Relatório de desenvolvimento.

O arquivo executável da aplicação se encontra na pasta release e se chama Pixeon.

Abaixo você poderá ler um breve relatório sobre o desenvolvimento desta aplicação. Ela foi desenvolvida em C++ com o Qt versão 6.1.2. Links para as referências estudadas estão no final do arquivo. Esta aplicação foi desenvolvida como parte de um processo seletivo.

**Metodologia e técnicas aplicadas**

Apesar de nunca ter trabalhado com a plataforma QT eu já havia trabalhado com a linguagem C++. Eu dividi a tarefa em partes, primeiramente criando uma aplicação bem simples que carregava uma única imagem num QLabel.

Depois criei uma versão com as ações básicas de manipulação da imagem: Zoom in, Zoom Out, rotações para ambos os sentidos e mais algumas opções.

Em seguida, habilitei o QDialogFile para aceitar múltiplos arquivos e com o QListView eu criei uma simples lista com todas as “imagens” selecionadas pelo usuário. Alterei minhas funções de carregamento de imagem, e configurei um evento ou Signal para detectar quando a seleção da lista muda.

Ao mudar a seleção eu carrego o arquivo selecionado. Não estou mantendo as imagens em memória, apenas carregando uma imagem por vez conforme a seleção do usuário.

E por último desenvolvi os Sliders para poder controlar o brilho e o contraste da imagem selecionada. Para o ajuste desses efeitos foi necessário criar uma cópia da imagem original em memória para não haver perda de informação dos pixels da imagem.

Finalizando eu ajustei para que ambos os sliders funcionem ao mesmo tempo e também repliquei as transformações de zoom e escala para a imagem original.

**Dificuldades**

Como eu dividi a tarefa em partes, foi possível também dividir as dificuldades em pacotes menores. Mas a tarefa mais difícil com certeza foi aplicar o efeito de brilho e contraste na imagem. Os cálculos e peculiaridades no tratamento de imagens tornou essa tarefa mais desafiadora.

**Melhorias**

Como não possuo um conhecimento aprofundado nas ferramentas de layout do Qt Creator, uma melhoria no layout da aplicação seria possível , ainda que esse não seja o foco da tarefa. Também não estou completamente satisfeito com a ferramenta de ajuste de contraste criada, gostaria que seu efeito fosse mais acentuado.

Notei que através do qml, seria possível criar esses mesmos sliders de forma mais simples e rápida, pois as operações seriam realizadas pela biblioteca OpenGL.

Outra melhoria que eu gostaria de implementar seria manter as alterações de cada imagem, não necessariamente salvar elas em disco, mas manter suas matrizes em memória para que quando o usuário selecionasse a mesma imagem, as alterações aplicadas permanecessem.

**Resultado**

Estou satisfeito com o resultado, creio ter executado a maior parte, senão todos os requisitos exigidos para a conclusão do desafio técnico. Ainda que melhorias sejam possíveis. Aprendi sobre desenvolvimento numa nova plataforma e aprendi como alterar pixel a pixel de uma imagem.

O código fonte pode ser achado na pasta Source.

**Referências:**

<https://doc.qt.io/qt-5/qtwidgets-widgets-imageviewer-example.html>

<https://www.qtcentre.org/threads/25346-Image-contrast>

<https://www.ics.com/blog/qt-graphical-effects-module>