

O projeto possui um arquivo chamado `cg.pro`, que contém as configurações, e diversas classes. Cada uma das classes possui um arquivo header, que contém as declarações dos métodos e atributos, enquanto os arquivos `.cpp` possuem as definições dos mesmos.

Main e Mainwindow

Responsáveis pela inicialização do programa.

Camera

Contém os atributos `QVector3D eye`, `at` e `up`, que descrevem a posição do observador.

Light

Contém os atributos `QVector4D position`, `ambiente`, `diffuse` e `especular`, que descrevem a fonte de luz.

Material

Contém os atributos `ambiente`, `diffuse`, `specular` e `shininess`, que descrevem a coloração do objeto que está sendo observado.

Trackball

Utilizada para administrar a movimentação na janela em que o objeto está sendo renderizado. Possui os atributos `rotation`, `axis`, `velocity`, `lastPos3D`, `lasttime`, `trackingMouse`, `viewportWidth` e `viewportHeight`, responsáveis por descrever o movimento do mouse.

Glwidget

Faz uso das outras classes para renderizar o objeto. Para esses cálculos, faz uso das variáveis câmera, light, material e trackBall que contém informações sobre os objetos criados fazendo uso das classes citadas anteriormente, e os atributos descritos a seguir:

`unsigned int numVertices` -> quantidade total de vértices do objeto.

`unsigned int numFaces` -> quantidade total de faces do objeto.

`QVector4D *vértices` -> matriz com as posições dos vértices.

`QVector3D *normals` -> matriz com os vetores normais.

`QVector2D *texCoords` -> matriz com as coordenadas da textura.

`QVector4D *tangents` -> matriz com as tangentes.

`unsigned int *indices` -> matriz com os índices que descrevem as faces (cada 3 índices seguidos descrevem os vértices que compõem a face).

Para maior agilidade no cálculo, esses atributos são depois armazenados em VBOs (Vertex Buffer Object). Os métodos que administram esses VBOs são os seguintes `vboVertices`, `vboNormals`, `vboTexCoords`, `vboTangents`, `vboIndices`, `createVBOs` e `destroyVBOs`.

`QGLShader *vertexShader` -> armazena os caminhos dos arquivos de vertex shading.

`QGLShader *fragmentShader` -> armazena os caminhos dos arquivos de fragment shading.

`QGLShaderProgram *shaderProgram` -> armazena informações sobre o shader.

`unsigned int currentShader` -> contém o índice do shader que está sendo utilizado no momento.

`QMatrix4x4 modelViewMatrix` -> matriz com informações sobre o objeto renderizado.

`QMatrix4x4 projectionMatrix` -> matriz com dados sobre a projeção do objeto.

A classe glwidget define os seguintes métodos:

statusBarMessage -> Mostra as informações sobre a quantidade de vertices e faces do objeto.
setShaderIndex -> Altera o shader da comboBox para o shader que está sendo utilizado no momento.

showFileDialog() -> selecionar arquivo off.

animate() -> Atualiza o objeto renderizado.

takeScreenshot() -> Salva uma imagem do objeto com o nome e caminho escolhidos.

definirCorFundo() -> Altera a cor de fundo para o valor selecionado .

changeShader(int index) -> altera o shader de acordo com o indice passado pela comboBox dos shaders.

Métodos para alterar valores das cores do material e da fonte de luz: changeAmbientLight, changeDiffuseLight, changeSpecularLight, changeAmbientMaterial, changeDiffuseMaterial, changeSpecularMaterial.

redefinirObjeto -> redefine as cores e o shader para o padrão.

initializeGL -> inicializar texturas.

resizeGL -> alterar perspectiva da matriz de projecao para redimensionar o objeto.

paintGL -> pinta o objeto no widget.

Métodos para controlar os movimentos do mouse: mouseMoveEvent, mousePressEvent, mouseReleaseEvent, wheelEvent.

keyPressEvent -> Troca os shaders dependendo da tecla apertada.

readOFFFile -> ler arquivo off e armazenar informacoes sobre vertices e faces. Caso existam faces com 4 vértices, as divide em 2 faces com 3 vértices cada.

genNormals -> calcula a normal em relacao a cada vértice.

genTexCoordsCylinder -> gerar coordenadas da textura.

genTangents -> calcular tangentes por vértice.

createShaders -> inicializa os shaders.

destroyShaders -> destrói os shaders.