyone else	-	-
Dec 9, 2023	2 students	-	-

Third Deliverable

Criteria	Ratings					Pts
Useful Progress	**20 pts** **Proficient** The program has clear use, no known faults, and the documentation is clear and makes the program easy to use.	**15 pts** **Sufficient** The program has use and has very few/no known faults. The documentation is informative enough to use without help.	**10 pts** **Competent** The program has use and has limited number of faults with user interaction. The documentation and implementation could use improvement.	**5 pts** **Novice** The program runs but its functionality is quite limited, difficult to use, or has faults.	**0 pts** **None** The program does not compile or the program does not do anything useful.	20 pts
C++ Fundamentals	**30 pts** **Proficient** Both the use and documentation show clear and evident understanding of the concept	**22.5 pts** **Sufficient** The concept is fairly consistent with best practices but the documented explanations could use clearer descriptions that demonstrate full understanding	**15 pts** **Competent** Most use is appropriate but there is room for improving how the concept is used and/or the documentation suggests incomplete understanding of the topic	**7.5 pts** **Novice** There is some appropriate use, but the examples suggest some major misunderstandings and/or poor practices	**0 pts** **None** This no appropriate use, or the examples do not demonstrate proper understanding.	30 pts
Classes	**40 pts** **Proficient** Both the use and documentation show clear and evident understanding of the concept	**30 pts** **Sufficient** The concept is fairly consistent with best practices but the documented explanations could use clearer descriptions that demonstrate full understanding	**20 pts** **Competent** Most use is appropriate but there is room for improving how the concept is used and/or the documentation suggests incomplete understanding of the topic	**10 pts** **Novice** There is some appropriate use, but the examples suggest some major misunderstandings and/or poor practices	**0 pts** **None** This no appropriate use, or the examples do not demonstrate proper understanding.	40 pts
Style and Best Practices	**10 pts** **Proficient** The code is clean, easy to ready and understand, and adopts best practices for software development	**7.5 pts** **Sufficient** The code is mostly clean, easy to ready and understand, and adopts best practices for software development.	**5 pts** **Competent** The code is relatively easy to understand but needs more attention to improving the	**2.5 pts** **Novice** The code is difficult to read and/or understand, and needs significant effort to improving the	**0 pts** **None** The code demonstrates little or no attention to adopting good style and practices	10 pts

Criteria	Ratings			Pts
	However, it has a few places where style and practices should be improved	style and best practices	style and best practices	Total Points: 100