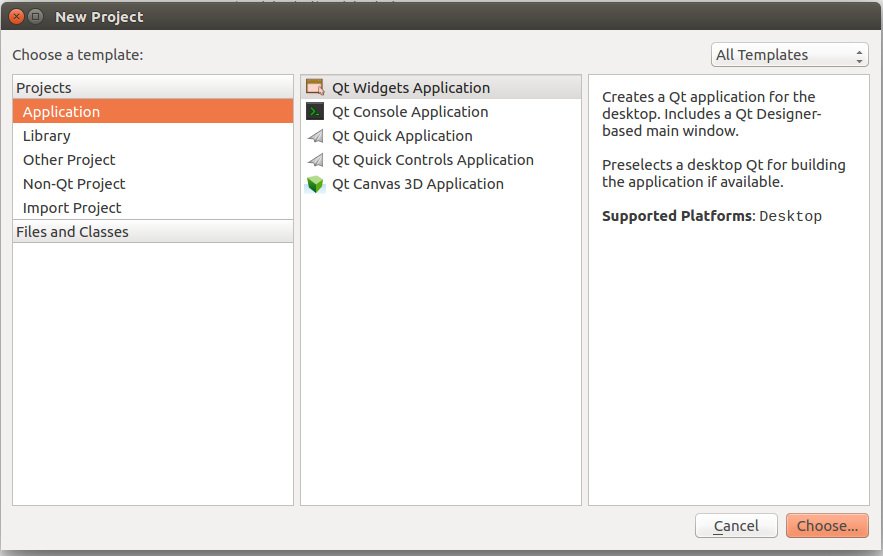
Organização do código:

Nosso grupo criou um novo projeto no QtCreator, na versão utilizada foi possível criar um Qt Widgets Application ao invés de um Qt Gui Application, como demonstrado no artigo que usamos de base para este trabalho.



Depois disso as configurações foram similares às do artigo.

Com isso são criados 4 arquivos:

1. myqtglproject.pro: configurações do projeto Qt project. Contém todos os arquivos e módulos usados.
2. mainwindow.h / mainwindow.cpp: responsável pela janela principal
3. main.cpp: contém as funções principais do programa (em C++)
4. mainwindow.ui: contém a descrição XML da interface do programa

Inicialmente são adicionados somente os módulos core e gui. Para ativar o módulo QtOpenGL é necessário adicionar o opengl no myqtglproject.pro

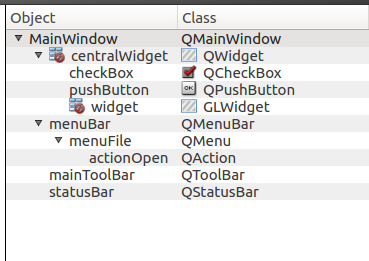


Depois que o projeto foi criado, é necessário adicionar uma classe GLWidget que estende a classe QGLWidget. Na classe são implementados os métodos:

1. GLWidget::GLWidget(QWidget \* parent): QGLWidget(parent) : Construtor do GLWidget
2. GLWidget::~GLWidget() : Destruidor do GLWidget
3. void GLWidget::initializeGL() : É chamado toda vez que o widget vai abrir um novo contexto OpenGL, inicializando algumas das variáveis
4. void GLWidget::paintGL() : Renderiza o OpenGL quando o widget precisa ser pintado novamente
5. void GLWidget::resizeGL(int w, int h) : Lida com o redimensionamento da janela do OpenGL
6. void GLWidget::toggleBackgroundColor(bool toBlack) : Troca a cor do widget de preto para branco e vice-versa
7. void GLWidget::showFileOpenDialog() : Mostra a janela para o usuário selecionar o arquivo off que deseja abrir e retorna o caminho completo para o arquivo
8. void GLWidget::readOFFFile(const QString & fileName) : Faz a leitura do arquivo off usando a biblioteca ifstream
9. void GLWidget::genNormals() : Calcula as normais nos vértices da malha que são salvas como objetos Qvector3D no array
10. void GLWidget::genTexCoordsCylinder() : Gera as coordenadas cilíndricas das texturas para os vértices da malha
11. void GLWidget::genTangents() : Faz a estimativa de vetores tangentes para cada vértice, a função foi baseada no método de Langeyl que usa o mapeamento de normais como base
12. void GLWidget::createVBOs() : Cria os Vertex Buffer Objects (VBOs) para fazer a renderização das malhas, também permitem a manipulação direta dos dados do servidor
13. void GLWidget::destroyVBOs() : Destrói os VBOs criados
14. void GLWidget::createShaders() : Método responsável por liberar e destruir o sombreamento anterior e ligar o novo, essa ligação será feita pelo método paintGL()
15. void GLWidget::destroyShaders() : Deleta os sombreamentos aplicados no programa
16. void GLWidget::keyPressEvent(QKeyEvent \* event) : Lida com os eventos vindos do teclado, ele troca os efeitos no objeto - na tecla 0 coloca o Gouraud shading, na tecla 1 o Phong shading, na 2 o Phong +texture e na 3 o normal mapping, também é possível fechar o programa na Esc
17. void GLWidget::mouseMoveEvent(QMouseEvent \* event) / void GLWidget::mousePressEvent(QMouseEvent \* event) / void GLWidget::mouseReleaseEvent(QMouseEvent \* event) : Lida com os eventos vindos do mouse
18. void GLWidget::wheelEvent(QWheelEvent \* event) : Lida com os eventos vindos do mouse wheel
19. void GLWidget::animate()

Depois de os métodos estarem funcionando e em ordem, criamos a interface do programa





A interface é bem simples, possuindo um widget (que foi promovido para um GLWidget) onde serão abertos os objetos, uma checkbox que está ligada ao método “toggleBackgroundColor” , um pushButton para fechar o programa e um menuBar que está ligado ao método “showFileOpenDialog” que abre a janela para selecionar o arquivo desejado

Problemas encontrados:

Dentro do método readOFFFile não foi possível encontrar a função 'toAscii', portanto

stream.open( fileName.toAscii(), std::ifstream::in);

foi substituído por

stream.open(fileName.toLatin1(), std::ifstream:: in );

Manual do usuário:

1. Abrindo o arquivo OFF, ele será carregado no widget e controlado pelo mouse
2. Para trocar entre as texturas é só apertar no teclado as teclas 0,1,2 e 3

Especificações da máquina:

Linux Ubuntu 16.04 LTS Qt creator 3.5.1 Qt 5.5.1

Testes:

