

Virtual Concierge Creator and Management System

V1.0

Generated by Doxygen 1.8.10

Wed Aug 12 2015 00:19:11

Contents

1	Namespace Index	1
1.1	Namespace List	1
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	File Index	7
4.1	File List	7
5	Namespace Documentation	9
5.1	Ui Namespace Reference	9
6	Class Documentation	11
6.1	DrawGL Class Reference	11
6.1.1	Constructor & Destructor Documentation	11
6.1.1.1	DrawGL()	11
6.1.1.2	~DrawGL()	11
6.1.2	Member Function Documentation	11
6.1.2.1	draw_if_true(ModelMesh *, QMatrix4x4, QVector3D, QVector3D, QVector3D, QOpenGLTexture *, QVector3D, QVector2D, QMatrix4x4, QOpenGLShaderProgram *, bool, float)	11
6.1.2.2	DrawLine(QVector3D, QVector3D, QMatrix4x4, QMatrix4x4, QMatrix4x4, QVector3D, QOpenGLShaderProgram *, QMatrix4x4, float)	12
6.1.2.3	DrawModel(ModelMesh *, QMatrix4x4, QMatrix4x4, QMatrix4x4, QOpenGLTexture *, QVector3D, QVector2D, QOpenGLShaderProgram *, QMatrix4x4, float)	12
6.1.2.4	ShaderDraw(ModelMesh *, QOpenGLShaderProgram *)	12
6.1.2.5	UpdateShaders(QMatrix4x4, QMatrix4x4, QMatrix4x4, QOpenGLTexture *, QVector3D, QVector2D, QOpenGLShaderProgram *, QMatrix4x4, float)	12
6.1.2.6	UpdateShaders(QMatrix4x4, QMatrix4x4, QMatrix4x4, QVector3D, QVector2D, QOpenGLShaderProgram *, QMatrix4x4, float)	12
6.2	MainWindow Class Reference	12
6.2.1	Constructor & Destructor Documentation	13

6.2.1.1	MainWindow(QWidget *parent=0)	13
6.2.1.2	~MainWindow()	13
6.2.2	Member Function Documentation	13
6.2.2.1	add_new_texture	13
6.2.2.2	change_floor_selected	13
6.2.2.3	change_node_name	13
6.2.2.4	change_rotationY	13
6.2.2.5	invert_mouseY	13
6.2.2.6	load_premises	13
6.2.2.7	node_links	13
6.2.2.8	place_door	14
6.2.2.9	place_floor_plan	14
6.2.2.10	place_node	14
6.2.2.11	place_pavement	14
6.2.2.12	place_tree	14
6.2.2.13	place_wall	14
6.2.2.14	remove_nodes	14
6.2.2.15	remove_trees	14
6.2.2.16	set_node_significant	14
6.2.2.17	set_object_scale	14
6.3	Mathematics Class Reference	14
6.3.1	Constructor & Destructor Documentation	15
6.3.1.1	Mathematics()	15
6.3.1.2	~Mathematics()	15
6.3.2	Member Function Documentation	15
6.3.2.1	flat_angle_from_vectors(QVector3D vector_a, QVector3D vector_b)	15
6.3.2.2	intersectYat(QVector3D raycast_vector, QVector3D camera_position, float y_↵ intersect_plane)	15
6.3.2.3	intersectYnull(QVector3D raycast_vector, QVector3D camera_position)	15
6.3.2.4	mouse_raycast(int mouse_x, int mouse_y, int screen_width, int screen_height, float invert_sign, QMatrix4x4 view_matrix, QMatrix4x4 projection_matrix)	15
6.3.2.5	point_on_line(float x, QVector3D point_a, QVector3D point_b)	15
6.3.2.6	return_near_degree(float degrees)	15
6.3.2.7	transform_3d_to_2d(QMatrix4x4 view_matrix, QMatrix4x4 projection_matrix, Q↵ Vector3D point, int screen_width, int screen_height)	15
6.4	ModelMesh Class Reference	16
6.4.1	Constructor & Destructor Documentation	16
6.4.1.1	ModelMesh(QString filename)	16
6.4.1.2	~ModelMesh()	16
6.4.2	Member Function Documentation	16
6.4.2.1	Draw()	16

6.4.2.2	LoadOBJ(QString filename)	16
6.4.3	Member Data Documentation	16
6.4.3.1	normalIndices	16
6.4.3.2	normals	16
6.4.3.3	textureCoordinates	16
6.4.3.4	uvIndices	16
6.4.3.5	vertexIndices	16
6.4.3.6	vertices	17
6.5	Node Class Reference	17
6.5.1	Constructor & Destructor Documentation	17
6.5.1.1	Node()	17
6.5.1.2	Node(QVector3D *position)	17
6.5.1.3	Node(QVector3D *position, QString *name)	18
6.5.2	Member Function Documentation	18
6.5.2.1	AddLink(QString *name, int index)	18
6.5.2.2	addShortest(int index)	18
6.5.2.3	clearPath()	18
6.5.2.4	countConnected()	18
6.5.2.5	getBike()	18
6.5.2.6	getColor()	18
6.5.2.7	getConnectedIndex(int index)	18
6.5.2.8	getG()	18
6.5.2.9	getLinkedName(int index)	18
6.5.2.10	getName()	18
6.5.2.11	getShortestIndex()	18
6.5.2.12	getSignificant()	18
6.5.2.13	getVehicle()	18
6.5.2.14	getWalk()	18
6.5.2.15	getWheelChair()	18
6.5.2.16	MoveLinkedIndexBack(int index)	18
6.5.2.17	Position()	18
6.5.2.18	RemoveLinkedFromIndex(int index)	18
6.5.2.19	setBike(bool can_use_bike)	18
6.5.2.20	setColor(QVector3D *rgb)	18
6.5.2.21	setDestinationNode()	18
6.5.2.22	setG(double g)	18
6.5.2.23	setName(QString)	19
6.5.2.24	setShortest(int index)	19
6.5.2.25	setSignificant(bool is_significant)	19
6.5.2.26	setSourceNode()	19

6.5.2.27	setVehicle(bool can_use_vehicle)	19
6.5.2.28	setWalk(bool can_walk)	19
6.5.2.29	setWheelChair(bool can_use_wheelchair)	19
6.6	NodeButton Class Reference	19
6.6.1	Constructor & Destructor Documentation	19
6.6.1.1	NodeButton(QWidget *parent=0)	19
6.6.1.2	~NodeButton()	19
6.6.2	Member Function Documentation	19
6.6.2.1	clicked_index	19
6.6.2.2	getIndex()	20
6.6.2.3	isDirectory()	20
6.6.2.4	mousePressEvent(QMouseEvent *event)	20
6.6.2.5	setDirectory(bool is_directory)	20
6.6.2.6	setIndex(int index)	20
6.7	NodeHandler Class Reference	20
6.7.1	Constructor & Destructor Documentation	20
6.7.1.1	NodeHandler()	20
6.7.2	Member Function Documentation	20
6.7.2.1	AddNode(Node *node)	20
6.7.2.2	AddNodeLink(int index, QString *name)	20
6.7.2.3	AddNodeLinkbyIndex(int index1, int index2)	20
6.7.2.4	CalculateShortest(int start, int finish)	20
6.7.2.5	count()	21
6.7.2.6	NodeFromIndex(unsigned int index)	21
6.7.2.7	pathcount()	21
6.7.2.8	pathindex(int shortest_index)	21
6.7.2.9	ReadFilePVC(QString filename)	21
6.8	PremisesExporter Class Reference	21
6.8.1	Constructor & Destructor Documentation	21
6.8.1.1	PremisesExporter()	21
6.8.2	Member Function Documentation	21
6.8.2.1	export_directories(QVector< QString > directory, QVector< QString > directory_list, QVector< QString > startup_menu, QString filename)	21
6.8.2.2	export_environment(QVector< VisualObject * > object_list, QString filename)	21
6.8.2.3	export_nodes(QVector< Node * > node_list, QString filename)	22
6.8.2.4	export_texture(QVector< QString > texture_paths, QString filename)	22
6.8.2.5	fileExists(QString filename)	22
6.9	RenderState Class Reference	22
6.9.1	Constructor & Destructor Documentation	22
6.9.1.1	RenderState(QWidget *parent=0)	22

6.9.1.2	<code>~RenderState()</code>	22
6.9.2	Member Function Documentation	22
6.9.2.1	<code>initializeGL()</code>	22
6.9.2.2	<code>mouseMoveEvent(QMouseEvent *event)</code>	23
6.9.2.3	<code>mousePressEvent(QMouseEvent *event)</code>	23
6.9.2.4	<code>mouseReleaseEvent(QMouseEvent *event)</code>	23
6.9.2.5	<code>opengl_initialised</code>	23
6.9.2.6	<code>paintGL()</code>	23
6.9.2.7	<code>resizeGL(int width, int height)</code>	23
6.9.2.8	<code>wheelEvent(QWheelEvent *event)</code>	23
6.10	Smtplib Class Reference	23
6.10.1	Constructor & Destructor Documentation	24
6.10.1.1	<code>Smtplib(const QString &user, const QString &pass, const QString &host, int port=465, int timeout=30000)</code>	24
6.10.1.2	<code>~Smtplib()</code>	24
6.10.2	Member Function Documentation	24
6.10.2.1	<code>sendMail(const QString &from, const QString &to, const QString &subject, const QString &body, QStringList files=QStringList())</code>	24
6.10.2.2	<code>status</code>	24
6.11	UserInterfaceCreator Class Reference	24
6.11.1	Constructor & Destructor Documentation	24
6.11.1.1	<code>UserInterfaceCreator(QWidget *parent=0)</code>	24
6.11.1.2	<code>~UserInterfaceCreator()</code>	24
6.12	VirtualConcierge Class Reference	24
6.12.1	Constructor & Destructor Documentation	25
6.12.1.1	<code>VirtualConcierge(QWidget *parent=0)</code>	25
6.12.1.2	<code>~VirtualConcierge()</code>	25
6.12.2	Member Function Documentation	25
6.12.2.1	<code>find_path</code>	25
6.13	VirtualConciergeRenderstate Class Reference	25
6.13.1	Constructor & Destructor Documentation	25
6.13.1.1	<code>VirtualConciergeRenderstate(QWidget *parent=0)</code>	25
6.13.1.2	<code>~VirtualConciergeRenderstate()</code>	25
6.13.2	Member Function Documentation	25
6.13.2.1	<code>initializeGL()</code>	25
6.13.2.2	<code>paintGL()</code>	26
6.13.2.3	<code>resizeGL(int w, int h)</code>	26
6.14	VisualObject Class Reference	26
6.14.1	Constructor & Destructor Documentation	27
6.14.1.1	<code>VisualObject(ModelMesh *modelmesh, QOpenGLTexture *texture, QVector3D translation, QVector3D rotation, QString type)</code>	27

6.14.1.2	<code>~VisualObject()</code>	27
6.14.2	Member Function Documentation	27
6.14.2.1	<code>getCornerLLeft()</code>	27
6.14.2.2	<code>getCornerLRight()</code>	27
6.14.2.3	<code>getCornerULeft()</code>	27
6.14.2.4	<code>getCornerURight()</code>	27
6.14.2.5	<code>getLMidHorizontal()</code>	27
6.14.2.6	<code>getModelMesh()</code>	27
6.14.2.7	<code>getRotation()</code>	27
6.14.2.8	<code>getScaling()</code>	27
6.14.2.9	<code>getTexture()</code>	27
6.14.2.10	<code>getTextureID()</code>	27
6.14.2.11	<code>getTexturePath()</code>	27
6.14.2.12	<code>getTranslation()</code>	27
6.14.2.13	<code>getType()</code>	27
6.14.2.14	<code>getUMidHorizontal()</code>	27
6.14.2.15	<code>setCornerLLeft(QVector3D position)</code>	28
6.14.2.16	<code>setCornerLRight(QVector3D position)</code>	28
6.14.2.17	<code>setCornerULeft(QVector3D position)</code>	28
6.14.2.18	<code>setCornerURight(QVector3D position)</code>	28
6.14.2.19	<code>setLMidHorizontal(QVector3D position)</code>	28
6.14.2.20	<code>setModel(ModelMesh *modelmesh)</code>	28
6.14.2.21	<code>setRotation(QVector3D rotation)</code>	28
6.14.2.22	<code>setScaling(QVector3D scale)</code>	28
6.14.2.23	<code>setTexture(QOpenGLTexture *texture)</code>	28
6.14.2.24	<code>setTextureID(int index)</code>	28
6.14.2.25	<code>setTexturePath(QString filename)</code>	28
6.14.2.26	<code>setTranslation(QVector3D translation)</code>	28
6.14.2.27	<code>setType(QString type)</code>	28
6.14.2.28	<code>setUMidHorizontal(QVector3D position)</code>	28
7	File Documentation	29
7.1	Functions/drawgl.cpp File Reference	29
7.2	Functions/drawgl.h File Reference	29
7.3	Functions/mathematics.cpp File Reference	29
7.4	Functions/mathematics.h File Reference	29
7.5	Functions/premises_exporter.cpp File Reference	30
7.6	Functions/premises_exporter.h File Reference	30
7.7	main.cpp File Reference	30
7.7.1	Function Documentation	30

7.7.1.1	main(int argc, char *argv[])	30
7.8	main.h File Reference	30
7.9	mainwindow.cpp File Reference	30
7.10	mainwindow.h File Reference	31
7.11	Objects/ModelMesh.cpp File Reference	31
7.12	Objects/ModelMesh.h File Reference	31
7.13	Objects/Node.cpp File Reference	31
7.14	Objects/Node.h File Reference	32
7.15	Objects/nodebutton.cpp File Reference	32
7.16	Objects/nodebutton.h File Reference	32
7.17	Objects/NodeHandler.cpp File Reference	32
7.18	Objects/NodeHandler.h File Reference	32
7.19	Objects/visualobject.cpp File Reference	33
7.20	Objects/visualobject.h File Reference	33
7.21	RenderStates/renderstate.cpp File Reference	33
7.22	RenderStates/renderstate.h File Reference	33
7.23	RenderStates/virtualconciergerenderstate.cpp File Reference	34
7.24	RenderStates/virtualconciergerenderstate.h File Reference	34
7.25	SMTP/smtp.cpp File Reference	34
7.26	SMTP/smtp.h File Reference	34
7.27	userinterfacecreator.cpp File Reference	35
7.28	userinterfacecreator.h File Reference	35
7.29	virtualconcierge.cpp File Reference	35
7.30	virtualconcierge.h File Reference	35
Index		37

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Ui	9
----------	---

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

DrawGL	11
Mathematics	14
ModelMesh	16
Node	17
NodeHandler	20
PremisesExporter	21
QDialog	
UserInterfaceCreator	24
QMainWindow	
MainWindow	12
QObject	
Sntp	23
QOpenGLFunctions	
RenderState	22
VirtualConciergeRenderstate	25
QOpenGLWidget	
RenderState	22
VirtualConciergeRenderstate	25
QPushButton	
NodeButton	19
QWidget	
VirtualConcierge	24
VisualObject	26

Chapter 3

Class Index

3.1 Class List

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

DrawGL	11
MainWindow	12
Mathematics	14
ModelMesh	16
Node	17
NodeButton	19
NodeHandler	20
PremisesExporter	21
RenderState	22
Sntp	23
UserInterfaceCreator	24
VirtualConcierge	24
VirtualConciergeRenderstate	25
VisualObject	26

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

main.cpp	30
main.h	30
mainwindow.cpp	30
mainwindow.h	31
userinterfacecreator.cpp	35
userinterfacecreator.h	35
virtualconciierge.cpp	35
virtualconciierge.h	35
Functions/ drawgl.cpp	29
Functions/ drawgl.h	29
Functions/ mathematics.cpp	29
Functions/ mathematics.h	29
Functions/ premises_exporter.cpp	30
Functions/ premises_exporter.h	30
Objects/ ModelMesh.cpp	31
Objects/ ModelMesh.h	31
Objects/ Node.cpp	31
Objects/ Node.h	32
Objects/ nodebutton.cpp	32
Objects/ nodebutton.h	32
Objects/ NodeHandler.cpp	32
Objects/ NodeHandler.h	32
Objects/ visualobject.cpp	33
Objects/ visualobject.h	33
RenderStates/ renderstate.cpp	33
RenderStates/ renderstate.h	33
RenderStates/ virtualconciierge_renderstate.cpp	34
RenderStates/ virtualconciierge_renderstate.h	34
SMTP/ smtp.cpp	34
SMTP/ smtp.h	34

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, QOpenGLShaderProgram *, QMatrix4x4, float)
- static void [ShaderDraw](#) ([ModelMesh](#) *, QOpenGLShaderProgram *)
- static void [UpdateShaders](#) (QMatrix4x4, QMatrix4x4, QMatrix4x4, QOpenGLTexture *, QVector3D, QVector2D, QOpenGLShaderProgram *, QMatrix4x4, float)
- static void [UpdateShaders](#) (QMatrix4x4, QMatrix4x4, QMatrix4x4, QVector3D, QVector2D, QOpenGLShaderProgram *, 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 *texturecoordinates*, 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 *texturecoordinates*, 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 Smtplib Class Reference

```
#include <smtplib.h>
```

Inheritance diagram for Smtplib:

Collaboration diagram for Smtplib:

Signals

- void [status](#) (const QString &)

Public Member Functions

- [Smtplib](#) (const QString &user, const QString &pass, const QString &host, int port=465, int timeout=30000)
- [~Smtplib](#) ()
- void [sendMail](#) (const QString &from, const QString &to, const QString &subject, const QString &body, Q<←
StringList files=QStringList())

6.10.1 Constructor & Destructor Documentation

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

6.10.1.2 `Smtplib::~Smtplib ()`

6.10.2 Member Function Documentation

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

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

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

- [SMTP/smtplib.h](#)
- [SMTP/smtplib.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 [setCornerULLeft](#) ([QVector3D](#) position)
- void [setCornerURRight](#) ([QVector3D](#) position)
- void [setCornerLLLeft](#) ([QVector3D](#) position)
- void [setCornerLRRight](#) ([QVector3D](#) position)
- void [setUMidHorizontal](#) ([QVector3D](#) position)
- void [setLMidHorizontal](#) ([QVector3D](#) position)
- void [setTextureID](#) (int index)
- void [setTexturePath](#) ([QString](#) filename)
- [QVector3D](#) [getCornerULLeft](#) ()
- [QVector3D](#) [getCornerURRight](#) ()
- [QVector3D](#) [getCornerLLLeft](#) ()
- [QVector3D](#) [getCornerLRRight](#) ()
- [QVector3D](#) [getUMidHorizontal](#) ()
- [QVector3D](#) [getLMidHorizontal](#) ()
- [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::getCornerLLLeft ()`

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::getLMidHorizontal ()`

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::getUMidHorizontal ()`

Here is the caller graph for this function:

- 6.14.2.15 void VisualObject::setCornerLLLeft (QVector3D *position*)
- 6.14.2.16 void VisualObject::setCornerLRight (QVector3D *position*)
- 6.14.2.17 void VisualObject::setCornerULLeft (QVector3D *position*)
- 6.14.2.18 void VisualObject::setCornerURight (QVector3D *position*)
- 6.14.2.19 void VisualObject::setLMidHorizontal (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::setUMidHorizontal (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 virtualconciierge.cpp File Reference

```
#include <QDebug>
#include <QFileDialog>
#include <QStringList>
#include <QSurfaceFormat>
#include "../virtualconciierge.h"
#include "../ui_virtualconciierge.h"
#include "SMTP/smtp.h"
Include dependency graph for virtualconciierge.cpp:
```

7.30 virtualconciierge.h File Reference

```
#include <QWidget>
#include <QVector>
#include <QPushButton>
#include "Objects/nodebutton.h"
#include "RenderStates/virtualconciergerenderstate.h"
Include dependency graph for virtualconciierge.h: This graph shows which files directly or indirectly include this file:
```

Classes

- class [VirtualConciierge](#)

Namespaces

- [Ui](#)

Index

- ~DrawGL
 - DrawGL, [11](#)
- ~MainWindow
 - MainWindow, [13](#)
- ~Mathematics
 - Mathematics, [15](#)
- ~ModelMesh
 - ModelMesh, [16](#)
- ~NodeButton
 - NodeButton, [19](#)
- ~RenderState
 - RenderState, [22](#)
- ~Smtplib
 - Smtplib, [24](#)
- ~UserInterfaceCreator
 - UserInterfaceCreator, [24](#)
- ~VirtualConcierge
 - VirtualConcierge, [25](#)
- ~VirtualConciergeRenderstate
 - VirtualConciergeRenderstate, [25](#)
- ~VisualObject
 - VisualObject, [27](#)
- add_new_texture
 - MainWindow, [13](#)
- AddLink
 - Node, [18](#)
- AddNode
 - NodeHandler, [20](#)
- AddNodeLink
 - NodeHandler, [20](#)
- AddNodeLinkbyIndex
 - NodeHandler, [20](#)
- addShortest
 - Node, [18](#)
- CalculateShortest
 - NodeHandler, [20](#)
- change_floor_selected
 - MainWindow, [13](#)
- change_node_name
 - MainWindow, [13](#)
- change_rotationY
 - MainWindow, [13](#)
- clearPath
 - Node, [18](#)
- clicked_index
 - NodeButton, [19](#)
- count
 - NodeHandler, [20](#)

- countConnected
 - Node, [18](#)
- Draw
 - ModelMesh, [16](#)
- draw_if_true
 - DrawGL, [11](#)
- DrawGL, [11](#)
 - ~DrawGL, [11](#)
 - draw_if_true, [11](#)
 - DrawGL, [11](#)
 - DrawLine, [11](#)
 - DrawModel, [12](#)
 - ShaderDraw, [12](#)
 - UpdateShaders, [12](#)
- DrawLine
 - DrawGL, [11](#)
- DrawModel
 - DrawGL, [12](#)
- export_directories
 - PremisesExporter, [21](#)
- export_environment
 - PremisesExporter, [21](#)
- export_nodes
 - PremisesExporter, [21](#)
- export_texture
 - PremisesExporter, [22](#)
- fileExists
 - PremisesExporter, [22](#)
- find_path
 - VirtualConcierge, [25](#)
- flat_angle_from_vectors
 - Mathematics, [15](#)
- Functions/drawgl.cpp, [29](#)
- Functions/drawgl.h, [29](#)
- Functions/mathematics.cpp, [29](#)
- Functions/mathematics.h, [29](#)
- Functions/premises_exporter.cpp, [30](#)
- Functions/premises_exporter.h, [30](#)
- getBike
 - Node, [18](#)
- getColor
 - Node, [18](#)
- getConnectedIndex
 - Node, [18](#)
- getCornerLLeft
 - VisualObject, [27](#)

- getCornerLRight
 - VisualObject, [27](#)
- getCornerULeft
 - VisualObject, [27](#)
- getCornerURight
 - VisualObject, [27](#)
- getG
 - Node, [18](#)
- getIndex
 - NodeButton, [19](#)
- getLMidHorizontal
 - VisualObject, [27](#)
- getLinkedName
 - Node, [18](#)
- getModelMesh
 - VisualObject, [27](#)
- getName
 - Node, [18](#)
- getRotation
 - VisualObject, [27](#)
- getScaling
 - VisualObject, [27](#)
- getShortestIndex
 - Node, [18](#)
- getSignificant
 - Node, [18](#)
- getTexture
 - VisualObject, [27](#)
- getTextureID
 - VisualObject, [27](#)
- getTexturePath
 - VisualObject, [27](#)
- getTranslation
 - VisualObject, [27](#)
- getType
 - VisualObject, [27](#)
- getUMidHorizontal
 - VisualObject, [27](#)
- getVehicle
 - Node, [18](#)
- getWalk
 - Node, [18](#)
- getWheelChair
 - Node, [18](#)
- initializeGL
 - RenderState, [22](#)
 - VirtualConciergeRenderstate, [25](#)
- intersectYat
 - Mathematics, [15](#)
- intersectYnull
 - Mathematics, [15](#)
- invert_mouseY
 - MainWindow, [13](#)
- isDirectory
 - NodeButton, [20](#)
- load_premises
 - MainWindow, [13](#)
- LoadOBJ
 - ModelMesh, [16](#)
- main
 - main.cpp, [30](#)
- main.cpp, [30](#)
 - main, [30](#)
- main.h, [30](#)
- MainWindow, [12](#)
 - ~MainWindow, [13](#)
 - add_new_texture, [13](#)
 - change_floor_selected, [13](#)
 - change_node_name, [13](#)
 - change_rotationY, [13](#)
 - invert_mouseY, [13](#)
 - load_premises, [13](#)
 - MainWindow, [13](#)
 - node_links, [13](#)
 - place_door, [13](#)
 - place_floor_plan, [14](#)
 - place_node, [14](#)
 - place_pavement, [14](#)
 - place_tree, [14](#)
 - place_wall, [14](#)
 - remove_nodes, [14](#)
 - remove_trees, [14](#)
 - set_node_significant, [14](#)
 - set_object_scale, [14](#)
- mainwindow.cpp, [30](#)
- mainwindow.h, [31](#)
- Mathematics, [14](#)
 - ~Mathematics, [15](#)
 - flat_angle_from_vectors, [15](#)
 - intersectYat, [15](#)
 - intersectYnull, [15](#)
 - Mathematics, [15](#)
 - mouse_raycast, [15](#)
 - point_on_line, [15](#)
 - return_near_degree, [15](#)
 - transform_3d_to_2d, [15](#)
- ModelMesh, [16](#)
 - ~ModelMesh, [16](#)
 - Draw, [16](#)
 - LoadOBJ, [16](#)
 - ModelMesh, [16](#)
 - normalIndices, [16](#)
 - normals, [16](#)
 - textureCoordinates, [16](#)
 - uvIndices, [16](#)
 - vertexIndices, [16](#)
 - vertices, [16](#)
- mouse_raycast
 - Mathematics, [15](#)
- mouseMoveEvent
 - RenderState, [22](#)
- mousePressEvent
 - NodeButton, [20](#)
 - RenderState, [23](#)
- mouseReleaseEvent

- RenderState, 23
- MoveLinkedIndexBack
 - Node, 18
- Node, 17
 - AddLink, 18
 - addShortest, 18
 - clearPath, 18
 - countConnected, 18
 - getBike, 18
 - getColor, 18
 - getConnectedIndex, 18
 - getG, 18
 - getLinkedName, 18
 - getName, 18
 - getShortestIndex, 18
 - getSignificant, 18
 - getVehicle, 18
 - getWalk, 18
 - getWheelChair, 18
 - MoveLinkedIndexBack, 18
 - Node, 17
 - Position, 18
 - RemoveLinkedFromIndex, 18
 - setBike, 18
 - setColor, 18
 - setDestinationNode, 18
 - setG, 18
 - setName, 18
 - setShortest, 19
 - setSignificant, 19
 - setSourceNode, 19
 - setVehicle, 19
 - setWalk, 19
 - setWheelChair, 19
- node_links
 - MainWindow, 13
- NodeButton, 19
 - ~NodeButton, 19
 - clicked_index, 19
 - getIndex, 19
 - isDirectory, 20
 - mousePressEvent, 20
 - NodeButton, 19
 - setDirectory, 20
 - setIndex, 20
- NodeFromIndex
 - NodeHandler, 21
- NodeHandler, 20
 - AddNode, 20
 - AddNodeLink, 20
 - AddNodeLinkbyIndex, 20
 - CalculateShortest, 20
 - count, 20
 - NodeFromIndex, 21
 - NodeHandler, 20
 - pathcount, 21
 - pathindex, 21
 - ReadFilePVC, 21
- normalIndices
 - ModelMesh, 16
- normals
 - ModelMesh, 16
- Objects/ModelMesh.cpp, 31
- Objects/ModelMesh.h, 31
- Objects/Node.cpp, 31
- Objects/Node.h, 32
- Objects/NodeHandler.cpp, 32
- Objects/NodeHandler.h, 32
- Objects/nodebutton.cpp, 32
- Objects/nodebutton.h, 32
- Objects/visualobject.cpp, 33
- Objects/visualobject.h, 33
- opengl_initialised
 - RenderState, 23
- paintGL
 - RenderState, 23
 - VirtualConciergeRenderstate, 25
- pathcount
 - NodeHandler, 21
- pathindex
 - NodeHandler, 21
- place_door
 - MainWindow, 13
- place_floor_plan
 - MainWindow, 14
- place_node
 - MainWindow, 14
- place_pavement
 - MainWindow, 14
- place_tree
 - MainWindow, 14
- place_wall
 - MainWindow, 14
- point_on_line
 - Mathematics, 15
- Position
 - Node, 18
- PremisesExporter, 21
 - export_directories, 21
 - export_environment, 21
 - export_nodes, 21
 - export_texture, 22
 - fileExists, 22
 - PremisesExporter, 21
- ReadFilePVC
 - NodeHandler, 21
- remove_nodes
 - MainWindow, 14
- remove_trees
 - MainWindow, 14
- RemoveLinkedFromIndex
 - Node, 18
- RenderState, 22
 - ~RenderState, 22

- initializeGL, 22
- mouseMoveEvent, 22
- mousePressEvent, 23
- mouseReleaseEvent, 23
- opengl_initialised, 23
- paintGL, 23
- RenderState, 22
- resizeGL, 23
- wheelEvent, 23
- RenderStates/renderstate.cpp, 33
- RenderStates/renderstate.h, 33
- RenderStates/virtualconciergerenderstate.cpp, 34
- RenderStates/virtualconciergerenderstate.h, 34
- resizeGL
 - RenderState, 23
 - VirtualConciergeRenderstate, 26
- return_near_degree
 - Mathematics, 15
- SMTP/smtp.cpp, 34
- SMTP/smtp.h, 34
- sendMail
 - Smtp, 24
- set_node_significant
 - MainWindow, 14
- set_object_scale
 - MainWindow, 14
- setBike
 - Node, 18
- setColor
 - Node, 18
- setCornerLLeft
 - VisualObject, 27
- setCornerLRight
 - VisualObject, 28
- setCornerULeft
 - VisualObject, 28
- setCornerURight
 - VisualObject, 28
- setDestinationNode
 - Node, 18
- setDirectory
 - NodeButton, 20
- setG
 - Node, 18
- setIndex
 - NodeButton, 20
- setLMidHorizontal
 - VisualObject, 28
- setModel
 - VisualObject, 28
- setName
 - Node, 18
- setRotation
 - VisualObject, 28
- setScaling
 - VisualObject, 28
- setShortest
 - Node, 19
- setSignificant
 - Node, 19
- setSourceNode
 - Node, 19
- setTexture
 - VisualObject, 28
- setTextureID
 - VisualObject, 28
- setTexturePath
 - VisualObject, 28
- setTranslation
 - VisualObject, 28
- setType
 - VisualObject, 28
- setUMidHorizontal
 - VisualObject, 28
- setVehicle
 - Node, 19
- setWalk
 - Node, 19
- setWheelChair
 - Node, 19
- ShaderDraw
 - DrawGL, 12
- Smtp, 23
 - ~Smtp, 24
 - sendMail, 24
 - Smtp, 24
 - status, 24
- status
 - Smtp, 24
- textureCoordinates
 - ModelMesh, 16
- transform_3d_to_2d
 - Mathematics, 15
- Ui, 9
- UpdateShaders
 - DrawGL, 12
- UserInterfaceCreator, 24
 - ~UserInterfaceCreator, 24
 - UserInterfaceCreator, 24
- userinterfacecreator.cpp, 35
- userinterfacecreator.h, 35
- uvIndices
 - ModelMesh, 16
- vertexIndices
 - ModelMesh, 16
- vertices
 - ModelMesh, 16
- VirtualConcierge, 24
 - ~VirtualConcierge, 25
 - find_path, 25
 - VirtualConcierge, 25
- VirtualConciergeRenderstate, 25
 - ~VirtualConciergeRenderstate, 25
 - initializeGL, 25

- paintGL, [25](#)
- resizeGL, [26](#)
- VirtualConciergeRenderstate, [25](#)
- virtualconcierge.cpp, [35](#)
- virtualconcierge.h, [35](#)
- VisualObject, [26](#)
 - ~VisualObject, [27](#)
 - getCornerLLeft, [27](#)
 - getCornerLRight, [27](#)
 - getCornerULeft, [27](#)
 - getCornerURight, [27](#)
 - getLMidHorizontal, [27](#)
 - getModelMesh, [27](#)
 - getRotation, [27](#)
 - getScaling, [27](#)
 - getTexture, [27](#)
 - getTextureID, [27](#)
 - getTexturePath, [27](#)
 - getTranslation, [27](#)
 - getType, [27](#)
 - getUMidHorizontal, [27](#)
 - setCornerLLeft, [27](#)
 - setCornerLRight, [28](#)
 - setCornerULeft, [28](#)
 - setCornerURight, [28](#)
 - setLMidHorizontal, [28](#)
 - setModel, [28](#)
 - setRotation, [28](#)
 - setScaling, [28](#)
 - setTexture, [28](#)
 - setTextureID, [28](#)
 - setTexturePath, [28](#)
 - setTranslation, [28](#)
 - setType, [28](#)
 - setUMidHorizontal, [28](#)
 - VisualObject, [27](#)
- wheelEvent
 - RenderState, [23](#)