Virtual Concierge Creator and Management System

V1.0

Generated by Doxygen 1.8.10

Wed Aug 12 2015 00:19:11

Contents

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 Ui Namespace Reference

Chapter 6

Class Documentation

6.1 DrawGL Class Reference

```
#include <drawgl.h>
```

Public Member Functions

- DrawGL ()
- ∼DrawGL ()

Static Public Member Functions

- static void DrawModel (ModelMesh ∗, QMatrix4x4, QMatrix4x4, QMatrix4x4, QOpenGLTexture ∗, QVector3D, QVector2D, QOpenGLShaderProgram ∗, QMatrix4x4, float)
- static void DrawLine (QVector3D, QVector3D, QMatrix4x4, QMatrix4x4, QMatrix4x4, QVector3D, QOpenG←LShaderProgram ∗, QMatrix4x4, float)
- static void ShaderDraw (ModelMesh ∗, QOpenGLShaderProgram ∗)
- static void UpdateShaders (QMatrix4x4, QMatrix4x4, QMatrix4x4, QOpenGLTexture ∗, QVector3D, Q←Vector2D, QOpenGLShaderProgram ∗, QMatrix4x4, float)
- static void UpdateShaders (QMatrix4x4, QMatrix4x4, QMatrix4x4, QVector3D, QVector2D, QOpenGL←ShaderProgram ∗, QMatrix4x4, float)
- static void draw_if_true (ModelMesh ∗, QMatrix4x4, QVector3D, QVector3D, QVector3D, QOpenGLTexture ∗, QVector3D, QVector2D, QMatrix4x4, QOpenGLShaderProgram ∗, bool, float)

6.1.1 Constructor & Destructor Documentation

6.1.1.1 DrawGL::DrawGL ( )

6.1.1.2 DrawGL::∼DrawGL ( )

6.1.2 Member Function Documentation

6.1.2.1 void DrawGL::draw_if_true (**ModelMesh** ∗ *model*, QMatrix4x4 *view*, QVector3D *position*, QVector3D *rotation*, QVector3D *scaling*, QOpenGLTexture ∗ *texture*, QVector3D *color*, QVector2D *texturecoord*, QMatrix4x4 *pmatrix*, QOpenGLShaderProgram ∗ *shader_program*, bool *value*, float *height*) [static]

Here is the call graph for this function:

6.1.2.2 void DrawGL::DrawLine (QVector3D *point1,* QVector3D *point2,* QMatrix4x4 *wvp,* QMatrix4x4 *mvp,* QMatrix4x4 *rotate,* QVector3D *color,* QOpenGLShaderProgram ∗ *shader_program,* QMatrix4x4 *pmatrix,* float *height*) `[static]`

Here is the call graph for this function:

Here is the caller graph for this function:

6.1.2.3 void DrawGL::DrawModel (ModelMesh ∗ *box,* QMatrix4x4 *wvp,* QMatrix4x4 *mvp,* QMatrix4x4 *rotate,* QOpenGLTexture ∗ *texture,* QVector3D *color,* QVector2D *texturecoordmulti,* QOpenGLShaderProgram ∗ *shader_program,* QMatrix4x4 *pmatrix,* float *height*) `[static]`

Here is the call graph for this function:

Here is the caller graph for this function:

6.1.2.4 void DrawGL::ShaderDraw (ModelMesh ∗ *box,* QOpenGLShaderProgram ∗ *shader_program*) `[static]`

Here is the call graph for this function:

Here is the caller graph for this function:

6.1.2.5 void DrawGL::UpdateShaders (QMatrix4x4 *wvp,* QMatrix4x4 *mvp,* QMatrix4x4 *rotate,* QOpenGLTexture ∗ *texture,* QVector3D *color,* QVector2D *texturecooredinates,* QOpenGLShaderProgram ∗ *shader_program,* QMatrix4x4 *pmatrix,* float *height*) `[static]`

Here is the caller graph for this function:

6.1.2.6 void DrawGL::UpdateShaders (QMatrix4x4 *wvp,* QMatrix4x4 *mvp,* QMatrix4x4 *rotate,* QVector3D *color,* QVector2D *texturecooredinates,* QOpenGLShaderProgram ∗ *shader_program,* QMatrix4x4 *pmatrix,* float *height*) `[static]`

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

- Functions/drawgl.h
- Functions/drawgl.cpp

6.2 MainWindow Class Reference

`#include <mainwindow.h>`

Inheritance diagram for MainWindow:

Collaboration diagram for MainWindow:

Signals

- void place_node (bool placable)
- void node_links (bool linkable)
- void remove_nodes (bool removable)
- void remove_trees (bool removable)
- void place_pavement (bool placable)
- void place_wall (bool placable)
- void place_door (bool placable)
- void change_rotationY (double angle)
- void place_tree (bool placable)

- void invert_mouseY (bool invert_mouse)
- void place_floor_plan (bool placable)
- void add_new_texture (QString file_name)
- void set_object_scale (QVector3D scale)
- void change_floor_selected (float height)
- void change_node_name (QString name)
- void set_node_significant (bool add_button)
- void load_premises (QString file_name)

Public Member Functions

- MainWindow (QWidget ∗parent=0)
- ∼MainWindow ()

6.2.1 Constructor & Destructor Documentation

6.2.1.1 MainWindow::MainWindow (QWidget ∗ *parent* = 0) [explicit]

6.2.1.2 MainWindow::∼MainWindow ( )

6.2.2 Member Function Documentation

6.2.2.1 void MainWindow::add_new_texture (QString *file_name*) [signal]

Here is the caller graph for this function:

6.2.2.2 void MainWindow::change_floor_selected (float *height*) [signal]

Here is the caller graph for this function:

6.2.2.3 void MainWindow::change_node_name (QString *name*) [signal]

Here is the caller graph for this function:

6.2.2.4 void MainWindow::change_rotationY (double *angle*) [signal]

Here is the caller graph for this function:

6.2.2.5 void MainWindow::invert_mouseY (bool *invert_mouse*) [signal]

Here is the caller graph for this function:

6.2.2.6 void MainWindow::load_premises (QString *file_name*) [signal]

Here is the caller graph for this function:

6.2.2.7 void MainWindow::node_links (bool *linkable*) [signal]

Here is the caller graph for this function:

6.2.2.8 void MainWindow::place_door (bool *placable*) [signal]

Here is the caller graph for this function:

6.2.2.9 void MainWindow::place_floor_plan (bool *placable*) [signal]

Here is the caller graph for this function:

6.2.2.10 void MainWindow::place_node (bool *placable*) [signal]

Here is the caller graph for this function:

6.2.2.11 void MainWindow::place_pavement (bool *placable*) [signal]

Here is the caller graph for this function:

6.2.2.12 void MainWindow::place_tree (bool *placable*) [signal]

Here is the caller graph for this function:

6.2.2.13 void MainWindow::place_wall (bool *placable*) [signal]

Here is the caller graph for this function:

6.2.2.14 void MainWindow::remove_nodes (bool *removable*) [signal]

Here is the caller graph for this function:

6.2.2.15 void MainWindow::remove_trees (bool *removable*) [signal]

Here is the caller graph for this function:

6.2.2.16 void MainWindow::set_node_significant (bool *add_button*) [signal]

Here is the caller graph for this function:

6.2.2.17 void MainWindow::set_object_scale (QVector3D *scale*) [signal]

Here is the caller graph for this function:

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

- mainwindow.h
- mainwindow.cpp

6.3 Mathematics Class Reference

```
#include <mathematics.h>
```

Public Member Functions

- Mathematics ()
- ∼Mathematics ()

Static Public Member Functions

- static QVector3D intersectYnull (QVector3D raycast_vector, QVector3D camera_position)
- static QVector3D intersectYat (QVector3D raycast_vector, QVector3D camera_position, float y_intersect_↩ plane)
- static float flat_angle_from_vectors (QVector3D vector_a, QVector3D vector_b)
- static float return_near_degree (float degrees)
- static QVector3D mouse_raycast (int mouse_x, int mouse_y, int screen_width, int screen_height, float invert↩ _sign, QMatrix4x4 view_matrix, QMatrix4x4 projection_matrix)
- static QVector3D point_on_line (float x, QVector3D point_a, QVector3D point_b)
- static QPoint transform_3d_to_2d (QMatrix4x4 view_matrix, QMatrix4x4 projection_matrix, QVector3D point, int screen_width, int screen_height)

6.3.1 Constructor & Destructor Documentation

6.3.1.1 Mathematics::Mathematics ( )

6.3.1.2 Mathematics::∼Mathematics ( )

6.3.2 Member Function Documentation

6.3.2.1 float Mathematics::flat_angle_from_vectors (QVector3D *vector_a,* QVector3D *vector_b*) [static]

6.3.2.2 QVector3D Mathematics::intersectYat (QVector3D *raycast_vector,* QVector3D *camera_position,* float *y_intersect_plane*) [static]

Here is the caller graph for this function:

6.3.2.3 QVector3D Mathematics::intersectYnull (QVector3D *raycast_vector,* QVector3D *camera_position*) [static]

6.3.2.4 QVector3D Mathematics::mouse_raycast (int *mouse_x,* int *mouse_y,* int *screen_width,* int *screen_height,* float *invert_sign,* QMatrix4x4 *view_matrix,* QMatrix4x4 *projection_matrix*) [static]

Here is the caller graph for this function:

6.3.2.5 QVector3D Mathematics::point_on_line (float *x,* QVector3D *point_a,* QVector3D *point_b*) [static]

6.3.2.6 float Mathematics::return_near_degree (float *degrees*) [static]

6.3.2.7 QPoint Mathematics::transform_3d_to_2d (QMatrix4x4 *view_matrix,* QMatrix4x4 *projection_matrix,* QVector3D *point,* int *screen_width,* int *screen_height*) [static]

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

- Functions/mathematics.h
- Functions/mathematics.cpp

6.4 ModelMesh Class Reference

```
#include <ModelMesh.h>
```

Public Member Functions

- ModelMesh (QString filename)
- ∼ModelMesh ()
- void Draw ()
- bool LoadOBJ (QString filename)

Public Attributes

- QVector< QVector2D > textureCoordinates
- QVector< QVector3D > vertices
- QVector< QVector3D > normals
- QVector< int > vertexIndices
- QVector< int > uvIndices
- QVector< int > normalIndices

6.4.1 Constructor & Destructor Documentation

6.4.1.1 ModelMesh::ModelMesh (QString *filename*) [explicit]

Here is the call graph for this function:

6.4.1.2 ModelMesh::∼ModelMesh ( )

6.4.2 Member Function Documentation

6.4.2.1 void ModelMesh::Draw ( )

Here is the caller graph for this function:

6.4.2.2 bool ModelMesh::LoadOBJ (QString *filename*)

Here is the caller graph for this function:

6.4.3 Member Data Documentation

6.4.3.1 QVector<int> ModelMesh::normalIndices

6.4.3.2 QVector<QVector3D> ModelMesh::normals

6.4.3.3 QVector<QVector2D> ModelMesh::textureCoordinates

6.4.3.4 QVector<int> ModelMesh::uvIndices

6.4.3.5 QVector<int> ModelMesh::vertexIndices

6.4.3.6 QVector<QVector3D> ModelMesh::vertices

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

- Objects/ModelMesh.h
- Objects/ModelMesh.cpp

6.5 Node Class Reference

```
#include <Node.h>
```

Public Member Functions

- Node ()
- Node (QVector3D ∗position)
- Node (QVector3D ∗position, QString ∗name)
- void AddLink (QString ∗name, int index)
- QVector3D Position ()
- QVector3D getColor ()
- void setSourceNode ()
- void setDestinationNode ()
- void setColor (QVector3D ∗rgb)
- int countConnected ()
- int getConnectedIndex (int index)
- void RemoveLinkedFromIndex (int index)
- void MoveLinkedIndexBack (int index)
- QString getLinkedName (int index)
- void clearPath ()
- void addShortest (int index)
- void setG (double g)
- void setName (QString)
- void setShortest (int index)
- void setWalk (bool can_walk)
- void setWheelChair (bool can_use_wheelchair)
- void setVehicle (bool can_use_vehicle)
- void setBike (bool can_use_bike)
- void setSignificant (bool is_significant)
- bool getWalk ()
- bool getWheelChair ()
- bool getVehicle ()
- bool getBike ()
- bool getSignificant ()
- int getShortestIndex ()
- double getG ()
- QString getName ()

6.5.1 Constructor & Destructor Documentation

6.5.1.1 Node::Node ( )

6.5.1.2 Node::Node (QVector3D ∗ *position*) [explicit]

Here is the call graph for this function:

6.5.1.3 Node::Node (QVector3D ∗ *position,* QString ∗ *name*)

Here is the call graph for this function:

6.5.2 Member Function Documentation

6.5.2.1 void Node::AddLink (QString ∗ *name,* int *index*)

6.5.2.2 void Node::addShortest (int *index*)

6.5.2.3 void Node::clearPath ( )

6.5.2.4 int Node::countConnected ( )

6.5.2.5 bool Node::getBike ( )

6.5.2.6 QVector3D Node::getColor ( )

6.5.2.7 int Node::getConnectedIndex (int *index*)

6.5.2.8 double Node::getG ( )

6.5.2.9 QString Node::getLinkedName (int *index*)

6.5.2.10 QString Node::getName ( )

6.5.2.11 int Node::getShortestIndex ( )

6.5.2.12 bool Node::getSignificant ( )

6.5.2.13 bool Node::getVehicle ( )

6.5.2.14 bool Node::getWalk ( )

6.5.2.15 bool Node::getWheelChair ( )

6.5.2.16 void Node::MoveLinkedIndexBack (int *index*)

6.5.2.17 QVector3D Node::Position ( )

Here is the caller graph for this function:

6.5.2.18 void Node::RemoveLinkedFromIndex (int *index*)

6.5.2.19 void Node::setBike (bool *can_use_bike*)

6.5.2.20 void Node::setColor (QVector3D ∗ *rgb*)

6.5.2.21 void Node::setDestinationNode ( )

6.5.2.22 void Node::setG (double *g*)

Here is the caller graph for this function:

6.5.2.23 void Node::setName (QString *value*)

6.5.2.24 void Node::setShortest (int *index*)

6.5.2.25 void Node::setSignificant (bool *is_significant*)

6.5.2.26 void Node::setSourceNode ( )

6.5.2.27 void Node::setVehicle (bool *can_use_vehicle*)

6.5.2.28 void Node::setWalk (bool *can_walk*)

6.5.2.29 void Node::setWheelChair (bool *can_use_wheelchair*)

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

- Objects/Node.h
- Objects/Node.cpp

6.6 NodeButton Class Reference

`#include <nodebutton.h>`

Inheritance diagram for NodeButton:

Collaboration diagram for NodeButton:

Signals

- void clicked_index (int, bool)

Public Member Functions

- NodeButton (QWidget ∗parent=0)
- ∼NodeButton ()
- int getIndex ()
- void setIndex (int index)
- void setDirectory (bool is_directory)
- bool isDirectory ()
- void mousePressEvent (QMouseEvent ∗event)

6.6.1 Constructor & Destructor Documentation

6.6.1.1 NodeButton::NodeButton (QWidget ∗ *parent* = 0) [explicit]

6.6.1.2 NodeButton::∼NodeButton ( )

6.6.2 Member Function Documentation

6.6.2.1 void NodeButton::clicked_index (int , bool) [signal]

Here is the caller graph for this function:

6.6.2.2 int NodeButton::getIndex ( )

6.6.2.3 bool NodeButton::isDirectory ( )

6.6.2.4 void NodeButton::mousePressEvent (QMouseEvent ∗ *event*)

6.6.2.5 void NodeButton::setDirectory (bool *is_directory*)

6.6.2.6 void NodeButton::setIndex (int *index*)

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

- Objects/nodebutton.h
- Objects/nodebutton.cpp

6.7 NodeHandler Class Reference

```
#include <NodeHandler.h>
```

Public Member Functions

- NodeHandler ()
- void AddNode (Node ∗node)
- Node NodeFromIndex (unsigned int index)
- void AddNodeLink (int index, QString ∗name)
- void AddNodeLinkbyIndex (int index1, int index2)
- void CalculateShortest (int start, int finish)
- void ReadFilePVC (QString filename)
- int count ()
- int pathcount ()
- int pathindex (int shortest_index)

6.7.1 Constructor & Destructor Documentation

6.7.1.1 NodeHandler::NodeHandler ( )

6.7.2 Member Function Documentation

6.7.2.1 void NodeHandler::AddNode (Node ∗ *node*)

Here is the caller graph for this function:

6.7.2.2 void NodeHandler::AddNodeLink (int *index,* QString ∗ *name*)

6.7.2.3 void NodeHandler::AddNodeLinkbyIndex (int *index1,* int *index2*)

Here is the caller graph for this function:

6.7.2.4 void NodeHandler::CalculateShortest (int *start,* int *finish*)

Here is the call graph for this function:

6.7.2.5 int NodeHandler::count ( )

6.7.2.6 Node NodeHandler::NodeFromIndex (unsigned int *index*)

6.7.2.7 int NodeHandler::pathcount ( )

Here is the caller graph for this function:

6.7.2.8 int NodeHandler::pathindex (int *shortest_index*)

Here is the caller graph for this function:

6.7.2.9 void NodeHandler::ReadFilePVC (QString *filename*)

Here is the call graph for this function:

Here is the caller graph for this function:

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

- Objects/NodeHandler.h
- Objects/NodeHandler.cpp

6.8 PremisesExporter Class Reference

```
#include <premises_exporter.h>
```

Public Member Functions

- PremisesExporter ()

Static Public Member Functions

- static void export_environment (QVector< VisualObject $*$ > object_list, QString filename)
- static void export_nodes (QVector< Node $*$ > node_list, QString filename)
- static void export_texture (QVector< QString > texture_paths, QString filename)
- static void export_directories (QVector< QString > directory, QVector< QString > directory_list, QVector< QString > startup_menu, QString filename)
- static bool fileExists (QString filename)

6.8.1 Constructor & Destructor Documentation

6.8.1.1 PremisesExporter::PremisesExporter ( )

6.8.2 Member Function Documentation

6.8.2.1 void PremisesExporter::export_directories (QVector< QString > *directory,* QVector< QString > *directory_list,* QVector< QString > *startup_menu,* QString *filename*) `[static]`

6.8.2.2 void PremisesExporter::export_environment (QVector< VisualObject $*$ > *object_list,* QString *filename*) `[static]`

6.8.2.3 void PremisesExporter::export_nodes (QVector< Node ∗ > *node_list,* QString *filename*) [static]

Here is the caller graph for this function:

6.8.2.4 void PremisesExporter::export_texture (QVector< QString > *texture_paths,* QString *filename*) [static]

6.8.2.5 bool PremisesExporter::fileExists (QString *filename*) [static]

Here is the caller graph for this function:

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

- Functions/premises_exporter.h
- Functions/premises_exporter.cpp

6.9 RenderState Class Reference

`#include <renderstate.h>`

Inheritance diagram for RenderState:

Collaboration diagram for RenderState:

Signals

- void opengl_initialised (bool)

Public Member Functions

- RenderState (QWidget ∗parent=0)
- ∼RenderState ()

Protected Member Functions

- void initializeGL ()
- void resizeGL (int width, int height)
- void paintGL ()
- void mouseMoveEvent (QMouseEvent ∗event)
- void mousePressEvent (QMouseEvent ∗event)
- void wheelEvent (QWheelEvent ∗event)
- void mouseReleaseEvent (QMouseEvent ∗event)

6.9.1 Constructor & Destructor Documentation

6.9.1.1 RenderState::RenderState (QWidget ∗ *parent* = 0) [explicit]

6.9.1.2 RenderState::∼RenderState ( )

6.9.2 Member Function Documentation

6.9.2.1 void RenderState::initializeGL ( ) [protected]

6.9.2.2 void RenderState::mouseMoveEvent (QMouseEvent ∗ *event*) [protected]

Here is the call graph for this function:

6.9.2.3 void RenderState::mousePressEvent (QMouseEvent ∗ *event*) [protected]

Here is the call graph for this function:

6.9.2.4 void RenderState::mouseReleaseEvent (QMouseEvent ∗ *event*) [protected]

Here is the call graph for this function:

6.9.2.5 void RenderState::opengl_initialised (bool) [signal]

Here is the caller graph for this function:

6.9.2.6 void RenderState::paintGL ( ) [protected]

Here is the call graph for this function:

6.9.2.7 void RenderState::resizeGL (int *width,* int *height*) [protected]

6.9.2.8 void RenderState::wheelEvent (QWheelEvent ∗ *event*) [protected]

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

- RenderStates/renderstate.h
- RenderStates/renderstate.cpp

6.10 Smtp Class Reference

```
#include <smtp.h>
```

Inheritance diagram for Smtp:

Collaboration diagram for Smtp:

Signals

- void status (const QString &)

Public Member Functions

- Smtp (const QString &user, const QString &pass, const QString &host, int port=465, int timeout=30000)
- ∼Smtp ()
- void sendMail (const QString &from, const QString &to, const QString &subject, const QString &body, QStringList files=QStringList())

6.10.1 Constructor & Destructor Documentation

6.10.1.1 Smtp::Smtp (const QString & *user*, const QString & *pass*, const QString & *host*, int *port* = 465, int *timeout* = 30000)

6.10.1.2 Smtp::∼Smtp ( )

6.10.2 Member Function Documentation

6.10.2.1 void Smtp::sendMail (const QString & *from*, const QString & *to*, const QString & *subject*, const QString & *body*, QStringList *files* = QStringList())

6.10.2.2 void Smtp::status (const QString &) [signal]

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

- SMTP/smtp.h
- SMTP/smtp.cpp

6.11 UserInterfaceCreator Class Reference

`#include <userinterfacecreator.h>`

Inheritance diagram for UserInterfaceCreator:

Collaboration diagram for UserInterfaceCreator:

Public Member Functions

- UserInterfaceCreator (QWidget ∗parent=0)
- ∼UserInterfaceCreator ()

6.11.1 Constructor & Destructor Documentation

6.11.1.1 UserInterfaceCreator::UserInterfaceCreator (QWidget ∗ *parent* = 0) [explicit]

6.11.1.2 UserInterfaceCreator::∼UserInterfaceCreator ( )

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

- userinterfacecreator.h
- userinterfacecreator.cpp

6.12 VirtualConcierge Class Reference

`#include <virtualconcierge.h>`

Inheritance diagram for VirtualConcierge:

Collaboration diagram for VirtualConcierge:

Signals

- void find_path (int start, int goal)

Public Member Functions

- VirtualConcierge (QWidget ∗parent=0)
- ∼VirtualConcierge ()

6.12.1 Constructor & Destructor Documentation

6.12.1.1 VirtualConcierge::VirtualConcierge (QWidget ∗ *parent* = 0) [explicit]

6.12.1.2 VirtualConcierge::∼VirtualConcierge ( )

6.12.2 Member Function Documentation

6.12.2.1 void VirtualConcierge::find_path (int *start*, int *goal*) [signal]

Here is the caller graph for this function:

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

- virtualconcierge.h
- virtualconcierge.cpp

6.13 VirtualConciergeRenderstate Class Reference

```
#include <virtualconciergerenderstate.h>
```

Inheritance diagram for VirtualConciergeRenderstate:

Collaboration diagram for VirtualConciergeRenderstate:

Public Member Functions

- VirtualConciergeRenderstate (QWidget ∗parent=0)
- ∼VirtualConciergeRenderstate ()

Protected Member Functions

- void initializeGL ()
- void paintGL ()
- void resizeGL (int w, int h)

6.13.1 Constructor & Destructor Documentation

6.13.1.1 VirtualConciergeRenderstate::VirtualConciergeRenderstate (QWidget ∗ *parent* = 0) [explicit]

Here is the call graph for this function:

6.13.1.2 VirtualConciergeRenderstate::∼VirtualConciergeRenderstate ( )

6.13.2 Member Function Documentation

6.13.2.1 void VirtualConciergeRenderstate::initializeGL ( ) [protected]

6.13.2.2 void VirtualConciergeRenderstate::paintGL ( ) [protected]

Here is the call graph for this function:

6.13.2.3 void VirtualConciergeRenderstate::resizeGL (int *w,* int *h*) [protected]

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

- RenderStates/virtualconciergerenderstate.h
- RenderStates/virtualconciergerenderstate.cpp

6.14 VisualObject Class Reference

`#include <visualobject.h>`

Public Member Functions

- VisualObject (ModelMesh ∗modelmesh, QOpenGLTexture ∗texture, QVector3D translation, QVector3D rotation, QString type)
- ∼VisualObject ()
- void setModel (ModelMesh ∗modelmesh)
- void setRotation (QVector3D rotation)
- void setTranslation (QVector3D translation)
- void setTexture (QOpenGLTexture ∗texture)
- void setScaling (QVector3D scale)
- void setCornerULeft (QVector3D position)
- void setCornerURight (QVector3D position)
- void setCornerLLeft (QVector3D position)
- void setCornerLRight (QVector3D position)
- void setUMidHorisontal (QVector3D position)
- void setLMidHorisontal (QVector3D position)
- void setTextureID (int index)
- void setTexturePath (QString filename)
- QVector3D getCornerULeft ()
- QVector3D getCornerURight ()
- QVector3D getCornerLLeft ()
- QVector3D getCornerLRight ()
- QVector3D getUMidHorisontal ()
- QVector3D getLMidHorisontal ()
- QVector3D getRotation ()
- QVector3D getTranslation ()
- QVector3D getScaling ()
- ModelMesh ∗ getModelMesh ()
- QOpenGLTexture ∗ getTexture ()
- QString getType ()
- QString getTexturePath ()
- int getTextureID ()
- void setType (QString type)

6.14.1 Constructor & Destructor Documentation

6.14.1.1 VisualObject::VisualObject (**ModelMesh** ∗ *modelmesh,* QOpenGLTexture ∗ *texture,* QVector3D *translation,* QVector3D *rotation,* QString *type*)

6.14.1.2 VisualObject::∼VisualObject ( )

6.14.2 Member Function Documentation

6.14.2.1 QVector3D VisualObject::getCornerLLeft ( )

6.14.2.2 QVector3D VisualObject::getCornerLRight ( )

6.14.2.3 QVector3D VisualObject::getCornerULeft ( )

6.14.2.4 QVector3D VisualObject::getCornerURight ( )

6.14.2.5 QVector3D VisualObject::getLMidHorisontal ( )

Here is the caller graph for this function:

6.14.2.6 **ModelMesh** ∗ VisualObject::getModelMesh ( )

Here is the caller graph for this function:

6.14.2.7 QVector3D VisualObject::getRotation ( )

Here is the caller graph for this function:

6.14.2.8 QVector3D VisualObject::getScaling ( )

Here is the caller graph for this function:

6.14.2.9 QOpenGLTexture ∗ VisualObject::getTexture ( )

6.14.2.10 int VisualObject::getTextureID ( )

6.14.2.11 QString VisualObject::getTexturePath ( )

6.14.2.12 QVector3D VisualObject::getTranslation ( )

Here is the caller graph for this function:

6.14.2.13 QString VisualObject::getType ( )

Here is the caller graph for this function:

6.14.2.14 QVector3D VisualObject::getUMidHorisontal ( )

Here is the caller graph for this function:

6.14.2.15 void VisualObject::setCornerLLeft (QVector3D *position*)

6.14.2.16 void VisualObject::setCornerLRight (QVector3D *position*)

6.14.2.17 void VisualObject::setCornerULeft (QVector3D *position*)

6.14.2.18 void VisualObject::setCornerURight (QVector3D *position*)

6.14.2.19 void VisualObject::setLMidHorisontal (QVector3D *position*)

6.14.2.20 void VisualObject::setModel (ModelMesh ∗ *modelmesh*)

6.14.2.21 void VisualObject::setRotation (QVector3D *rotation*)

6.14.2.22 void VisualObject::setScaling (QVector3D *scale*)

6.14.2.23 void VisualObject::setTexture (QOpenGLTexture ∗ *texture*)

6.14.2.24 void VisualObject::setTextureID (int *index*)

6.14.2.25 void VisualObject::setTexturePath (QString *filename*)

6.14.2.26 void VisualObject::setTranslation (QVector3D *translation*)

6.14.2.27 void VisualObject::setType (QString *type*)

6.14.2.28 void VisualObject::setUMidHorisontal (QVector3D *position*)

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

- Objects/visualobject.h
- Objects/visualobject.cpp

Chapter 7

File Documentation

7.1 Functions/drawgl.cpp File Reference

```
#include "./drawgl.h"
```

Include dependency graph for drawgl.cpp:

7.2 Functions/drawgl.h File Reference

```
#include <QOpenGLWidget>
#include <QOpenGLFunctions>
#include <QOpenGLTexture>
#include <QMatrix4x4>
#include <QOpenGLShaderProgram>
#include <QMouseEvent>
#include <QVector3D>
#include <QtMath>
#include "Objects/ModelMesh.h"
```

Include dependency graph for drawgl.h: This graph shows which files directly or indirectly include this file:

Classes

- class DrawGL

7.3 Functions/mathematics.cpp File Reference

```
#include <QtMath>
#include "./mathematics.h"
```

Include dependency graph for mathematics.cpp:

7.4 Functions/mathematics.h File Reference

```
#include <QVector3D>
#include <QVector4D>
#include <QPoint>
#include <QMatrix4x4>
```

Include dependency graph for mathematics.h: This graph shows which files directly or indirectly include this file:

Classes

- class Mathematics

7.5 Functions/premises_exporter.cpp File Reference

```
#include <QFile>
#include <QTextStream>
#include <QDir>
#include <QDebug>
#include "./premises_exporter.h"
```

Include dependency graph for premises_exporter.cpp:

7.6 Functions/premises_exporter.h File Reference

```
#include "Objects/visualobject.h"
#include "Objects/Node.h"
```

Include dependency graph for premises_exporter.h: This graph shows which files directly or indirectly include this file:

Classes

- class PremisesExporter

7.7 main.cpp File Reference

```
#include "./main.h"
```

Include dependency graph for main.cpp:

Functions

- int main (int argc, char ∗argv[ ])

7.7.1 Function Documentation

7.7.1.1 int main (int *argc,* char ∗ *argv[ ]*)

7.8 main.h File Reference

```
#include <QApplication>
#include "./mainwindow.h"
#include "Functions/premises_exporter.h"
```

Include dependency graph for main.h: This graph shows which files directly or indirectly include this file:

7.9 mainwindow.cpp File Reference

```
#include "./mainwindow.h"
#include "./ui_mainwindow.h"
#include "./userinterfacecreator.h"
```

Include dependency graph for mainwindow.cpp:

7.10 mainwindow.h File Reference

```
#include <QMainWindow>
#include <QFileDialog>
#include "./virtualconcierge.h"
#include "Functions/premises_exporter.h"
```

Include dependency graph for mainwindow.h: This graph shows which files directly or indirectly include this file:

Classes

- class MainWindow

Namespaces

- Ui

7.11 Objects/ModelMesh.cpp File Reference

```
#include <QFile>
#include <QTextStream>
#include <QOpenGLFunctions>
#include <QVector2D>
#include <QVector3D>
#include <QString>
#include "Objects/ModelMesh.h"
```

Include dependency graph for ModelMesh.cpp:

7.12 Objects/ModelMesh.h File Reference

```
#include <QVector>
```

Include dependency graph for ModelMesh.h: This graph shows which files directly or indirectly include this file:

Classes

- class ModelMesh

7.13 Objects/Node.cpp File Reference

```
#include "Objects/Node.h"
#include <QVector2D>
#include <QVector3D>
#include <QString>
```

Include dependency graph for Node.cpp:

7.14 Objects/Node.h File Reference

```
#include <QVector>
```

Include dependency graph for Node.h: This graph shows which files directly or indirectly include this file:

Classes

- class Node

7.15 Objects/nodebutton.cpp File Reference

```
#include "./nodebutton.h"
```

Include dependency graph for nodebutton.cpp:

7.16 Objects/nodebutton.h File Reference

```
#include <QPushButton>
#include <QWidget>
```

Include dependency graph for nodebutton.h: This graph shows which files directly or indirectly include this file:

Classes

- class NodeButton

7.17 Objects/NodeHandler.cpp File Reference

```
#include "Objects/NodeHandler.h"
#include <QVector3D>
#include <qdebug.h>
#include <QFile>
#include <QTextStream>
```

Include dependency graph for NodeHandler.cpp:

7.18 Objects/NodeHandler.h File Reference

```
#include <QVector>
#include <QString>
#include <qdebug.h>
#include "Objects/Node.h"
```

Include dependency graph for NodeHandler.h: This graph shows which files directly or indirectly include this file:

Classes

- class NodeHandler

7.19 Objects/visualobject.cpp File Reference

```
#include "Objects/visualobject.h"
```

Include dependency graph for visualobject.cpp:

7.20 Objects/visualobject.h File Reference

```
#include <QVector3D>
#include <QMatrix>
#include <QOpenGLTexture>
#include <Objects/ModelMesh.h>
```

Include dependency graph for visualobject.h: This graph shows which files directly or indirectly include this file:

Classes

- class VisualObject

7.21 RenderStates/renderstate.cpp File Reference

```
#include <algorithm>
#include "./renderstate.h"
#include "Functions/mathematics.h"
```

Include dependency graph for renderstate.cpp:

7.22 RenderStates/renderstate.h File Reference

```
#include <QOpenGLWidget>
#include <QOpenGLFunctions>
#include <QOpenGLTexture>
#include <QMatrix4x4>
#include <QOpenGLShaderProgram>
#include <QMouseEvent>
#include <QVector3D>
#include <QTimer>
#include <QtMath>
#include <QLabel>
#include <QMessageBox>
#include <QPainter>
#include "Objects/ModelMesh.h"
#include "Objects/Node.h"
#include "Objects/visualobject.h"
#include "Functions/premises_exporter.h"
#include "Functions/drawgl.h"
```

Include dependency graph for renderstate.h: This graph shows which files directly or indirectly include this file:

Classes

- class RenderState

7.23 RenderStates/virtualconciergerenderstate.cpp File Reference

```
#include <QFileDialog>
#include <QPainter>
#include "./virtualconciergerenderstate.h"
```

Include dependency graph for virtualconciergerenderstate.cpp:

7.24 RenderStates/virtualconciergerenderstate.h File Reference

```
#include <QOpenGLWidget>
#include <QOpenGLFunctions>
#include <QOpenGLTexture>
#include <QOpenGLShaderProgram>
#include "Objects/ModelMesh.h"
#include "Objects/NodeHandler.h"
#include "Functions/drawgl.h"
#include "Objects/visualobject.h"
#include "Functions/premises_exporter.h"
```

Include dependency graph for virtualconciergerenderstate.h: This graph shows which files directly or indirectly include this file:

Classes

- class VirtualConciergeRenderstate

7.25 SMTP/smtp.cpp File Reference

```
#include <QNetworkProxy>
#include "SMTP/smtp.h"
```

Include dependency graph for smtp.cpp:

7.26 SMTP/smtp.h File Reference

```
#include <QtNetwork/QAbstractSocket>
#include <QtNetwork/QSslSocket>
#include <QString>
#include <QTextStream>
#include <QDebug>
#include <QtWidgets/QMessageBox>
#include <QByteArray>
#include <QFile>
#include <QFileInfo>
```

Include dependency graph for smtp.h: This graph shows which files directly or indirectly include this file:

Classes

- class Smtp

7.27 userinterfacecreator.cpp File Reference

```
#include <QDebug>
#include "./userinterfacecreator.h"
#include "./ui_userinterfacecreator.h"
#include "Functions/premises_exporter.h"
```

Include dependency graph for userinterfacecreator.cpp:

7.28 userinterfacecreator.h File Reference

```
#include <QDialog>
#include <QTreeWidget>
#include <QFile>
#include <QVector>
```

Include dependency graph for userinterfacecreator.h: This graph shows which files directly or indirectly include this file:

Classes

- class UserInterfaceCreator

Namespaces

- Ui

7.29 virtualconcierge.cpp File Reference

```
#include <QDebug>
#include <QFileDialog>
#include <QStringList>
#include <QSurfaceFormat>
#include "./virtualconcierge.h"
#include "./ui_virtualconcierge.h"
#include "SMTP/smtp.h"
```

Include dependency graph for virtualconcierge.cpp:

7.30 virtualconcierge.h File Reference

```
#include <QWidget>
#include <QVector>
#include <QPushButton>
#include "Objects/nodebutton.h"
#include "RenderStates/virtualconciergerenderstate.h"
```

Include dependency graph for virtualconcierge.h: This graph shows which files directly or indirectly include this file:

Classes

- class VirtualConcierge

Namespaces

- Ui

Index