

3D Model Viewer

Generated by Doxygen 1.9.4

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

BaseModel	??
ModelObj	??
ModelStl	??
Camera_t	??
face	??
QGLWidget	
OpenGLView	??
QMainWindow	
MainWindow	??
QSlider	
CustomSlider	??
vertex	??

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

BaseModel	??
Camera_t	??
CustomSlider	??
face	??
MainWindow	??
ModelObj	??
ModelStl	??
OpenGLView	??
vertex	??

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

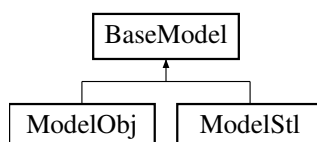
BaseModel.h	??
CustomSlider.h	??
mainwindow.h	??
modelobj.h	??
modelstl.h	??
openglview.h	??

Chapter 4

Class Documentation

4.1 BaseModel Class Reference

Inheritance diagram for BaseModel:



Public Member Functions

- void [updateModelSourceFile](#) (QString file)
- virtual void [LoadMdl](#) ()
- vtxArray [GetVertices](#) ()
- faceArray [GetFaces](#) ()
- vtxArray [GetNormals](#) ()

Protected Attributes

- QFile * **modelFile**
- vtxArray [vtxArr](#)
- vtxArray [normalsArr](#)
- faceArray [fArr](#)

4.1.1 Member Function Documentation

4.1.1.1 GetFaces()

```
faceArray BaseModel::GetFaces ( )
```

This Method returns the list of faces.

Returns

Array of model triangle faces.

4.1.1.2 GetNormals()

```
vtxArray BaseModel::GetNormals ( )
```

This Method returns the list of normal vectors.

Returns

Array of normal vectors.

4.1.1.3 GetVertices()

```
vtxArray BaseModel::GetVertices ( )
```

This Method returns the list of vertices.

Returns

Array of model triangle vertices.

4.1.1.4 LoadMdl()

```
void BaseModel::LoadMdl ( ) [virtual]
```

This method loads 3d model.

Reimplemented in [ModelObj](#), and [ModelStl](#).

4.1.1.5 updateModelSourceFile()

```
void BaseModel::updateModelSourceFile (
    QString file )
```

This method updates 3d model file path.

Parameters

<i>IN::file</i>	3d model file path
-----------------	--------------------

4.1.2 Member Data Documentation

4.1.2.1 fArr

```
faceArray BaseModel::fArr [protected]
```

Array of normal vectors

4.1.2.2 normalsArr

```
vtxArray BaseModel::normalsArr [protected]
```

Array of vertices

4.1.2.3 vtxArr

```
vtxArray BaseModel::vtxArr [protected]
```

3D Model File

The documentation for this class was generated from the following files:

- BaseModel.h
- BaseModel.cpp

4.2 Camera_t Struct Reference

Public Attributes

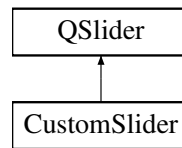
- GLfloat **posX**
- GLfloat **posY**
- GLfloat **posZ**
- GLfloat **rotX**
- GLfloat **rotY**
- GLfloat **rotZ**
- QPoint **lastPos**

The documentation for this struct was generated from the following file:

- openglview.h

4.3 CustomSlider Class Reference

Inheritance diagram for CustomSlider:



Public Member Functions

- **CustomSlider** (Qt::Orientation orientation, QWidget *parent=nullptr)
- **CustomSlider** (QWidget *parent=nullptr)

Protected Member Functions

- virtual void **paintEvent** (QPaintEvent *ev)

The documentation for this class was generated from the following file:

- CustomSlider.h

4.4 face Struct Reference

Public Member Functions

- **face** (const uint xCord, const uint yCord, const uint zCord)

Public Attributes

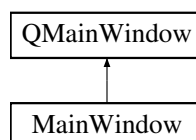
- uint **x**
- uint **y**
- uint **z**

The documentation for this struct was generated from the following file:

- BaseModel.h

4.5 MainWindow Class Reference

Inheritance diagram for MainWindow:



Public Slots

- void [keyPressEvent](#) (QKeyEvent *ke)
- void [onBtnLoadModelClicked](#) ()

Public Member Functions

- **MainWindow** (QWidget *parent=nullptr)

4.5.1 Member Function Documentation

4.5.1.1 keyPressEvent

```
void MainWindow::keyPressEvent (
    QKeyEvent * ke ) [slot]
```

This method intercept all key events.

Parameters

<i>IN::ke</i>	Key event to process.
---------------	-----------------------

4.5.1.2 onBtnLoadModelClicked

```
void MainWindow::onBtnLoadModelClicked ( ) [slot]
```

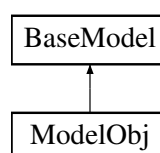
This method handles load new model request.

The documentation for this class was generated from the following files:

- mainwindow.h
- mainwindow.cpp

4.6 ModelObj Class Reference

Inheritance diagram for ModelObj:



Public Member Functions

- void [LoadMdl](#) () override

Additional Inherited Members

4.6.1 Member Function Documentation

4.6.1.1 LoadMdl()

```
void ModelObj::LoadMdl ( ) [override], [virtual]
```

See also

[BaseModel::LoadMdl](#)

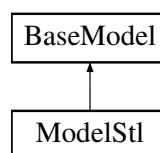
Reimplemented from [BaseModel](#).

The documentation for this class was generated from the following files:

- modelobj.h
- modelobj.cpp

4.7 ModelStl Class Reference

Inheritance diagram for ModelStl:



Public Member Functions

- void [LoadMdl](#) () override

Additional Inherited Members

4.7.1 Member Function Documentation

4.7.1.1 LoadMdl()

```
void ModelStl::LoadMdl ( ) [override], [virtual]
```

See also

[BaseModel::LoadMdl](#)

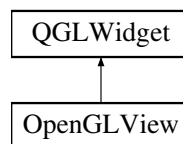
Reimplemented from [BaseModel](#).

The documentation for this class was generated from the following files:

- modelstl.h
- modelstl.cpp

4.8 OpenGLView Class Reference

Inheritance diagram for OpenGLView:



Public Slots

- void [setDirectionalLightMode](#) (const bool isDirectional)
- bool [directionalLightMode](#) () const
- void [setXRotation](#) (int xAngle)
- int [modelXRotation](#) () const
- void [setYRotation](#) (int yAngle)
- int [modelYRotation](#) () const
- void [setZRotation](#) (int zAngle)
- int [modelZRotation](#) () const
- void [setScale](#) (int [scaleFactor](#))
- int [scaleFactor](#) () const
- void [setLightIntensity](#) (int intensity)
- void [setLightXPosition](#) (int xPosition)
- void [setLightYPosition](#) (int yPosition)
- void [setLightZPosition](#) (int zPosition)
- void [setAmbientLightR](#) (double value)
- void [setAmbientLightG](#) (double value)
- void [setAmbientLightB](#) (double value)
- void [setDiffuseLightR](#) (double value)
- void [setDiffuseLightG](#) (double value)
- void [setDiffuseLightB](#) (double value)
- void [setSpecularLightR](#) (double value)
- void [setSpecularLightG](#) (double value)
- void [setSpecularLightB](#) (double value)

- void [setCameraXPosition](#) (int xPosition)
- void [setCameraYPosition](#) (int yPosition)
- void [setCameraZPosition](#) (int zPosition)
- void [setModelXPosition](#) (int xPosition)
- int [modelXPosition](#) () const
- void [setModelYPosition](#) (int yPosition)
- int [modelYPosition](#) () const
- void [setModelZPosition](#) (int zPosition)
- int [modelZPosition](#) () const

Signals

- void [modelXRotationChanged](#) ()
- void [modelYRotationChanged](#) ()
- void [modelZRotationChanged](#) ()
- void [modelXPositionChanged](#) ()
- void [modelYPositionChanged](#) ()
- void [modelZPositionChanged](#) ()
- void [scaleChanged](#) ()

Public Member Functions

- **OpenGLView** (QWidget *parent=0)
- void [keyPressEvent](#) (QKeyEvent *ke) override
- void [paintModelFromFile](#) (const QString filePath)

4.8.1 Member Function Documentation

4.8.1.1 [directionalLightMode](#)

```
bool OpenGLView::directionalLightMode ( ) const [inline], [slot]
```

This method returns directional light mode flag value.

Returns

True if ambient light mode is directional.

4.8.1.2 [keyPressEvent\(\)](#)

```
void OpenGLView::keyPressEvent (
    QKeyEvent * ke ) [override]
```

This method intercept keyboard button pressed event.

Parameters

<i>IN::ke</i>	keyboard event.
---------------	-----------------

4.8.1.3 modelXPosition

```
int OpenGLView::modelXPosition ( ) const [inline], [slot]
```

This method returns model x position.

Returns

Model x position.

4.8.1.4 modelXRotation

```
int OpenGLView::modelXRotation ( ) const [inline], [slot]
```

This method returns X axis rotation value.

Returns

X axis Rotation angle value

4.8.1.5 modelXRotationChanged

```
void OpenGLView::modelXRotationChanged ( ) [signal]
```

Opengl view camera

4.8.1.6 modelYPosition

```
int OpenGLView::modelYPosition ( ) const [inline], [slot]
```

This method returns model y position.

Returns

Model y position.

4.8.1.7 modelYRotation

```
int OpenGLView::modelYRotation ( ) const [inline], [slot]
```

This method returns Y axis rotation value.

Returns

Y axis Rotation angle value

4.8.1.8 modelZPosition

```
int OpenGLView::modelZPosition ( ) const [inline], [slot]
```

This method returns model z position.

Returns

Model z position.

4.8.1.9 modelZRotation

```
int OpenGLView::modelZRotation ( ) const [inline], [slot]
```

This method returns Z axis rotation value.

Returns

Z axis Rotation angle value

4.8.1.10 paintModelFromFile()

```
void OpenGLView::paintModelFromFile (
    const QString filePath )
```

This method requests 3d model painting providing a 3d model file path.

Parameters

<i>In::filePath</i>	3d model file path.
---------------------	---------------------

4.8.1.11 scaleFactor

```
int OpenGLView::scaleFactor ( ) const [inline], [slot]
```

This method returns 3d model scale factor.

Returns

3d model scale factor.

4.8.1.12 setAmbientLightB

```
void OpenGLView::setAmbientLightB (
    double value ) [slot]
```

This method sets ambient light blue component.

Parameters

<i>Ambient</i>	light blue component intensity.
----------------	---------------------------------

4.8.1.13 setAmbientLightG

```
void OpenGLView::setAmbientLightG (
    double value ) [slot]
```

This method sets ambient light green component.

Parameters

<i>Ambient</i>	light green component intensity.
----------------	----------------------------------

4.8.1.14 setAmbientLightR

```
void OpenGLView::setAmbientLightR (
    double value ) [slot]
```

This method sets ambient light red component.

Parameters

<i>Ambient</i>	light red component intensity.
----------------	--------------------------------

4.8.1.15 setCameraXPosition

```
void OpenGLView::setCameraXPosition (
    int xPosition ) [slot]
```

This method sets camera x position.

Parameters

<i>Camera</i>	x position.
---------------	-------------

4.8.1.16 setCameraYPosition

```
void OpenGLView::setCameraYPosition (
    int yPosition ) [slot]
```

This method sets camera y position.

Parameters

<i>Camera</i>	y position.
---------------	-------------

4.8.1.17 setCameraZPosition

```
void OpenGLView::setCameraZPosition (
    int zPosition ) [slot]
```

This method sets camera z position.

Parameters

<i>Camera</i>	z position.
---------------	-------------

4.8.1.18 setDiffuseLightB

```
void OpenGLView::setDiffuseLightB (
    double value ) [slot]
```

This method sets diffuse light blue component.

Parameters

<i>Diffuse</i>	light blue component intensity.
----------------	---------------------------------

4.8.1.19 setDiffuseLightG

```
void OpenGLView::setDiffuseLightG (  
    double value ) [slot]
```

This method sets diffuse light green component.

Parameters

<i>Diffuse</i>	light green component intensity.
----------------	----------------------------------

4.8.1.20 setDiffuseLightR

```
void OpenGLView::setDiffuseLightR (  
    double value ) [slot]
```

This method sets diffuse light red component.

Parameters

<i>Diffuse</i>	light red component intensity.
----------------	--------------------------------

4.8.1.21 setDirectionalLightMode

```
void OpenGLView::setDirectionalLightMode (  
    const bool isDirectional ) [slot]
```

This method sets ambient light mode

Parameters

<i>IN::isDirectional</i>	Directional light mode flag
--------------------------	-----------------------------

4.8.1.22 setLightIntensity

```
void OpenGLView::setLightIntensity (
    int intensity ) [slot]
```

This method sets light intensity.

Parameters

<i>Light</i>	intensity value.
--------------	------------------

4.8.1.23 setLightXPosition

```
void OpenGLView::setLightXPosition (
    int xPosition ) [slot]
```

This method sets light x position.

Parameters

<i>Light</i>	x position.
--------------	-------------

4.8.1.24 setLightYPosition

```
void OpenGLView::setLightYPosition (
    int yPosition ) [slot]
```

This method sets light y position.

Parameters

<i>Light</i>	y position.
--------------	-------------

4.8.1.25 setLightZPosition

```
void OpenGLView::setLightZPosition (
    int zPosition ) [slot]
```

This method sets light z position.

Parameters

<i>Light</i>	z position.
--------------	-------------

4.8.1.26 setModelXPosition

```
void OpenGLView::setModelXPosition (  
    int xPosition ) [slot]
```

This method sets model x position.

Parameters

<i>Model</i>	x position.
--------------	-------------

4.8.1.27 setModelYPosition

```
void OpenGLView::setModelYPosition (  
    int yPosition ) [slot]
```

This method sets model y position.

Parameters

<i>Model</i>	y position.
--------------	-------------

4.8.1.28 setModelZPosition

```
void OpenGLView::setModelZPosition (  
    int zPosition ) [slot]
```

This method sets model z position.

Parameters

<i>Model</i>	z position.
--------------	-------------

4.8.1.29 setScale

```
void OpenGLView::setScale (
    int scaleFactor ) [slot]
```

This method sets 3d model scale factor.

Parameters

<i>3D</i>	model scale factor.
-----------	---------------------

4.8.1.30 setSpecularLightB

```
void OpenGLView::setSpecularLightB (
    double value ) [slot]
```

This method sets specular light blue component.

Parameters

<i>Specular</i>	light blue component intensity.
-----------------	---------------------------------

4.8.1.31 setSpecularLightG

```
void OpenGLView::setSpecularLightG (
    double value ) [slot]
```

This method sets specular light green component.

Parameters

<i>Specular</i>	light green component intensity.
-----------------	----------------------------------

4.8.1.32 setSpecularLightR

```
void OpenGLView::setSpecularLightR (
    double value ) [slot]
```

This method sets specular light red component.

Parameters

<i>Specular</i>	light red component intensity.
-----------------	--------------------------------

4.8.1.33 setXRotation

```
void OpenGLView::setXRotation (
    int xAngle ) [slot]
```

This method sets X axis rotation value.

Parameters

<i>IN::xAngle</i>	Rotation angle value
-------------------	----------------------

4.8.1.34 setYRotation

```
void OpenGLView::setYRotation (
    int yAngle ) [slot]
```

This method sets Y axis rotation value.

Parameters

<i>IN::yAngle</i>	Rotation angle value
-------------------	----------------------

4.8.1.35 setZRotation

```
void OpenGLView::setZRotation (
    int zAngle ) [slot]
```

This method sets Z axis rotation value.

Parameters

<i>IN::zAngle</i>	Rotation angle value
-------------------	----------------------

The documentation for this class was generated from the following files:

- openglview.h
- openglview.cpp

4.9 vertex Struct Reference

Public Member Functions

- **vertex** (const float xCord, const float yCord, const float zCord)

Public Attributes

- float **x**
- float **y**
- float **z**

The documentation for this struct was generated from the following file:

- BaseModel.h

Chapter 5

File Documentation

5.1 BaseModel.h

```
1 #ifndef BASEMODEL_H
2 #define BASEMODEL_H
3 #include <QtCore>
4
5 struct vertex {
6     float x;
7     float y;
8     float z;
9     vertex() { }
10    vertex(const float xCord,const float yCord,const float zCord):
11        x(xCord),y(yCord) , z(zCord) { }
12 };
13 struct face {
14     uint x;
15     uint y;
16     uint z;
17     face() { }
18     face(const uint xCord,const uint yCord,const uint zCord):
19         x(xCord),y(yCord) , z(zCord) { }
20 };
21
22 typedef std::vector<vertex> vtxArray;
23 typedef std::vector<face> faceArray;
24
25 class BaseModel {
26 public:
27     BaseModel();
28     virtual ~BaseModel();
29     void updateModelSourceFile(QString file);
30
31     virtual void LoadMdl();
32
33     vtxArray GetVertices();
34
35     faceArray GetFaces();
36
37     vtxArray GetNormals();
38
39 protected:
40     QFile* modelFile;
41     vtxArray vtxArr;
42     vtxArray normalsArr;
43     faceArray fArr;
44 };
45 #endif // BASEMODEL_H
```

5.2 CustomSlider.h

```
1 #ifndef CUSTOMSLIDER_H
2 #define CUSTOMSLIDER_H
3 #include <QStylePainter>
4 #include <QStyleOptionSlider>
5 #include <QStyleOptionComplex>
6 #include <QSlider>
```

```

7 #include <QColor>
8 #include "math.h"
9
10 class CustomSlider:public QSlider
11 {
12 public:
13     explicit CustomSlider(Qt::Orientation orientation, QWidget *parent = nullptr):QSlider(orientation,
14         parent){};
15     explicit CustomSlider(QWidget *parent = nullptr):QSlider(parent){
16         this->setStyleSheet("\
17             QSlider::groove:horizontal {\
18                 height: 8px; \
19                 background: qlineargradient(x1:0, y1:0, x2:0, y2:1, stop:0 #B1B1B1,
20                 stop:1 #c4c4c4);\
21                 margin: 2px 0;\
22             }\
23             QSlider::handle:horizontal {\
24                 background: qlineargradient(x1:0, y1:0, x2:1, y2:1, stop:0 #b4b4b4,
25                 stop:1 #8f8f8f);\
26                 border: 1px solid #5c5c5c;\
27                 width: 18px;\
28                 margin: -2px 0; \
29                 border-radius: 3px;\
30             }\
31         ");
32     };
33 protected:
34     virtual void paintEvent(QPaintEvent *ev)
35     {
36         QStylePainter p(this);
37         QStyleOptionSlider opt;
38         initStyleOption(&opt);
39
40         QRect handle = style()->subControlRect(QStyle::CC_Slider, &opt, QStyle::SC_SliderHandle, this);
41
42         int interval = tickInterval();
43         if (interval == 0)
44         {
45             interval = pageStep();
46         }
47
48         if (tickPosition() != NoTicks)
49         {
50             for (int i = minimum(); i <= maximum(); i += interval)
51             {
52                 int x = std::round((double)((double)(i - this->minimum()) /
53                     (double)(this->maximum() - this->minimum())) * (double)(this->width() - handle.width()) +
54                     (double)(handle.width() / 2.0))) - 1;
55                 int h = 4;
56                 p.setPen(QColor("#a5a294"));
57                 if (tickPosition() == TicksBothSides || tickPosition() == TicksAbove)
58                 {
59                     int y = this->rect().top();
60                     p.drawLine(x, y, x, y + h);
61                 }
62                 if (tickPosition() == TicksBothSides || tickPosition() == TicksBelow)
63                 {
64                     int y = this->rect().bottom();
65                     p.drawLine(x, y, x, y - h);
66                 }
67             }
68         }
69         QSlider::paintEvent(ev);
70     }
71 };
72 #endif // CUSTOMSLIDER_H

```

5.3 mainwindow.h

```

1 #ifndef MAINWINDOW_H
2 #define MAINWINDOW_H
3
4 #include <QMainWindow>
5
6 namespace Ui {
7 class MainWindow;
8 }
9
10 class MainWindow : public QMainWindow
11 {
12     Q_OBJECT

```

```

13
14 public:
15     explicit MainWindow(QWidget *parent = nullptr);
16     ~MainWindow();
17
18 private:
19     Ui::MainWindow *ui;
20     void initModelControls();
21
22     void initLights();
23
24 public slots:
25     void keyPressEvent(QKeyEvent *ke);
26
27     void onBtnLoadModelClicked();
28
29 private slots:
30     void hideControls();
31
32     void updateTabContent();
33 };
34 #endif // MAINWINDOW_H

```

5.4 modelobj.h

```

1 #ifndef MODEL_OBJ_H
2 #define MODEL_OBJ_H
3 #include <QtCore>
4 #include <QFile>
5 #include "BaseModel.h"
6
7 class ModelObj : public BaseModel
8 {
9 public:
10     ModelObj();
11
12     void LoadMdl() override; //Load model file
13
14 private:
15     void ParseVertexData(QByteArray line);
16
17     void ParseFacesData(QByteArray line);
18 };
19 #endif // MODEL_OBJ_H

```

5.5 modelstl.h

```

1 #ifndef MODEL_STL_H
2 #define MODEL_STL_H
3 #include <QtCore>
4 #include <QFile>
5 #include "BaseModel.h"
6
7 class ModelStl : public BaseModel
8 {
9 public:
10     ModelStl();
11
12     void LoadMdl() override; //Load model file
13
14 private:
15     void readBinary();
16
17     void readText();
18
19     vertex coordinateFromString(const QString &line);
20 };
21 #endif // MODEL_STL_H

```

5.6 openglview.h

```

1 #ifndef GLWIN_H

```

```

2 #define GLWIN_H
3
4 #include <QtGui>
5 #include <QGLWidget>
6 #include <QtOpenGL>
7 #include <QTimer>
8 #include <math.h>
9 #include <vector>
10 #include "BaseModel.h"
11
12 typedef struct
13 {
14     GLfloat posX,posY,posZ;
15     GLfloat rotX,rotY,rotZ;
16     QPoint lastPos;
17 }
18 Camera_t;
19
20 class OpenGLView : public QGLWidget
21 {
22     Q_OBJECT
23
24 public:
25     OpenGLView(QWidget *parent = 0);
26     ~OpenGLView();
27
28     void keyPressEvent(QKeyEvent *ke) override;
29
30     void paintModelFromFile(const QString filePath);
31
32 public slots:
33
34     void setDirectionalLightMode(const bool isDirectional);
35
36     inline bool directionalLightMode() const { return m_isDirectionalLight; }
37
38     void setXRotation(int xAngle);
39
40     inline int modelXRotation() const { return qRound(m_xRot); }
41
42     void setYRotation(int yAngle);
43
44     inline int modelYRotation() const { return qRound(m_yRot); }
45
46     void setZRotation(int zAngle);
47
48     inline int modelZRotation() const { return qRound(m_zRot); }
49
50     void setScale(int scaleFactor);
51
52     int scaleFactor() const {return static_cast<int>(scale * 10.0f);}
53
54     void setLightIntensity(int intensity);
55
56     void setLightXPosition(int xPosition);
57
58     void setLightYPosition(int yPosition);
59
60     void setLightZPosition(int zPosition);
61
62     void setAmbientLightR(double value);
63
64     void setAmbientLightG(double value);
65
66     void setAmbientLightB(double value);
67
68     void setDiffuseLightR(double value);
69
70     void setDiffuseLightG(double value);
71
72     void setDiffuseLightB(double value);
73
74     void setSpecularLightR(double value);
75
76     void setSpecularLightG(double value);
77
78     void setSpecularLightB(double value);
79
80     void setCameraXPosition(int xPosition);
81
82     void setCameraYPosition(int yPosition);
83
84     void setCameraZPosition(int zPosition);
85
86     void setModelXPosition(int xPosition);
87
88     inline int modelXPosition() const {return static_cast<int>(xTra);}

```

```

209
214     void setModelYPosition(int yPosition);
215
220     inline int modelYPosition() const {return static_cast<int>(yTra);}
221
226     void setModelZPosition(int zPosition);
227
232     inline int modelZPosition() const {return static_cast<int>(zTra);}
233
234 private:
240     void drawFloorGrid(float size, float step);
241
246     void drawAxis(GLfloat axisLength);
247
248 private slots:
253     void wheelEvent(QWheelEvent *event) override;
254 private:
255
259     void initializeGL() override;
265     void resizeGL(int w, int h) override;
266
270     void paintGL() override;
271
279     void setFrustum(GLdouble fovY, GLdouble aspect, GLdouble zNear, GLdouble zFar);
280
281 private:
286     GLuint paintMdl();
287
291     void lights();
292
293 private:
294     BaseModel *model;
295     vtxArray vtxArr;
296     QTimer *timer;
297     ushort iRot;
298     float m_xRot;
299     float m_yRot;
300     float m_zRot;
301     GLuint idx;
302     GLfloat xTra, yTra, zTra;
303     GLuint fov;
304     GLdouble zN;
305     GLdouble zF;
306     GLboolean shadingMode;
307     GLfloat LightAmbient[4];
308     GLfloat LightDiffuse[4];
309     GLfloat LightSpecular[4];
310     GLfloat LightPosition[4];
311     GLfloat CameraPosition[3];
312     GLfloat m_lightIntensity;
313     float m_lastDiffuseR = 0.0f ;
314     float m_lastDiffuseG = 0.0f;
315     float m_lastDiffuseB = 0.0f;
316     bool m_isDirectionalLight;
317     bool wireFrameMode;
318     float scale, s0;
319     GLuint m_gridLenght;
320     GLuint vao;
321     Camera_t m_camera;
323 signals:
324     void modelXRotationChanged();
325     void modelYRotationChanged();
326     void modelZRotationChanged();
327     void modelXPositionChanged();
328     void modelYPositionChanged();
329     void modelZPositionChanged();
330     void scaleChanged();
331 };
332
333 #endif // GLWIN_H

```

