

# GUI C++ Windows Desktop Developer

## recruitment task

Greetings,

This task is aimed at testing your proficiency in the area that is key for our product. Thank you for participating.

For this position we are looking for a person who enjoys working with application GUI and would like to assess your ability to work with a chosen GUI framework (Qt or Dear ImGui), understanding color spaces and maintaining a simple application state. Our project caters to the customers who value both usability and aesthetics, and they will both come into play when we assess your input.

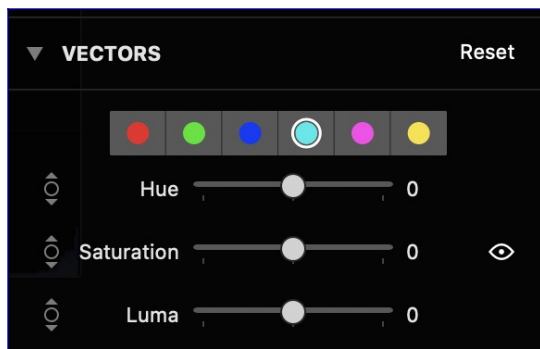
This task is relatively simple and in typical work conditions should not take you more than 8 hours to complete. We understand that it might be a serious time commitment for you, but we also need to make sure that you really are the person who can deliver what we need. As you will see in the deliverables section, we will not be asking for your code without compensation, however we will only ask for it if we like your deliverable.

It is OK to let us know that you are not able to complete or do not even want to approach the test. Also, if you spend more than 24 hours on this task, it might not be a job for you.

## Specification

Create a desktop application using your chosen framework (Qt or ImGui) that:

- Reads an image from file and displays it on screen.
- When a user clicks on the image, the application will:
  - mark the spot with a visible but not distracting cross,
  - sample the image color,
  - convert sampled color to normalized RGBA float values (0.0-1.0 instead of 0-255),
  - store the sampled color in a local variable,
  - show the values somewhere else on screen in the correct order.
- Additional clicks will move the sample point to another spot and update the values.
- When displaying the color values, do your best to stay consistent with the look of the 6-Vector Component shown below.
- Create the 6-Vector Component that looks like this or very similar:



- This component should display the following behavior:
  - Hue slider has range -180.0 to +180.0 with 0.1 precision
  - Other sliders have range -100 to 100 with 0.1 precision
  - Values of all 3 sliders are stored for each of the six colors (RGBCMY) as float values.
  - Clicking on a given color button marks it as active (white circle around it) and the sliders should update based on the stored state and selected active color.
  - Reset should reset all values for all colors.
  - Eyeball icon should disable or enable the color controls and the sliders and gray out the values when disabled
  - You can disregard the icons on the left to the description
  - For bonus points you may animate the open/close of the controller when the triangle in the upper-left corner is pressed
- Make sure that the refresh loop does not consume CPU and GPU resources when not changed (avoid constant refresh)

## Deliverable

- Executable that we can run on our machine

Please use any of your favorite file sharing services to pass the deliverables to us. Send links to the following email address: [jobs@colorgradingcentral.com](mailto:jobs@colorgradingcentral.com)

## Other considerations

- Compile for 64-bit version of Windows.
- If you need better resolution graphics for the controller, let us know.
- Styling the components to escape the default look is an extremely important part of this task.

If you have technical questions, send them to [bart@creativeimpatience.com](mailto:bart@creativeimpatience.com).

Thank you and good luck!