C++ Project Report

Prepared by: *Polina Ivanilova*

General Structure of the Project

This project implements basic functionality for working with matrices and shapes. The system supports both black-and-white (BW) and colored (RGB) images, as well as functions for drawing shapes (rectangles, circles, triangles) on these images.

Key Features:

- Utilization of a templated `draw()` function for working with various types of shapes.
- Dynamic memory management using `std::unique_ptr` to handle objects of type `Color`.
- Integration with OpenCV libraries for displaying and saving images.

Implementation of the `draw()` Function

The `draw()` function is implemented in header files (`bw_matrix.h` and `rgb_matrix.h`):

```
    template<typename T>
    void draw(T& shape){}
```

Advantages:

- Thanks to the use of templates, `draw()` can handle any shape (rectangles, circles, etc.) with any color either black-and-white (`BWColor`) or colored (`RGBColor`) properties.

Limitations:

1. Lack of Virtuality: Template functions do not support virtual methods. This is because a template function must be defined at compile time, while virtual methods require dynamic polymorphism (resolved at runtime).
2. Code Placement: Template functions must be implemented in header files (`.h`) since they are not visible during compilation when defined in `.cpp` files.

Handling Colors (`Color.h`, `BWColor.cpp`, `RGBColor.cpp`)

Shapes inherit `std::unique_ptr<Color>` via their constructor, which ensures safe management of dynamically allocated memory for color objects. Example:

```
  this->color = std::move(color);
```

Using `std::unique_ptr` eliminates the possibility of memory leaks and simplifies working with `Color` objects. And we don’t need a destructor.

Performance Notes

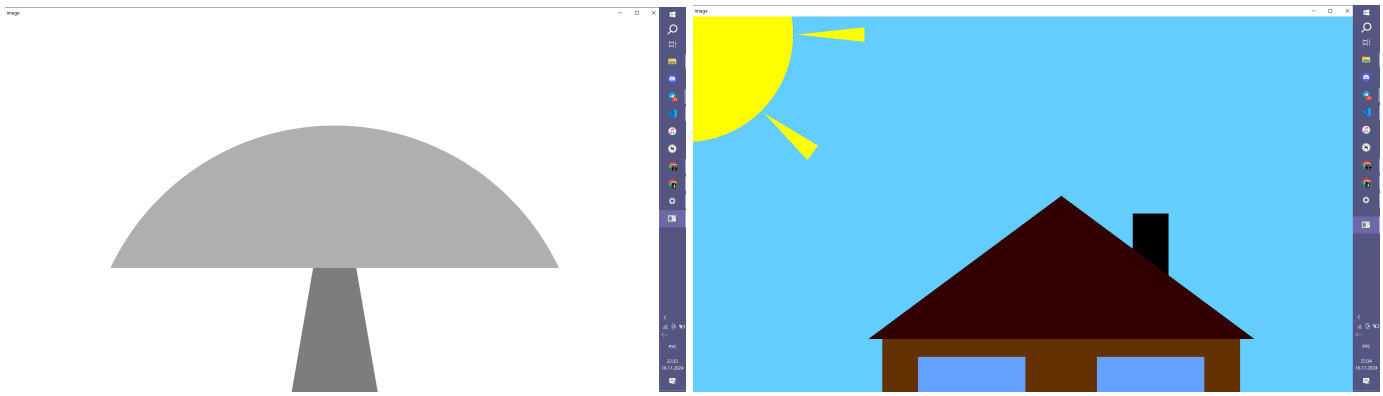
The program's execution time depends on how quickly the graphical window (triggered by the `matrix.display()` function) is closed. Removing this display() call significantly speeds up the program.

```
// bw.display();
```

```
Mushroom small time: 105.503ms
House small time: 216.068ms
Mushroom time: 405.499ms
House time: 854.95ms
Fish time: 137.168ms
```

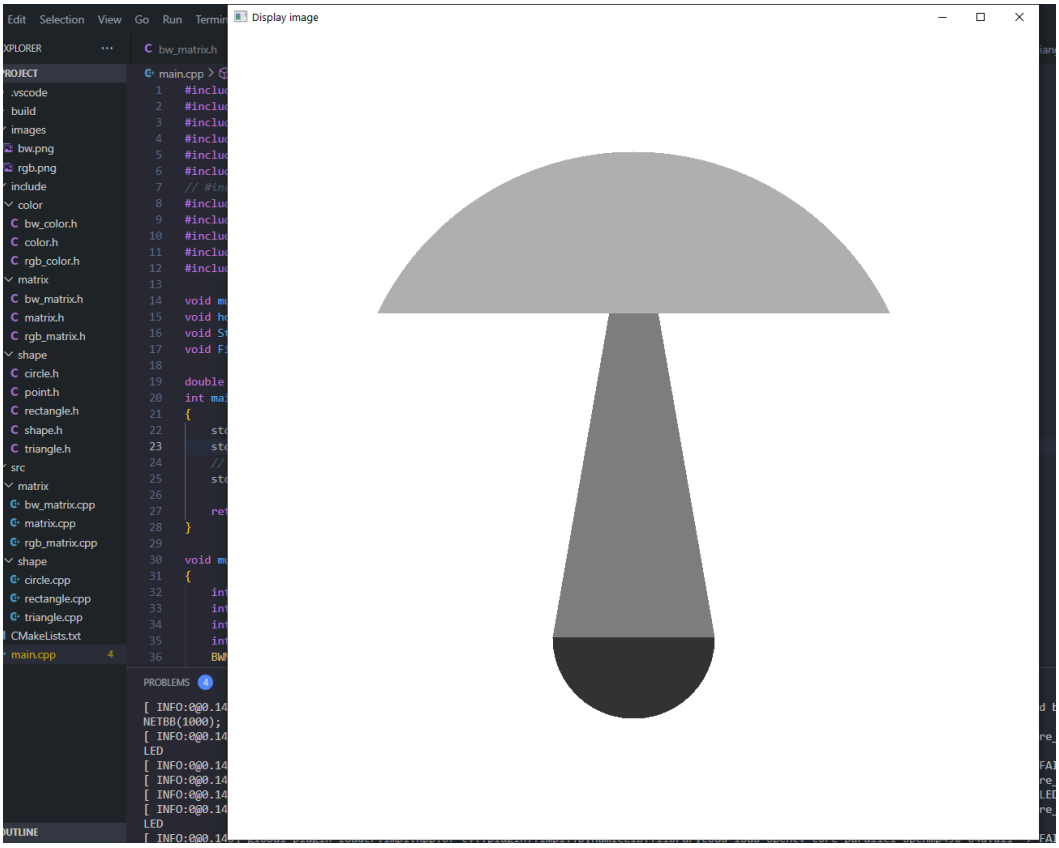
Examples of Execution

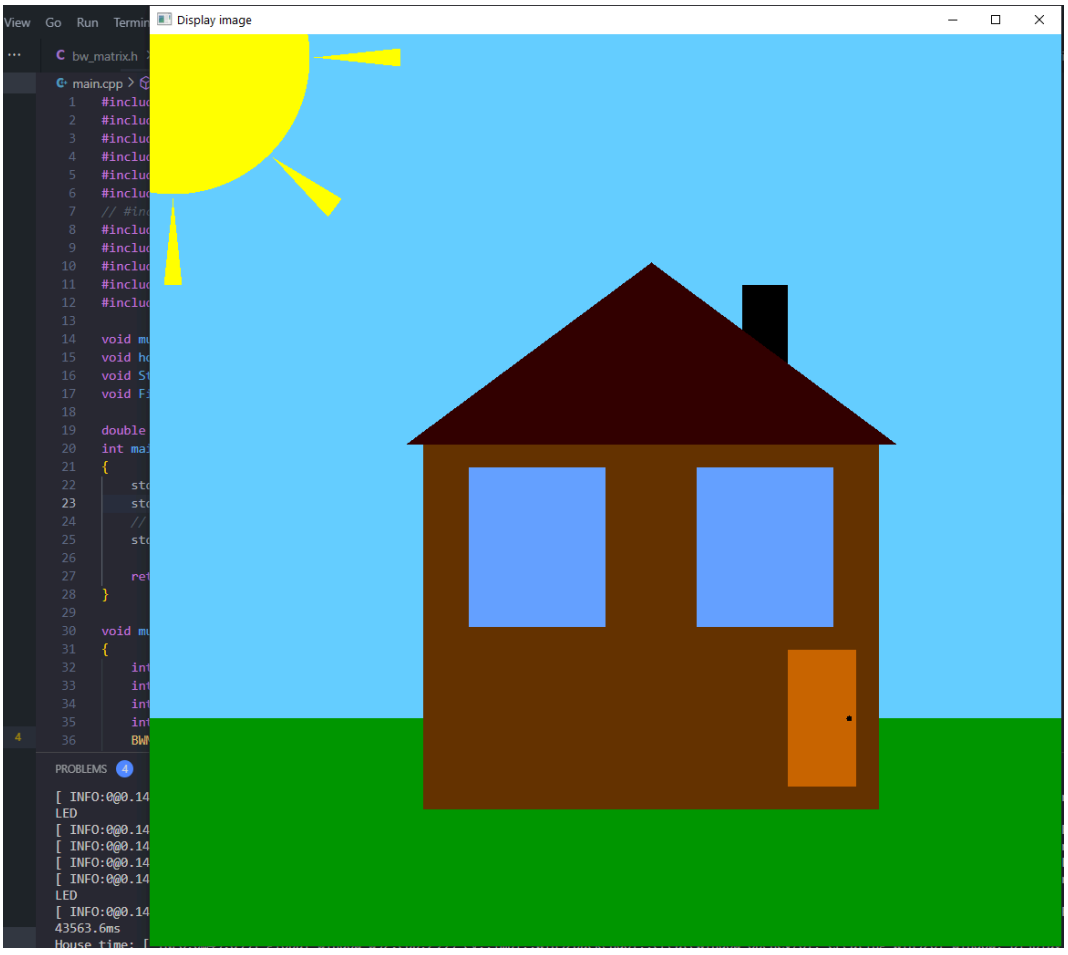
Issues with Initial Examples



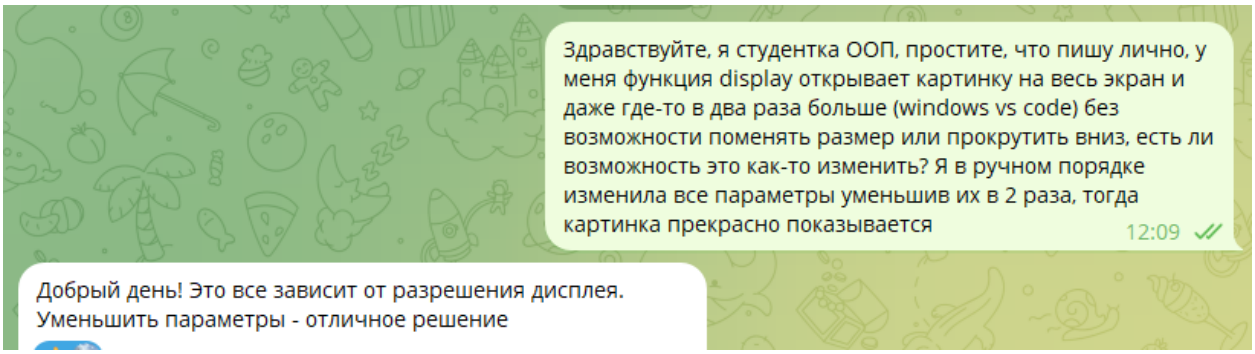
The initial examples (`mushroom()` and `house()`) opened images that were too large on my laptop, making them inconvenient to view. To resolve this:I reduced all coordinates and shape sizes by half. This resulted in the following versions:

- `mushroom_small()`
- `house_small()`


Display image

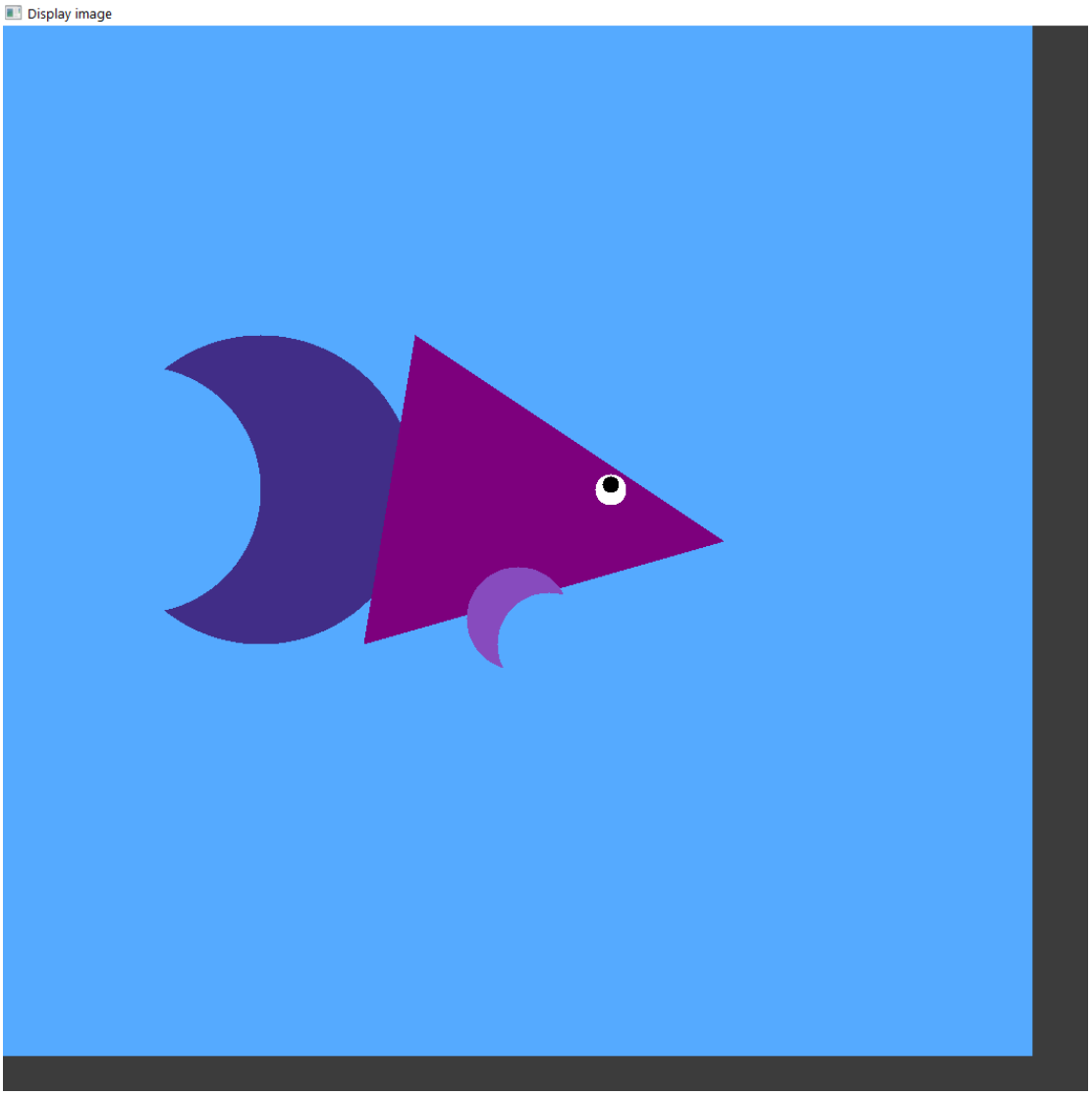

Display image

This modification was approved by the coordinator.



My Example: Fish

My example, `void Fish()`, was added to draw a fish figure.



Conclusion

The project successfully implements the assigned tasks:

- Support for templated functions to work with various data types.
- Safe memory management via `std::unique_ptr`.
- Interaction with the OpenCV library for displaying and saving images.

The modifications and improvements made the program more adaptable to real-world usage and more user-friendly.