

Castle

Généré par Doxygen 1.8.7

Lundi 2 Juin 2014 17 :01 :07

Table des matières

1	Liste des choses à faire	1
2	Index hiérarchique	3
2.1	Hiérarchie des classes	3
3	Index des classes	5
3.1	Liste des classes	5
4	Index des fichiers	7
4.1	Liste des fichiers	7
5	Documentation des classes	9
5.1	Référence de la classe Camera	9
5.1.1	Description détaillée	11
5.1.2	Documentation des constructeurs et destructeur	11
5.1.2.1	Camera	11
5.1.2.2	~Camera	11
5.1.3	Documentation des fonctions membres	12
5.1.3.1	display	12
5.1.3.2	go	12
5.1.3.3	mouseMoveEvent	12
5.1.3.4	move	13
5.1.3.5	position	13
5.1.3.6	vectorFromAngle	13
5.1.4	Documentation des données membres	14
5.1.4.1	_arriere_presse	14
5.1.4.2	_avant	14
5.1.4.3	_avant_presse	14
5.1.4.4	_bas_presse	14
5.1.4.5	_center	14
5.1.4.6	_droite_presse	14
5.1.4.7	_eye	14
5.1.4.8	_gauche	14

5.1.4.9	_gauche_presse	14
5.1.4.10	_haut	14
5.1.4.11	_haut_presse	14
5.1.4.12	_phi	14
5.1.4.13	_scene	14
5.1.4.14	_theta	15
5.1.4.15	_vitesse	15
5.2	Référence de la classe Cube	15
5.2.1	Description détaillée	18
5.2.2	Documentation des constructeurs et destructeur	18
5.2.2.1	Cube	18
5.2.2.2	~Cube	18
5.2.3	Documentation des fonctions membres	18
5.2.3.1	draw	18
5.2.3.2	genVao	19
5.2.4	Documentation des données membres	19
5.2.4.1	_vao	19
5.3	Référence de la classe Donuts	19
5.3.1	Description détaillée	22
5.3.2	Documentation des constructeurs et destructeur	22
5.3.2.1	Donuts	22
5.3.2.2	~Donuts	22
5.3.3	Documentation des fonctions membres	22
5.3.3.1	draw	22
5.3.3.2	genVao	23
5.3.4	Documentation des données membres	23
5.3.4.1	_vao	23
5.3.4.2	m_radius	23
5.3.4.3	m_radius_donuts	23
5.3.4.4	m_slices	23
5.3.4.5	m_stacks	23
5.3.4.6	nbvertex	23
5.4	Référence de la classe Light	23
5.4.1	Description détaillée	26
5.4.2	Documentation des constructeurs et destructeur	26
5.4.2.1	Light	26
5.4.2.2	~Light	26
5.4.3	Documentation des fonctions membres	27
5.4.3.1	ambient	27
5.4.3.2	ambient	28

5.4.3.3	disable	28
5.4.3.4	get	28
5.4.3.5	get	29
5.4.3.6	set	29
5.4.3.7	update	30
5.4.3.8	updateAmbient	30
5.4.4	Documentation des données membres	31
5.4.4.1	_ambient	31
5.4.4.2	_diffuse	31
5.4.4.3	_position	31
5.4.4.4	_specular	31
5.5	Référence de la classe MainWindow	31
5.5.1	Documentation des constructeurs et destructeur	36
5.5.1.1	MainWindow	37
5.5.1.2	~MainWindow	37
5.5.2	Documentation des fonctions membres	37
5.5.2.1	aboutassimp	37
5.5.2.2	affichagerecnoderestant	38
5.5.2.3	ajoutlumiere	38
5.5.2.4	ajoutmaterial	38
5.5.2.5	ajoutmur	39
5.5.2.6	ajoutmурcreatemodelmurcombo	39
5.5.2.7	ajoutobjet	40
5.5.2.8	ajoutpiece	40
5.5.2.9	Apercut	41
5.5.2.10	changeshader	41
5.5.2.11	createActions	42
5.5.2.12	createListeDockwidget	42
5.5.2.13	createMenus	43
5.5.2.14	createToolBar	43
5.5.2.15	Culface	44
5.5.2.16	Enregistrer	44
5.5.2.17	Grille	45
5.5.2.18	Importation3D	45
5.5.2.19	Nouveau	46
5.5.2.20	Ouvrir	46
5.5.2.21	reloadshader	46
5.5.2.22	selectioncheminobj	47
5.5.2.23	supresionelement	47
5.5.2.24	validajoutlumiere	47

5.5.2.25	validajoutmaterial	48
5.5.2.26	validajoutmur	48
5.5.2.27	validajoutpiece	49
5.5.2.28	validationajoutobjet	50
5.5.3	Documentation des données membres	50
5.5.3.1	about	50
5.5.3.2	aboutassimpAct	50
5.5.3.3	aboutqtAct	50
5.5.3.4	Affichage	50
5.5.3.5	Aide	50
5.5.3.6	ajoutelement	50
5.5.3.7	ajoutlumierAct	50
5.5.3.8	ajoutmaterialAct	50
5.5.3.9	ajoutmurAct	50
5.5.3.10	ajoutobjetAct	50
5.5.3.11	ajoutpieceAct	51
5.5.3.12	ajoutpiecedimx	51
5.5.3.13	ajoutpiecedimy	51
5.5.3.14	ajoutpiecedimz	51
5.5.3.15	ajoutpiececelabmure	51
5.5.3.16	ajoutpiececelayoutcombo	51
5.5.3.17	ajoutpiececelayoutdim	51
5.5.3.18	ajoutpiececelayoutnom	51
5.5.3.19	ajoutpiecepiece1	51
5.5.3.20	ajoutpiecepiece2	51
5.5.3.21	ajoutpiecepiece3	51
5.5.3.22	ajoutpieceshaderlayout	51
5.5.3.23	ambajoutmaterial	51
5.5.3.24	apercutAct	51
5.5.3.25	boutonajoutlumiere	51
5.5.3.26	boutonajoutmaterial	51
5.5.3.27	boutonajoutmur	51
5.5.3.28	boutonajoutobjet	51
5.5.3.29	boutonajoutpiece	51
5.5.3.30	boutonlisteelement	51
5.5.3.31	boutonlisteelementdelete	51
5.5.3.32	boutonpieceajoutmur	51
5.5.3.33	boutonselectionobjajoutobjet	51
5.5.3.34	champnomajoutobjet	51
5.5.3.35	checkmur1	51

5.5.3.36	checkmur2	51
5.5.3.37	checkmur3	51
5.5.3.38	checkmur4	51
5.5.3.39	checkmur5	52
5.5.3.40	checkmur6	52
5.5.3.41	comboajoutpiece	52
5.5.3.42	comboajoutpieceshader	52
5.5.3.43	combomurajoutmur	52
5.5.3.44	combopieceajoutmur	52
5.5.3.45	combopieceajoutobjet	52
5.5.3.46	culfaceAct	52
5.5.3.47	difajoutlumiere	52
5.5.3.48	difajoutmaterial	52
5.5.3.49	dock_list_elements	52
5.5.3.50	dock_perso	52
5.5.3.51	dockajoutlumiere	52
5.5.3.52	dockajoutmateriaux	52
5.5.3.53	dockajoutmur	52
5.5.3.54	dockajoutobjet	52
5.5.3.55	dockajoutpiece	52
5.5.3.56	Edition	52
5.5.3.57	enregistrerAct	52
5.5.3.58	Fichier	52
5.5.3.59	fileToolBar	52
5.5.3.60	grilleAct	52
5.5.3.61	importation3DAct	52
5.5.3.62	item	52
5.5.3.63	kiterAct	52
5.5.3.64	labelajoutpieceshader	52
5.5.3.65	labelcheminobjajoutobj	52
5.5.3.66	labelcomboajoutpiece	52
5.5.3.67	labeldimajoutpiece	53
5.5.3.68	labelmur1	53
5.5.3.69	labelmur2	53
5.5.3.70	labelmur3	53
5.5.3.71	labelmur4	53
5.5.3.72	labelmur5	53
5.5.3.73	labelmur6	53
5.5.3.74	labelmurajoutpiece	53
5.5.3.75	labelmurcomboajoutmur	53

5.5.3.76	labelnomajoutobjet	53
5.5.3.77	labelnomajoutpiece	53
5.5.3.78	labelpieceajoutobjet	53
5.5.3.79	labelpiececomboajoutmur	53
5.5.3.80	layoutajoutlumiere	53
5.5.3.81	layoutajoutmaterial	53
5.5.3.82	layoutajoutobjetnom	53
5.5.3.83	layoutajoutobjetpiece	53
5.5.3.84	layoutambajoutmaterial	53
5.5.3.85	layoutcheminajoutobjet	53
5.5.3.86	layoutcombotnommurajoutmur	53
5.5.3.87	layoutcombotnompieceajoutmur	53
5.5.3.88	layoutdifajoutlumiere	53
5.5.3.89	layoutdifajoutmaterial	53
5.5.3.90	layoutlistescene	53
5.5.3.91	layoutnomajoutlumiere	53
5.5.3.92	layoutnomajoutmaterial	53
5.5.3.93	layoutposajoutlumiere	53
5.5.3.94	layoutspeajoutlumiere	53
5.5.3.95	layoutspeajoutmaterial	54
5.5.3.96	light	54
5.5.3.97	lineeditajoutpiece	54
5.5.3.98	lineeditnomajoutlumiere	54
5.5.3.99	lineeditnomajoutmaterial	54
5.5.3.100	lineeditobjajoutobj	54
5.5.3.101	listtemplight	54
5.5.3.102	listtempmaterial	54
5.5.3.103	listtempobjet	54
5.5.3.104	listitemshader	54
5.5.3.105	mainlayoutajoutmur	54
5.5.3.106	mainlayoutajoutobjet	54
5.5.3.107	mainlayoutajoutpiece	54
5.5.3.108	material	54
5.5.3.109	modelajoutmurmur	54
5.5.3.110	modelajoutmurpiece	54
5.5.3.111	modelajoutobjet	54
5.5.3.112	modele	54
5.5.3.113	modelemateriaupiece	54
5.5.3.114	modelepieceshader	54
5.5.3.115	modelshaderpostpro	54

5.5.3.116 nomajoutlumiere	54
5.5.3.117 nomajoutmaterial	54
5.5.3.118 nouveauAct	54
5.5.3.119 objet	54
5.5.3.120 Outil	54
5.5.3.121 ouvrirAct	54
5.5.3.122 positionajoutlumiere	54
5.5.3.123 reloadshaderAct	55
5.5.3.124 scenetemp	55
5.5.3.125 shader	55
5.5.3.126 shaderpospro	55
5.5.3.127 speajoutlumiere	55
5.5.3.128 speajoutmaterial	55
5.5.3.129 spinambajoutmaterialx	55
5.5.3.130 spinambajoutmaterialy	55
5.5.3.131 spinambajoutmaterialz	55
5.5.3.132 spindifajoutlumierex	55
5.5.3.133 spindifajoutlumierey	55
5.5.3.134 spindifajoutlumierez	55
5.5.3.135 spindifajoutmaterialx	55
5.5.3.136 spindifajoutmaterialy	55
5.5.3.137 spindifajoutmaterialz	55
5.5.3.138 spinpositionajoutlumierex	55
5.5.3.139 spinpositionajoutlumierey	55
5.5.3.140 spinpositionajoutlumierez	55
5.5.3.141 spinspeajoutlumierex	55
5.5.3.142 spinspeajoutlumierey	55
5.5.3.143 spinspeajoutlumierez	55
5.5.3.144 spinspeajoutmateriala	55
5.5.3.145 spinspeajoutmaterialx	55
5.5.3.146 spinspeajoutmaterialy	55
5.5.3.147 spinspeajoutmaterialz	55
5.5.3.148 vue	55
5.5.3.149 widget	55
5.5.3.150 widgetajoutlumiere	55
5.5.3.151 widgetajoutmaterial	56
5.5.3.152 widgetajoutmur	56
5.5.3.153 widgetajoutobjet	56
5.5.3.154 widgetajoutpiece	56
5.5.3.155 widgetdocklistscene	56

5.6	Référence de la structure mat3	56
5.6.1	Documentation des constructeurs et destructeur	57
5.6.1.1	mat3	57
5.6.1.2	mat3	57
5.6.1.3	mat3	57
5.6.1.4	mat3	57
5.6.2	Documentation des fonctions membres	57
5.6.2.1	operator*	57
5.6.2.2	operator*	58
5.6.2.3	operator*	58
5.6.2.4	operator*=	58
5.6.2.5	operator+	58
5.6.2.6	operator+=	58
5.6.2.7	operator-	58
5.6.2.8	operator-=	58
5.6.2.9	operator=	58
5.6.2.10	operator[]	59
5.6.2.11	operator[]	59
5.6.3	Documentation des données membres	59
5.6.3.1	m	59
5.7	Référence de la structure mat4	59
5.7.1	Documentation des constructeurs et destructeur	60
5.7.1.1	mat4	60
5.7.1.2	mat4	60
5.7.1.3	mat4	60
5.7.1.4	mat4	60
5.7.1.5	mat4	60
5.7.2	Documentation des fonctions membres	60
5.7.2.1	det	60
5.7.2.2	inverse	61
5.7.2.3	operator*	61
5.7.2.4	operator*	61
5.7.2.5	operator*	61
5.7.2.6	operator*=	61
5.7.2.7	operator*=	61
5.7.2.8	operator+	62
5.7.2.9	operator+=	62
5.7.2.10	operator-	62
5.7.2.11	operator-=	62
5.7.2.12	operator=	62

5.7.2.13 <code>operator[]</code>	62
5.7.2.14 <code>operator[]</code>	62
5.7.2.15 <code>transpose</code>	62
5.7.3 Documentation des données membres	63
5.7.3.1 <code>m</code>	63
5.8 Référence de la classe Material	63
5.8.1 Description détaillée	66
5.8.2 Documentation des constructeurs et destructeur	66
5.8.2.1 <code>Material</code>	67
5.8.2.2 <code>~Material</code>	67
5.8.3 Documentation des fonctions membres	67
5.8.3.1 <code>addTexture</code>	67
5.8.3.2 <code>clear</code>	67
5.8.3.3 <code>fromXML</code>	68
5.8.3.4 <code>get</code>	68
5.8.3.5 <code>get</code>	68
5.8.3.6 <code>getDiffuseTextureName</code>	69
5.8.3.7 <code>getNormalTextureName</code>	69
5.8.3.8 <code>getSpecularTextureName</code>	69
5.8.3.9 <code>hasDiffuseTexture</code>	69
5.8.3.10 <code>hasNormalTexture</code>	70
5.8.3.11 <code>hasSpecularTexture</code>	70
5.8.3.12 <code>hasTexture</code>	70
5.8.3.13 <code>isFromXML</code>	70
5.8.3.14 <code>set</code>	70
5.8.3.15 <code>set</code>	71
5.8.3.16 <code>shininess</code>	71
5.8.3.17 <code>update</code>	71
5.8.4 Documentation des fonctions amies et associées	72
5.8.4.1 <code>operator<<</code>	72
5.8.5 Documentation des données membres	72
5.8.5.1 <code>_ambient</code>	72
5.8.5.2 <code>_diffuse</code>	72
5.8.5.3 <code>_diffuse_texture</code>	72
5.8.5.4 <code>_emissive</code>	72
5.8.5.5 <code>_fromXML</code>	72
5.8.5.6 <code>_normal_texture</code>	73
5.8.5.7 <code>_shininess</code>	73
5.8.5.8 <code>_specular</code>	73
5.8.5.9 <code>_specular_texture</code>	73

5.8.5.10 <code>_texturesLoaded</code>	73
5.9 Référence de la classe <code>Mesh</code>	73
5.9.1 Description détaillée	76
5.9.2 Documentation des constructeurs et destructeur	76
5.9.2.1 <code>Mesh</code>	76
5.9.2.2 <code>Mesh</code>	77
5.9.2.3 <code>~Mesh</code>	77
5.9.3 Documentation des fonctions membres	77
5.9.3.1 <code>draw</code>	77
5.9.3.2 <code>loadMesh</code>	77
5.9.4 Documentation des fonctions amies et associées	78
5.9.4.1 <code>Node</code>	78
5.9.5 Documentation des données membres	78
5.9.5.1 <code>_infos</code>	78
5.10 Référence de la structure <code>Mesh::MeshInfo</code>	78
5.10.1 Documentation des données membres	78
5.10.1.1 <code>nbFaces</code>	78
5.10.1.2 <code>nbReferences</code>	78
5.10.1.3 <code>vao</code>	78
5.10.1.4 <code>vbos</code>	79
5.11 Référence de la classe <code>Mondock</code>	79
5.11.1 Documentation des constructeurs et destructeur	84
5.11.1.1 <code>Mondock</code>	84
5.11.1.2 <code>~Mondock</code>	84
5.11.2 Documentation des fonctions membres	84
5.11.2.1 <code>amblightfuncx</code>	85
5.11.2.2 <code>amblightfuncy</code>	85
5.11.2.3 <code>amblightfuncz</code>	86
5.11.2.4 <code>ambmaterialfuncx</code>	86
5.11.2.5 <code>ambmaterialfuncy</code>	87
5.11.2.6 <code>ambmaterialfuncz</code>	87
5.11.2.7 <code>diflightfuncx</code>	88
5.11.2.8 <code>diflightfuncy</code>	88
5.11.2.9 <code>diflightfuncz</code>	89
5.11.2.10 <code>difmaterialfuncx</code>	89
5.11.2.11 <code>difmaterialfuncy</code>	90
5.11.2.12 <code>difmaterialfuncz</code>	90
5.11.2.13 <code>lumiereambiantejustx</code>	91
5.11.2.14 <code>lumiereambiantejusty</code>	91
5.11.2.15 <code>lumiereambiantejustz</code>	92

5.11.2.16 matobjet	92
5.11.2.17 pereobjet	93
5.11.2.18 poslightfuncx	93
5.11.2.19 poslightfuncy	94
5.11.2.20 poslightfuncz	94
5.11.2.21 rotobjectx	95
5.11.2.22 rotobjecty	95
5.11.2.23 rotobjectz	96
5.11.2.24 rotpiececx	96
5.11.2.25 rotpiececy	97
5.11.2.26 rotpiececz	97
5.11.2.27 scaleobjectx	98
5.11.2.28 scaleobjecty	98
5.11.2.29 scaleobjectz	99
5.11.2.30 scalepiececx	99
5.11.2.31 scalepiececy	100
5.11.2.32 scalepiececz	100
5.11.2.33 scalesliderobject	101
5.11.2.34 scalesliderpiece	101
5.11.2.35 selectionlight	102
5.11.2.36 slotpositionpiececx	103
5.11.2.37 slotpositionpiececy	103
5.11.2.38 slotpositionpiececz	104
5.11.2.39 spelightfuncx	104
5.11.2.40 spelightfuncy	105
5.11.2.41 spelightfuncz	105
5.11.2.42 spematerialfuncx	106
5.11.2.43 spematerialfuncy	106
5.11.2.44 spematerialfuncz	107
5.11.2.45 spematerialfuncu	107
5.11.2.46 traitementambiante	108
5.11.2.47 traitementlumiere	109
5.11.2.48 traitementmaterial	110
5.11.2.49 traitementobjet	111
5.11.2.50 traitementpiece	112
5.11.2.51 transobjectx	113
5.11.2.52 transobjecty	113
5.11.2.53 transobjectz	114
5.11.3 Documentation des données membres	114
5.11.3.1 _itemmaterial	114

5.11.3.2 _itempiece	114
5.11.3.3 _light	114
5.11.3.4 _materiaux	114
5.11.3.5 _objet	114
5.11.3.6 _piece	114
5.11.3.7 boxobjetrotationx	114
5.11.3.8 boxobjetrotationy	114
5.11.3.9 boxobjetrotationz	114
5.11.3.10 boxobjetscalex	114
5.11.3.11 boxobjetscaley	114
5.11.3.12 boxobjetscalez	114
5.11.3.13 boxobjettransx	114
5.11.3.14 boxobjettransy	114
5.11.3.15 boxobjettransz	114
5.11.3.16 combomaterial	115
5.11.3.17 comboparent	115
5.11.3.18 dimentionpiecex	115
5.11.3.19 dimentionpiecey	115
5.11.3.20 dimentionpiecez	115
5.11.3.21 dockmodele	115
5.11.3.22 dockscene	115
5.11.3.23 dockvue	115
5.11.3.24 editionambiante	115
5.11.3.25 elementSelectionneGParent	115
5.11.3.26 elementSelectionneParent	115
5.11.3.27 indexElementSelectionne	115
5.11.3.28 indexgparent	115
5.11.3.29 indexlightSelectionne	115
5.11.3.30 indexmaterialSelectionne	115
5.11.3.31 indexobjetSelectionne	115
5.11.3.32 indexpieceSelectionne	115
5.11.3.33 labelamblayout	115
5.11.3.34 labelambx	115
5.11.3.35 labelamby	115
5.11.3.36 labelambz	115
5.11.3.37 labeldiflayout	115
5.11.3.38 labeldifx	115
5.11.3.39 labelfify	115
5.11.3.40 labelfifz	115
5.11.3.41 labelobjetproprmaterial	115

5.11.3.42 labelobjetproprparent	115
5.11.3.43 labelsliderobjetscale	115
5.11.3.44 labelspelayout	116
5.11.3.45 labelspepx	116
5.11.3.46 labelspey	116
5.11.3.47 labelspez	116
5.11.3.48 layouteditionlighamb	116
5.11.3.49 layoutglobalscaleobjet	116
5.11.3.50 layoutobjetscaleall	116
5.11.3.51 layouttabobjetprop	116
5.11.3.52 layouttabobjetpropcombo	116
5.11.3.53 layouttabobjetproplabel	116
5.11.3.54 layouttabobjetrotation	116
5.11.3.55 layouttabobjetscale	116
5.11.3.56 layouttabobjettrans	116
5.11.3.57 lightamblayout	116
5.11.3.58 lightdifflayout	116
5.11.3.59 lightdoublespinboxx	116
5.11.3.60 lightdoublespinboxy	116
5.11.3.61 lightdoublespinboxz	116
5.11.3.62 lightselectioner	116
5.11.3.63 lightspelayout	116
5.11.3.64 materialselectioner	116
5.11.3.65 materialspinboxambx	116
5.11.3.66 materialspinboxamby	116
5.11.3.67 materialspinboxambz	116
5.11.3.68 materialspinboxdfix	116
5.11.3.69 materialspinboxdfy	116
5.11.3.70 materialspinboxdfz	116
5.11.3.71 materialspinboxspea	116
5.11.3.72 materialspinboxspex	117
5.11.3.73 materialspinboxspey	117
5.11.3.74 materialspinboxspez	117
5.11.3.75 modelmaterial	117
5.11.3.76 modelpiece	117
5.11.3.77 objetselectioner	117
5.11.3.78 piecedimlayout	117
5.11.3.79 pieceposilayout	117
5.11.3.80 piecerotatelayout	117
5.11.3.81 piecescalelayout	117

5.11.3.82 pieceselectioner	117
5.11.3.83 positionpiecex	117
5.11.3.84 positionpiecey	117
5.11.3.85 positionpiecez	117
5.11.3.86 rotspinboxx	117
5.11.3.87 rotspinboxy	117
5.11.3.88 rotspinboxz	117
5.11.3.89 scalelabel	117
5.11.3.90 scaleslider	117
5.11.3.91 scalesliderlayout	117
5.11.3.92 scalespinboxlayout	117
5.11.3.93 scalespinboxx	117
5.11.3.94 scalespinboxy	117
5.11.3.95 scalespinboxz	117
5.11.3.96 selection	117
5.11.3.97 sliderambx	117
5.11.3.98 slideramby	117
5.11.3.99 sliderambz	117
5.11.3.100sliderdifx	118
5.11.3.101sliderdify	118
5.11.3.102sliderdifz	118
5.11.3.103sliderobjetscale	118
5.11.3.104sliderspex	118
5.11.3.105sliderspey	118
5.11.3.106sliderspez	118
5.11.3.107spinboxlumiereambiantex	118
5.11.3.108spinboxlumiereambiantey	118
5.11.3.109spinboxlumiereambiantez	118
5.11.3.110tabamblayout	118
5.11.3.111tabdiflayout	118
5.11.3.112ablight	118
5.11.3.113ablightamb	118
5.11.3.114ablightdif	118
5.11.3.115ablightpos	118
5.11.3.116ablightposlayout	118
5.11.3.117ablightspe	118
5.11.3.118abmaterial	118
5.11.3.119abmaterialamb	118
5.11.3.120abmaterialamblayout	118
5.11.3.121tabmaterialdif	118

5.11.3.12 <code>tabmaterialdiflayout</code>	118
5.11.3.12 <code>tabmaterialspe</code>	118
5.11.3.12 <code>tabmaterialspeLayout</code>	118
5.11.3.12 <code>tabobjet</code>	118
5.11.3.12 <code>tabobjetprop</code>	118
5.11.3.12 <code>tabobjetrotation</code>	118
5.11.3.12 <code>tabobjetscale</code>	119
5.11.3.12 <code>tabobjettrans</code>	119
5.11.3.13 <code>tabpiece</code>	119
5.11.3.13 <code>tabspelayout</code>	119
5.11.3.13 <code>widgetpiececdim</code>	119
5.11.3.13 <code>widgetpieceposi</code>	119
5.11.3.13 <code>widgetpiicerotate</code>	119
5.11.3.13 <code>widgetpiecescale</code>	119
5.12 Référence de la classe <code>MyOpenGLWidget</code>	119
5.12.1 Documentation des constructeurs et destructeur	122
5.12.1.1 <code>MyOpenGLWidget</code>	122
5.12.1.2 <code>~MyOpenGLWidget</code>	122
5.12.2 Documentation des fonctions membres	122
5.12.2.1 <code>addShader</code>	122
5.12.2.2 <code>getScene</code>	123
5.12.2.3 <code>getShaderNames</code>	123
5.12.2.4 <code>initFramebuffer</code>	123
5.12.2.5 <code>initializeGL</code>	123
5.12.2.6 <code>keyPressEvent</code>	124
5.12.2.7 <code>keyReleaseEvent</code>	124
5.12.2.8 <code>loadShaders</code>	124
5.12.2.9 <code>minimumSizeHint</code>	124
5.12.2.10 <code>mouseMoveEvent</code>	124
5.12.2.11 <code>mousePressEvent</code>	125
5.12.2.12 <code>paintGL</code>	125
5.12.2.13 <code>resizeGL</code>	125
5.12.2.14 <code>saveShaders</code>	125
5.12.2.15 <code>setScene</code>	125
5.12.2.16 <code>sizeHint</code>	125
5.12.2.17 <code>useShader</code>	126
5.12.3 Documentation des données membres	126
5.12.3.1 <code>_activeProgram</code>	126
5.12.3.2 <code>_captureMouse</code>	126
5.12.3.3 <code>_framebuffer</code>	126

5.12.3.4 _loadedShaders	126
5.12.3.5 _path	126
5.12.3.6 _pixel_scale	126
5.12.3.7 _postProcessPrograms	126
5.12.3.8 _renderbuffer	126
5.12.3.9 _scene	126
5.12.3.10 _texture	126
5.12.3.11 _timer	126
5.12.3.12 _vao_quad	126
5.13 Référence de la classe Node	126
5.13.1 Description détaillée	129
5.13.2 Documentation des constructeurs et destructeur	129
5.13.2.1 Node	129
5.13.2.2 Node	130
5.13.2.3 ~Node	130
5.13.3 Documentation des fonctions membres	130
5.13.3.1 draw	130
5.13.3.2 getChild	131
5.13.3.3 getChildrenNames	131
5.13.3.4 getMesh	131
5.13.3.5 getMeshNames	132
5.13.3.6 getModelName	132
5.13.3.7 loadMaterial	132
5.13.3.8 loadModel	133
5.13.3.9 loadNode	134
5.13.3.10 removeChild	135
5.13.3.11 removeMesh	135
5.13.4 Documentation des données membres	136
5.13.4.1 _children	136
5.13.4.2 _loadedModels	136
5.13.4.3 _meshs	136
5.14 Référence de la classe Objet	136
5.14.1 Description détaillée	139
5.14.2 Documentation des constructeurs et destructeur	140
5.14.2.1 Objet	140
5.14.2.2 Objet	140
5.14.2.3 ~Objet	140
5.14.3 Documentation des fonctions membres	140
5.14.3.1 activateShader	140
5.14.3.2 applyMaterial	141

5.14.3.3 draw	142
5.14.3.4 drawHitbox	143
5.14.3.5 drawHitbox	143
5.14.3.6 material	144
5.14.3.7 material	144
5.14.3.8 name	144
5.14.3.9 name	144
5.14.3.10 parent	145
5.14.3.11 parent	145
5.14.3.12 position	145
5.14.3.13 position	146
5.14.3.14 position	146
5.14.3.15 rotation	146
5.14.3.16 rotation	147
5.14.3.17 rotation	147
5.14.3.18 scale	147
5.14.3.19 scale	148
5.14.3.20 scale	148
5.14.3.21 shaderId	149
5.14.3.22 shaderId	150
5.14.3.23 updateModel	150
5.14.4 Documentation des données membres	151
5.14.4.1 _drawHitbox	151
5.14.4.2 _mat	151
5.14.4.3 _model	151
5.14.4.4 _name	151
5.14.4.5 _parent	151
5.14.4.6 _position	151
5.14.4.7 _rotation	151
5.14.4.8 _scale	151
5.14.4.9 _shaderId	151
5.15 Référence de la classe Piece	151
5.15.1 Description détaillée	154
5.15.2 Documentation des constructeurs et destructeur	154
5.15.2.1 Piece	154
5.15.2.2 ~Piece	154
5.15.3 Documentation des fonctions membres	154
5.15.3.1 addChild	154
5.15.3.2 addChild	155
5.15.3.3 deleteChild	155

5.15.3.4	dimensions	155
5.15.3.5	dimensions	155
5.15.3.6	dimensions	156
5.15.3.7	draw	156
5.15.3.8	getChild	156
5.15.3.9	getChildren	157
5.15.3.10	removeChild	157
5.15.4	Documentation des données membres	157
5.15.4.1	_children	157
5.15.4.2	_dimensions	157
5.16	Référence de la classe Plan	157
5.16.1	Description détaillée	160
5.16.2	Documentation des constructeurs et destructeur	160
5.16.2.1	Plan	160
5.16.2.2	~Plan	160
5.16.3	Documentation des fonctions membres	160
5.16.3.1	draw	160
5.16.3.2	getFenetres	161
5.16.4	Documentation des données membres	161
5.16.4.1	_fenetres	161
5.16.4.2	_maxP	161
5.16.4.3	_minP	161
5.16.4.4	_nblIndices	161
5.16.4.5	_vao	161
5.16.4.6	_vbo_indices	161
5.16.4.7	_vbo_normals	161
5.16.4.8	_vbo_texCoord	161
5.16.4.9	_vbo_vertices	161
5.17	Référence de la classe Scene	161
5.17.1	Description détaillée	165
5.17.2	Documentation des constructeurs et destructeur	165
5.17.2.1	Scene	165
5.17.2.2	Scene	165
5.17.2.3	~Scene	166
5.17.3	Documentation des fonctions membres	166
5.17.3.1	addLight	166
5.17.3.2	addMaterial	167
5.17.3.3	addPiece	167
5.17.3.4	addShader	168
5.17.3.5	draw	168

5.17.3.6	getLight	169
5.17.3.7	getLightsNames	170
5.17.3.8	getMaterial	170
5.17.3.9	getMaterialName	171
5.17.3.10	getMaterialsNames	172
5.17.3.11	getPiece	172
5.17.3.12	getPiecesName	173
5.17.3.13	getShader	173
5.17.3.14	getShaderNameByID	173
5.17.3.15	getShadersNames	174
5.17.3.16	loadLights	174
5.17.3.17	loadMaterials	175
5.17.3.18	loadPieces	175
5.17.3.19	loadShaders	176
5.17.3.20	orderLights	177
5.17.3.21	readPosition	178
5.17.3.22	readRotation	179
5.17.3.23	readScale	179
5.17.3.24	removeLight	180
5.17.3.25	removePiece	180
5.17.3.26	saveAsXML	180
5.17.3.27	saveLights	181
5.17.3.28	saveMaterials	181
5.17.3.29	savePieces	182
5.17.3.30	saveShaders	182
5.17.4	Documentation des données membres	183
5.17.4.1	_camera	183
5.17.4.2	_lights	183
5.17.4.3	_loadedShaders	183
5.17.4.4	_materials	183
5.17.4.5	_orderedLights	183
5.17.4.6	_parent	183
5.17.4.7	_pieces	183
5.17.4.8	_projectionMatrix	183
5.17.4.9	_shaders	183
5.18	Référence de la classe Sphere	183
5.18.1	Description détaillée	186
5.18.2	Documentation des constructeurs et destructeur	186
5.18.2.1	Sphere	186
5.18.2.2	~Sphere	186

5.18.3 Documentation des fonctions membres	186
5.18.3.1 draw	186
5.18.3.2 genVao	187
5.18.4 Documentation des données membres	187
5.18.4.1 _vao	187
5.18.4.2 m_radius	187
5.18.4.3 m_radius2	187
5.18.4.4 m_slices	187
5.18.4.5 m_stacks	187
5.18.4.6 nbvertex	187
5.19 Référence de la structure vec2	187
5.19.1 Documentation des constructeurs et destructeur	188
5.19.1.1 vec2	188
5.19.1.2 vec2	188
5.19.1.3 vec2	188
5.19.1.4 vec2	188
5.19.1.5 vec2	189
5.19.2 Documentation des fonctions membres	189
5.19.2.1 length	189
5.19.2.2 normalize	189
5.19.2.3 operator()	190
5.19.2.4 operator()	190
5.19.2.5 operator*	190
5.19.2.6 operator*=	190
5.19.2.7 operator+	190
5.19.2.8 operator+=	190
5.19.2.9 operator-	190
5.19.2.10 operator-	191
5.19.2.11 operator-=	191
5.19.2.12 operator/	191
5.19.2.13 operator/=	191
5.19.2.14 operator=	191
5.19.2.15 operator[]	191
5.19.2.16 operator[]	191
5.19.3 Documentation des données membres	191
5.19.3.1 v	191
5.20 Référence de la structure vec3	191
5.20.1 Documentation des constructeurs et destructeur	192
5.20.1.1 vec3	192
5.20.1.2 vec3	192

5.20.1.3	vec3	192
5.20.1.4	vec3	193
5.20.1.5	vec3	193
5.20.2	Documentation des fonctions membres	193
5.20.2.1	length	193
5.20.2.2	normalize	193
5.20.2.3	operator()	194
5.20.2.4	operator()	194
5.20.2.5	operator*	194
5.20.2.6	operator*	194
5.20.2.7	operator*= <td>194</td>	194
5.20.2.8	operator+	194
5.20.2.9	operator+=	194
5.20.2.10	operator-	195
5.20.2.11	operator-	195
5.20.2.12	operator-=	195
5.20.2.13	operator/	195
5.20.2.14	operator/=	195
5.20.2.15	operator=	195
5.20.2.16	operator[]	195
5.20.2.17	operator[]	195
5.20.2.18	operator^	196
5.20.3	Documentation des données membres	196
5.20.3.1	v	196
5.21	Référence de la structure vec4	196
5.21.1	Documentation des constructeurs et destructeur	197
5.21.1.1	vec4	197
5.21.1.2	vec4	197
5.21.1.3	vec4	197
5.21.1.4	vec4	197
5.21.1.5	vec4	197
5.21.2	Documentation des fonctions membres	197
5.21.2.1	length	198
5.21.2.2	normalize	198
5.21.2.3	operator()	198
5.21.2.4	operator()	198
5.21.2.5	operator*	198
5.21.2.6	operator*= <td>198</td>	198
5.21.2.7	operator+	199
5.21.2.8	operator+=	199

5.21.2.9 operator-	199
5.21.2.10 operator-	199
5.21.2.11 operator-=	199
5.21.2.12 operator/	200
5.21.2.13 operator/=	200
5.21.2.14 operator[]	200
5.21.2.15 operator[]	200
5.21.3 Documentation des données membres	200
5.21.3.1 v	200
6 Documentation des fichiers	201
6.1 Référence du fichier camera.cpp	201
6.2 Référence du fichier camera.hpp	201
6.3 Référence du fichier gui/mainwindows.cpp	202
6.4 Référence du fichier gui/mainwindows.hpp	202
6.5 Référence du fichier gui/mondock.cpp	204
6.6 Référence du fichier gui/mondock.hpp	204
6.7 Référence du fichier gui/MyOpenGLWidget.cpp	205
6.8 Référence du fichier gui/MyOpenGLWidget.hpp	205
6.9 Référence du fichier helper.cpp	206
6.9.1 Documentation des fonctions	208
6.9.1.1 __openGL_check_error	208
6.9.1.2 currentMatrix	208
6.9.1.3 getActiveShader	209
6.9.1.4 operator<<	209
6.9.1.5 operator<<	209
6.9.1.6 operator<<	209
6.9.1.7 operator<<	209
6.9.1.8 operator<<	209
6.9.1.9 operator<<	209
6.9.1.10 operator<<	209
6.9.1.11 operator<<	209
6.9.1.12 operator<<	209
6.9.1.13 operator<<	209
6.9.1.14 popMatrix	210
6.9.1.15 print	210
6.9.1.16 print	210
6.9.1.17 print	210
6.9.1.18 print	210
6.9.1.19 print	210

6.9.1.20	projectionMatrix	210
6.9.1.21	pushMatrix	211
6.9.1.22	scaleMatrix	211
6.9.1.23	scaleMatrix	211
6.9.1.24	scales	211
6.9.1.25	setActiveShader	212
6.9.1.26	setModelMatrix	212
6.9.1.27	setProjectionMatrix	213
6.9.1.28	setViewMatrix	213
6.9.1.29	translate	213
6.9.1.30	translationMatrix	214
6.9.1.31	translationMatrix	214
6.9.1.32	viewMatrix	214
6.9.1.33	Xrotate	215
6.9.1.34	XrotationMatrix	216
6.9.1.35	Yrotate	216
6.9.1.36	YrotationMatrix	217
6.9.1.37	Zrotate	217
6.9.1.38	ZrotationMatrix	218
6.9.2	Documentation des variables	218
6.9.2.1	activeShader	218
6.9.2.2	currentModelMatrix	218
6.9.2.3	currentProjectionMatrix	218
6.9.2.4	currentViewMatrix	218
6.9.2.5	matrix_stack	218
6.10	Référence du fichier helper.hpp	218
6.10.1	Documentation des macros	220
6.10.1.1	M_PI	220
6.10.1.2	openGL_check_error	220
6.10.1.3	to_rad	220
6.10.2	Documentation des définitions de type	220
6.10.2.1	mat3	220
6.10.2.2	mat4	220
6.10.2.3	vec2	220
6.10.2.4	vec3	220
6.10.2.5	vec4	220
6.10.3	Documentation des fonctions	220
6.10.3.1	__openGL_check_error	220
6.10.3.2	currentMatrix	221
6.10.3.3	getActiveShader	221

6.10.3.4 operator<<	221
6.10.3.5 operator<<	221
6.10.3.6 operator<<	222
6.10.3.7 operator<<	222
6.10.3.8 operator<<	222
6.10.3.9 operator<<	222
6.10.3.10 operator<<	222
6.10.3.11 operator<<	222
6.10.3.12 operator<<	222
6.10.3.13 operator<<	222
6.10.3.14 popMatrix	222
6.10.3.15 print	222
6.10.3.16 print	222
6.10.3.17 print	222
6.10.3.18 print	222
6.10.3.19 print	222
6.10.3.20 projectionMatrix	223
6.10.3.21 pushMatrix	223
6.10.3.22 scaleMatrix	223
6.10.3.23 scaleMatrix	224
6.10.3.24 scales	224
6.10.3.25 setActiveShader	224
6.10.3.26 setModelMatrix	225
6.10.3.27 setProjectionMatrix	225
6.10.3.28 setViewMatrix	225
6.10.3.29 translate	226
6.10.3.30 translationMatrix	226
6.10.3.31 translationMatrix	226
6.10.3.32 viewMatrix	227
6.10.3.33 Xrotate	227
6.10.3.34 XrotationMatrix	228
6.10.3.35 Yrotate	228
6.10.3.36 YrotationMatrix	229
6.10.3.37 Zrotate	229
6.10.3.38 ZrotationMatrix	230
6.11 Référence du fichier main.cpp	230
6.11.1 Documentation des fonctions	231
6.11.1.1 main	231
6.12 Référence du fichier objets/cube.cpp	231
6.13 Référence du fichier objets/cube.hpp	232

6.14 Référence du fichier objets/donuts.cpp	232
6.15 Référence du fichier objets/donuts.hpp	233
6.16 Référence du fichier objets/mesh.cpp	234
6.17 Référence du fichier objets/mesh.hpp	234
6.18 Référence du fichier objets/node.cpp	235
6.19 Référence du fichier objets/node.hpp	236
6.20 Référence du fichier objets/objet.cpp	237
6.21 Référence du fichier objets/objet.hpp	237
6.22 Référence du fichier objets/plan.cpp	238
6.23 Référence du fichier objets/plan.hpp	239
6.24 Référence du fichier objets/sphere.cpp	240
6.25 Référence du fichier objets/sphere.hpp	241
6.26 Référence du fichier scene/light.cpp	241
6.27 Référence du fichier scene/light.hpp	242
6.28 Référence du fichier scene/material.cpp	243
6.28.1 Documentation des fonctions	243
6.28.1.1 operator<<	243
6.29 Référence du fichier scene/material.hpp	244
6.29.1 Documentation des fonctions	244
6.29.1.1 operator<<	245
6.30 Référence du fichier scene/piece.cpp	245
6.31 Référence du fichier scene/piece.hpp	245
6.32 Référence du fichier scene/scene.cpp	246
6.33 Référence du fichier scene/scene.hpp	247
6.33.1 Documentation des macros	248
6.33.1.1 MAX_LIGHT	248
Index	249

Chapitre 1

Liste des choses à faire

Classe **Light**

atténuation de la lumière par lumière et plus globale

Membre **Light** : `~Light()`

Vérifier que la lumière est désactivée dans tous les shader

Membre **Material** : `~Material()`

Gérer libération de la texture quand plus utilisée

Membre **Piece** : `:getChildren() const`

A revoir !

Renvoie

liste des noms des objets

Chapitre 2

Index hiérarchique

2.1 Hiérarchie des classes

Cette liste d'héritage est classée approximativement par ordre alphabétique :

Camera	9
mat3	56
mat4	59
Mesh : :MeshInfo	78
QDockWidget	
Mondock	79
QGLWidget	
MyOpenGLWidget	119
QMainWindow	
MainWindow	31
QOpenGLFunctions_3_2_Core	
Light	23
Material	63
MyOpenGLWidget	119
Objet	136
Cube	15
Donuts	19
Mesh	73
Node	126
Piece	151
Plan	157
Sphere	183
Scene	161
vec2	187
vec3	191
vec4	196

Chapitre 3

Index des classes

3.1 Liste des classes

Liste des classes, structures, unions et interfaces avec une brève description :

Camera	
Camera pour Scene	9
Cube	
Primitive cube	15
Donuts	
Primitive tore	19
Light	
Lumière pour la Scene	23
MainWindow	
mat3	31
mat4	56
Material	59
Matériaux pour la Scene	63
Mesh	
Charge un modèle 3D	73
Mesh : :MeshInfo	78
Mondock	79
MyOpenGLWidget	119
Node	
Contient un node de modèle 3D	126
Objet	
Classe de base affichable	136
Piece	
Pièce, contient un ensemble d'objet qu'elle contient	151
Plan	
Plan utilisé par la Piece	157
Scene	
Classe principale	161
Sphere	
Primitive tore	183
vec2	
vec3	187
vec4	191
	196

Chapitre 4

Index des fichiers

4.1 Liste des fichiers

Liste de tous les fichiers avec une brève description :

camera.cpp	201
camera.hpp	201
helper.cpp	206
helper.hpp	218
main.cpp	230
gui/mainwindows.cpp	202
gui/mainwindows.hpp	202
gui/mondock.cpp	204
gui/mondock.hpp	204
gui/MyOpenGLWidget.cpp	205
gui/MyOpenGLWidget.hpp	205
objets/cube.cpp	231
objets/cube.hpp	232
objets/donuts.cpp	232
objets/donuts.hpp	233
objets/mesh.cpp	234
objets/mesh.hpp	234
objets/node.cpp	235
objets/node.hpp	236
objets/objet.cpp	237
objets/objet.hpp	237
objets/plan.cpp	238
objets/plan.hpp	239
objets/sphere.cpp	240
objets/sphere.hpp	241
scene/light.cpp	241
scene/light.hpp	242
scene/material.cpp	243
scene/material.hpp	244
scene/piece.cpp	245
scene/piece.hpp	245
scene/scene.cpp	246
scene/scene.hpp	247

Chapitre 5

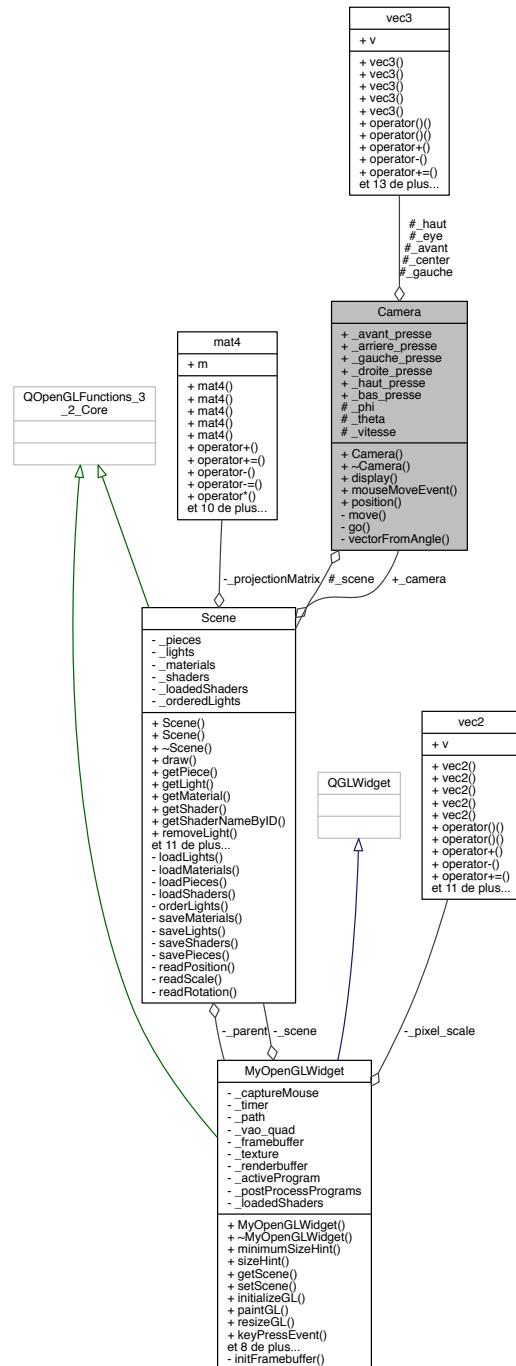
Documentation des classes

5.1 Référence de la classe Camera

Camera pour Scene.

```
#include <camera.hpp>
```

Graphe de collaboration de Camera :



Fonctions membres publiques

- **Camera** (*Scene* *scene=NULL, float eyeX=0.0f, float eyeY=0.0f, float eyeZ=0.0f)
Constructeur.
 - **~Camera** ()
Destructeur.
 - void **display** ()
Actualise la caméra.
 - void **mouseMoveEvent** (int x, int y, int width, int height)

- Fonction de déplacement de la caméra à la souris.
- const `vec3 & position () const`
Retourne la position de la caméra.

Attributs publics

- bool `_avant_presse`
- bool `_arriere_presse`
- bool `_gauche_presse`
- bool `_droite_presse`
- bool `_haut_presse`
- bool `_bas_presse`

Attributs protégés

- `vec3 _eye`
- `vec3 _center`
- float `_phi`
- float `_theta`
- `vec3 _avant`
- `vec3 _gauche`
- `vec3 _haut`
- float `_vitesse`
- Scene * `_scene`

Fonctions membres privées

- void `move ()`
Calcul le déplacement de la Camera.
- void `go (float x, float y, float z)`
Déplace la caméra aux coordonnées données.
- void `vectorFromAngle ()`
Recalcule les vecteurs avant/côté pour le mouvement.

5.1.1 Description détaillée

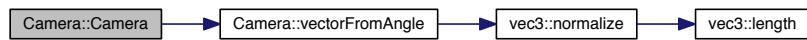
Camera pour Scene.

5.1.2 Documentation des constructeurs et destructeur

5.1.2.1 Camera :: Camera (Scene * scene = NULL, float eyeX = 0.0f, float eyeY = 0.0f, float eyeZ = 0.0f)

Constructeur.

Voici le graphe d'appel pour cette fonction :



5.1.2.2 Camera ::~Camera () [inline]

Destructeur.

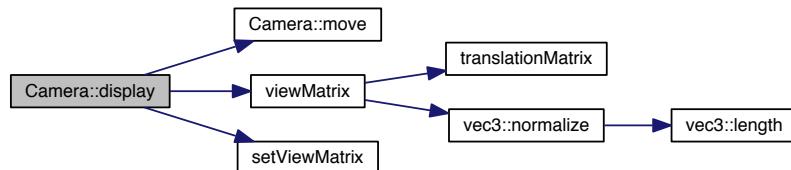
5.1.3 Documentation des fonctions membres

5.1.3.1 void Camera ::display()

Actualise la caméra.

Met à jour la position de la caméra en fonction des booléens de déplacement et actualise la matrice view

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



5.1.3.2 void Camera ::go(float x, float y, float z) [private]

Déplace la caméra aux coordonnées données.

5.1.3.3 void Camera ::mouseMoveEvent(int x, int y, int width, int height)

Fonction de déplacement de la caméra à la souris.

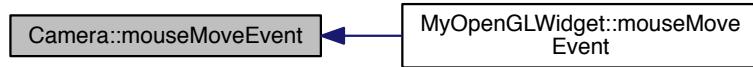
Paramètres

<i>x</i>	position en X du curseur de la souris
<i>y</i>	position en Y du curseur de la souris
<i>width</i>	largeur de la fenêtre
<i>height</i>	hauteur de la fenêtre

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



5.1.3.4 void Camera ::move() [private]

Calcul le déplacement de la [Camera](#).

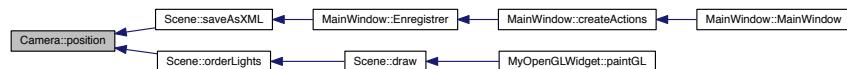
Voici le graphe des appelants de cette fonction :



5.1.3.5 const vec3& Camera ::position() const [inline]

Retourne la position de la caméra.

Voici le graphe des appelants de cette fonction :



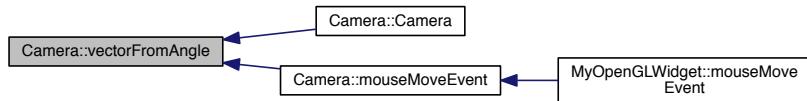
5.1.3.6 void Camera ::vectorFromAngle() [private]

Recalcule les vecteurs avant/côté pour le mouvement.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



5.1.4 Documentation des données membres

5.1.4.1 bool Camera ::_ARRIERE_PRESSE

Indique si la caméra doit reculer au prochain affichage

5.1.4.2 vec3 Camera ::_AVANT [protected]

5.1.4.3 bool Camera ::_AVANT_PRESSE

Indique si la caméra doit avancer au prochain affichage

5.1.4.4 bool Camera ::_BAS_PRESSE

Indique si la caméra doit descendre au prochain affichage

5.1.4.5 vec3 Camera ::_CENTRE [protected]

5.1.4.6 bool Camera ::_DROITE_PRESSE

Indique si la caméra doit aller à droite au prochain affichage

5.1.4.7 vec3 Camera ::_EYE [protected]

5.1.4.8 vec3 Camera ::_GAUCHE [protected]

5.1.4.9 bool Camera ::_GAUCHE_PRESSE

Indique si la caméra doit aller à gauche au prochain affichage

5.1.4.10 vec3 Camera ::_HAUT [protected]

5.1.4.11 bool Camera ::_HAUT_PRESSE

Indique si la caméra doit monter au prochain affichage

5.1.4.12 float Camera ::_PHI [protected]

5.1.4.13 Scene* Camera ::_SCENE [protected]

5.1.4.14 float Camera::_theta [protected]

5.1.4.15 float Camera::_vitesse [protected]

La documentation de cette classe a été générée à partir des fichiers suivants :

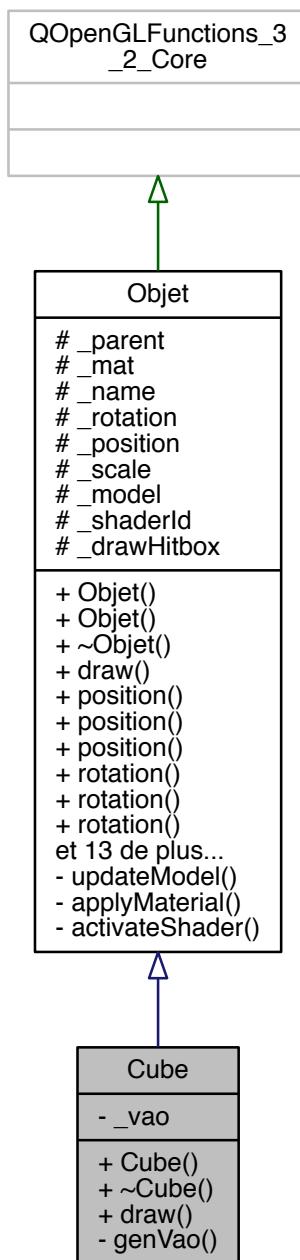
- [camera.hpp](#)
- [camera.cpp](#)

5.2 Référence de la classe Cube

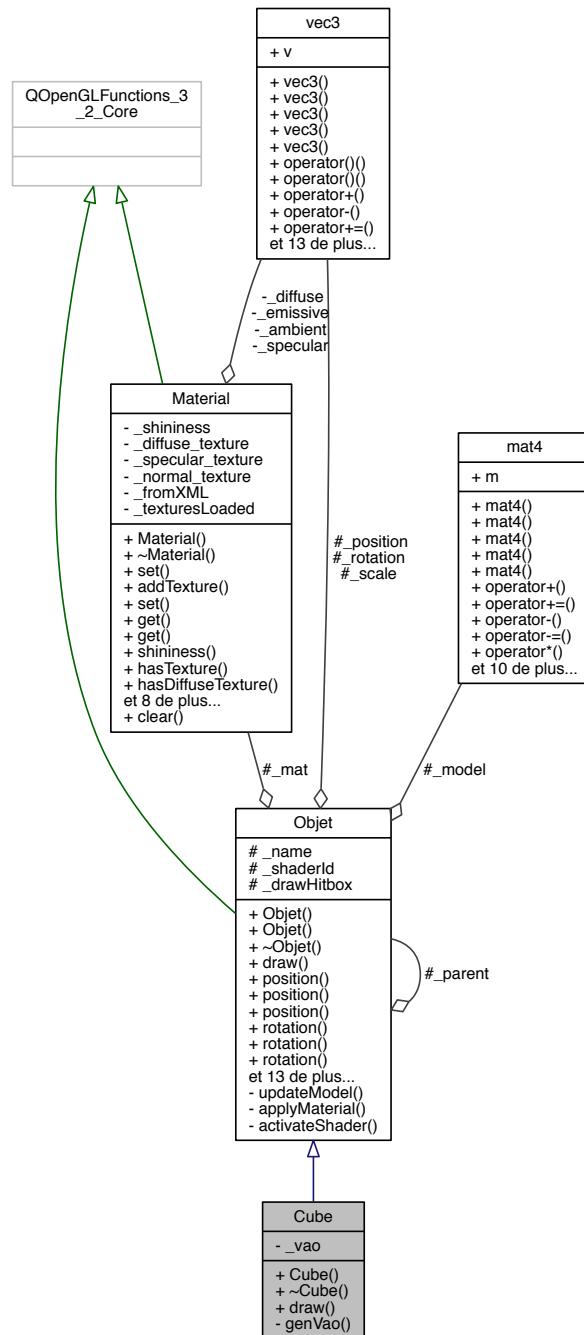
Primitive cube.

```
#include <cube.hpp>
```

Graphe d'héritage de Cube :



Graphe de collaboration de Cube :



Fonctions membres publiques

- `Cube (Material *mat=NULL, vec3 rotation=vec3(), vec3 position=vec3())`
Constructeur du cube.
- `~Cube ()`
Destructeur du cube.
- `void draw ()`
Affiche le cube.

Fonctions membres privées

- void [genVao \(\)](#)
Génère les vbo et le vao pour l'affichage du cube.

Attributs privés statiques

- static GLuint [_vao](#) = 0

Membres hérités additionnels

5.2.1 Description détaillée

Primitive cube.

Avertissement

Peut être plus utilisable

5.2.2 Documentation des constructeurs et destructeur

5.2.2.1 Cube : :Cube (Material * mat = NULL, vec3 rotation = vec3 (), vec3 position = vec3 ())

Constructeur du cube.

Génère le vao et les vbo s'ils n'ont pas déjà été générés

Voici le graphe d'appel pour cette fonction :



5.2.2.2 Cube : :~Cube ()

Destructeur du cube.

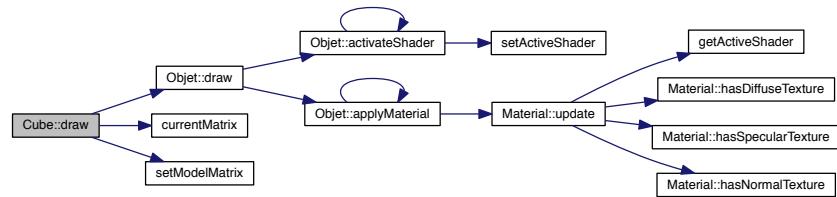
5.2.3 Documentation des fonctions membres

5.2.3.1 void Cube : :draw () [virtual]

Affiche le cube.

Réimplémentée à partir de [Objet](#).

Voici le graphe d'appel pour cette fonction :



5.2.3.2 void Cube ::genVao() [private]

Génère les vbo et le vao pour l'affichage du cube.

Voici le graphe des appels de cette fonction :



5.2.4 Documentation des données membres

5.2.4.1 GLuint Cube ::_vao = 0 [static], [private]

< Id du Vao du cube

La documentation de cette classe a été générée à partir des fichiers suivants :

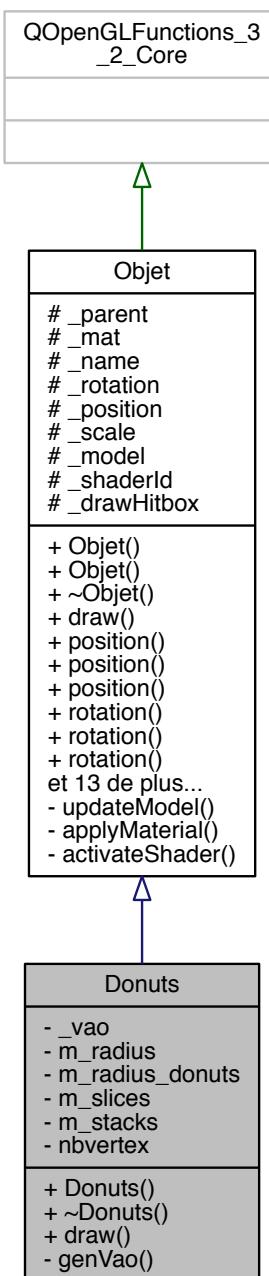
- objets/cube.hpp
- objets/cube.cpp

5.3 Référence de la classe Donuts

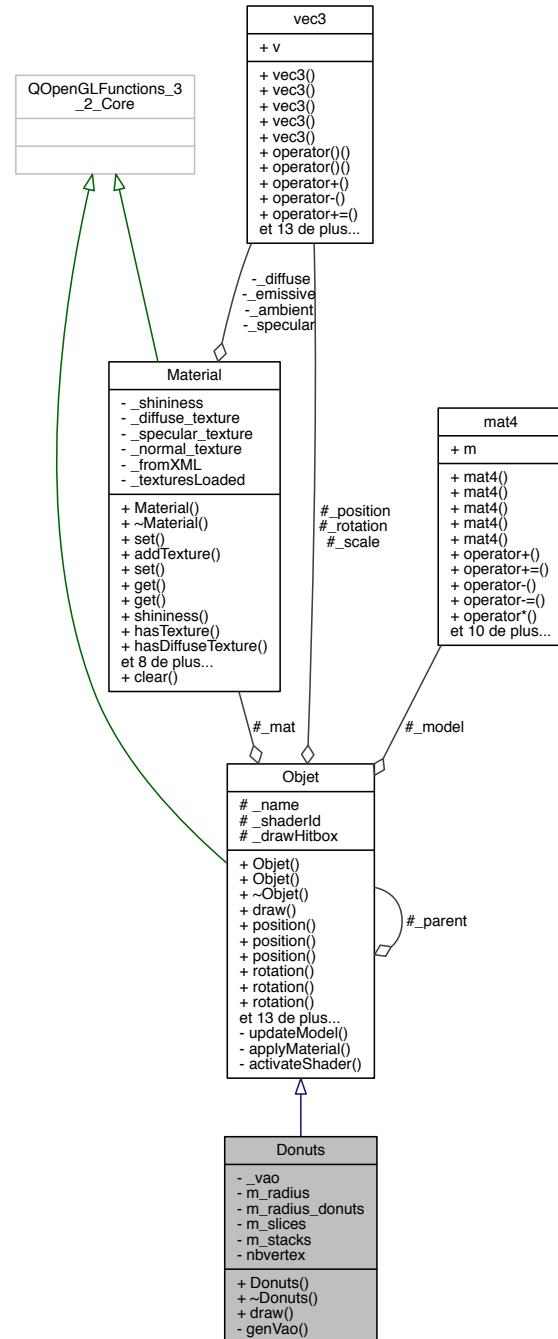
Primitive tore.

```
#include <donuts.hpp>
```

Graphe d'héritage de Donuts :



Graphe de collaboration de Donuts :



Fonctions membres publiques

- **Donuts** (GLdouble `m_radius=1`, GLdouble `m_radius_donuts=1`, GLint `m_slices=1`, GLint `m_stacks=1`, Material *`mat=NULL`, vec3 `rotation=vec3()`, vec3 `position=vec3()`)
- **~Donuts ()**
- void **draw ()**

Affichage de l'*Objet*.

Fonctions membres privées

— void [genVao \(\)](#)

Attributs privés

- GLuint [_vao](#)
- GLdouble [m_radius](#)
- GLdouble [m_radius_donuts](#)
- GLint [m_slices](#)
- GLint [m_stacks](#)
- GLsizei [nbvertex](#)

Membres hérités additionnels

5.3.1 Description détaillée

Primitive tore.

Avertissement

Peut être plus utilisable

Voir également

[Cube](#)

5.3.2 Documentation des constructeurs et destructeur

5.3.2.1 `Donuts ::Donuts (GLdouble m_radius = 1, GLdouble m_radius_donuts = 1, GLint m_slices = 1, GLint m_stacks = 1, Material * mat = NULL, vec3 rotation = vec3 (), vec3 position = vec3 ())`

Voici le graphe d'appel pour cette fonction :



5.3.2.2 `Donuts ::~Donuts ()`

5.3.3 Documentation des fonctions membres

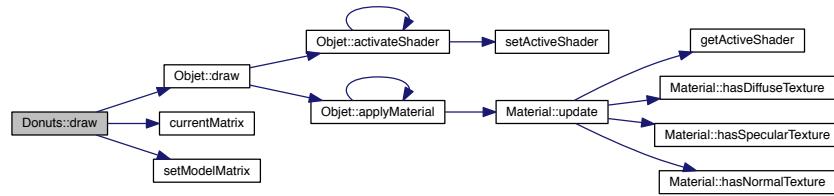
5.3.3.1 void `Donuts ::draw () [virtual]`

Affichage de l'[Objet](#).

Active le shader de l'[Objet](#) et applique le material

Réimplémentée à partir de [Objet](#).

Voici le graphe d'appel pour cette fonction :



5.3.3.2 void Donuts::genVao() [private]

Voici le graphe des appelants de cette fonction :



5.3.4 Documentation des données membres

5.3.4.1 GLuint Donuts::_vao [private]

5.3.4.2 GLdouble Donuts::_m_radius [private]

5.3.4.3 GLdouble Donuts::_m_radius_donuts [private]

5.3.4.4 GLint Donuts::_m_slices [private]

5.3.4.5 GLint Donuts::_m_stacks [private]

5.3.4.6 GLsizei Donuts::_nbvertex [private]

La documentation de cette classe a été générée à partir des fichiers suivants :

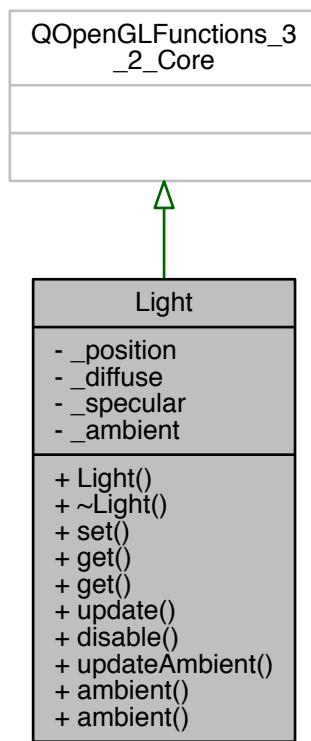
- objets/donuts.hpp
- objets/donuts.cpp

5.4 Référence de la classe Light

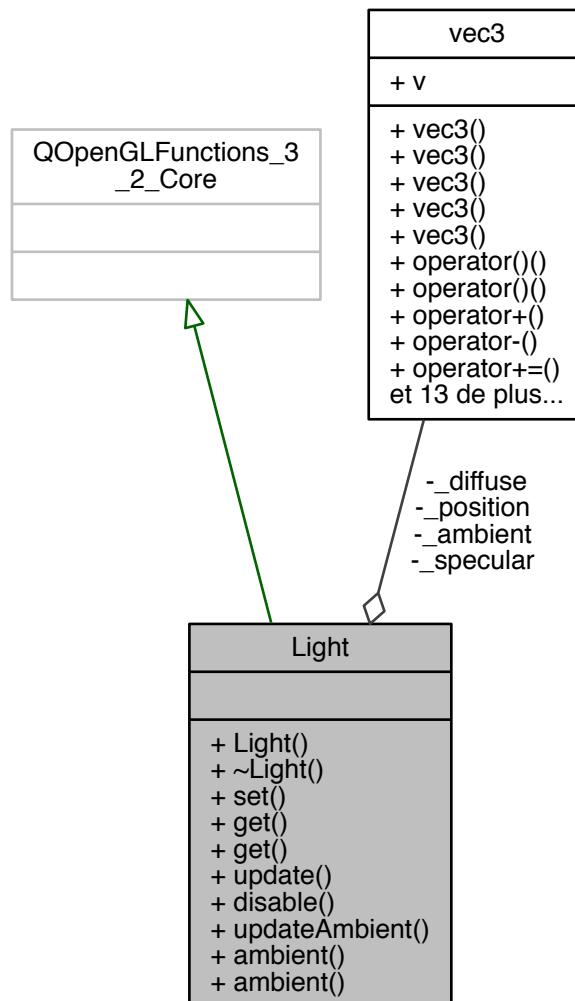
Lumière pour la [Scene](#).

```
#include <light.hpp>
```

Graphe d'héritage de Light :



Graphe de collaboration de Light :



Fonctions membres publiques

- `Light (vec3 position=vec3(0.0, 0.0, 0.0), vec3 diffuse=vec3(0.8, 0.8, 0.8), vec3 specular=vec3(0.8, 0.8, 0.8))`
Constructeur d'une lumière.
- `~Light ()`
Destructeur.
- `void set (GLenum type, vec3 value)`
Met à jour une composante de la lumière.
- `vec3 get (GLenum type) const`
Récupère la valeur d'une composante de la lumière.
- `vec3 & get (GLenum type)`
Récupère la valeur d'une composante de la lumière.
- `void update (unsigned char number)`
Met à jour le shader avec les infos de la lumière.

Fonctions membres publiques statiques

- static void **disable** (unsigned char number)
Désactive la lumière dans le shader.
- static void **updateAmbient** ()
Met à jour la lumière ambient.
- static void **ambient** (const **vec3** &v)
Modifie la composante ambient de la lumière.
- static **vec3** & **ambient** ()
retourne le vecteur des composantes ambient

Attributs privés

- **vec3 _position**
- **vec3 _diffuse**
- **vec3 _specular**

Attributs privés statiques

- static **vec3 _ambient**

5.4.1 Description détaillée

Lumière pour la [Scene](#).

Gère une lumière pour la scène

Avertissement

seul les lumières positionnelles sont gérées pour l'instant

A faire attenuation de la lumière par lumière et plus globale

5.4.2 Documentation des constructeurs et destructeur

5.4.2.1 Light ::Light (**vec3 position = vec3(0.0, 0.0, 0.0)**, **vec3 diffuse = vec3(0.8, 0.8, 0.8)**, **vec3 specular = vec3(0.8, 0.8, 0.8)**)

Constructeur d'une lumière.

Paramètres

position	position de la lumière
diffuse	composante diffuse de la lumière
specular	composante spéculaire de la lumière
number	correspond au slot de la lumière qui sera utilisé dans le shader

Avertissement

number doit appartenir à [0,7]

5.4.2.2 Light ::~Light ()

Destructeur.

A faire Vérifier que la lumière est désactivée dans tout les shader

5.4.3 Documentation des fonctions membres

5.4.3.1 static void Light::ambient(const vec3 & v) [inline], [static]

Modifie la composante ambient de la lumière.

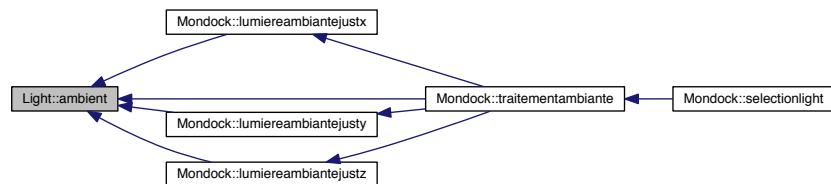
Paramètres

v	nouvelle composante
---	---------------------

5.4.3.2 static vec3& Light::ambient() [inline], [static]

retourne le vecteur des composantes ambient

Voici le graphe des appelants de cette fonction :



5.4.3.3 void Light::disable(unsigned char number) [static]

Désactive la lumière dans le shader.

Paramètres

number	numéro de la lumière à désactiver
--------	-----------------------------------

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



5.4.3.4 vec3 Light::get(GLenum type) const

Récupère la valeur d'une composante de la lumière.

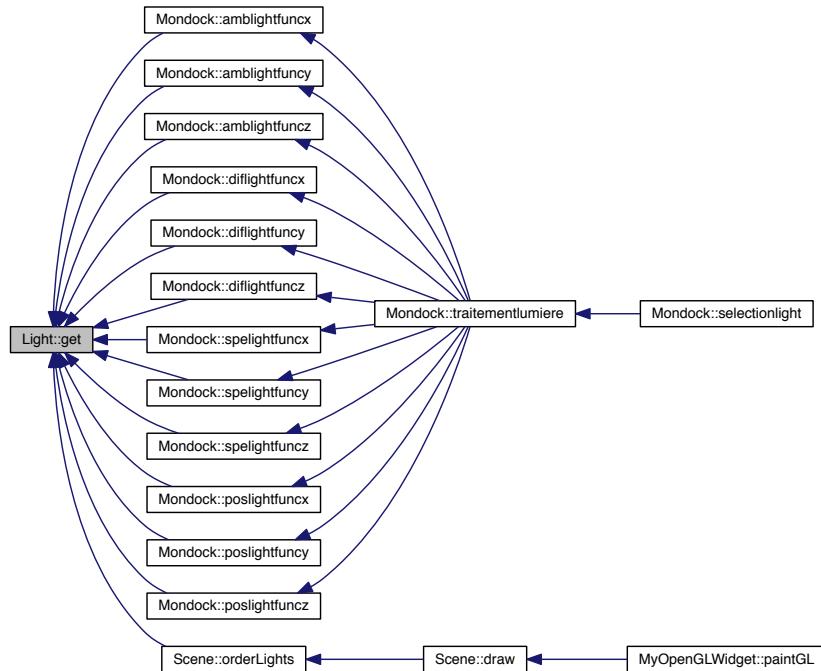
Paramètres

<code>type</code>	composante voulu, doit être parmis (GL_POSITION, GL_AMBIENT, GL_DIFFUSE, GL_SPECULAR)
-------------------	---

Renvoie

valeur

Voici le graphe des appels de cette fonction :

**5.4.3.5 vec3 & Light::get(GLenum type)**

Récupère la valeur d'une composante de la lumière.

Paramètres

<code>type</code>	composante voulu, doit être parmis (GL_POSITION, GL_AMBIENT, GL_DIFFUSE, GL_SPECULAR)
-------------------	---

Renvoie

valeur

Avertissement

si le type n'est pas dans les 4 gérés résultat imprévisible

5.4.3.6 void Light::set(GLenum type, vec3 value)

Met à jour une composante de la lumière.

Paramètres

<i>type</i>	composante à modifier, doit être parmis (GL_POSITION, GL_AMBIENT, GL_DIFFUSE, GL_SPECULAR)
<i>value</i>	valeur

Voici le graphe des appelants de cette fonction :

**5.4.3.7 void Light ::update (unsigned char number)**

Met à jour le shader avec les infos de la lumière.

Paramètres

<i>number</i>	slot de lumière à utiliser dans la shader
---------------	---

Voici le graphe d'appel pour cette fonction :

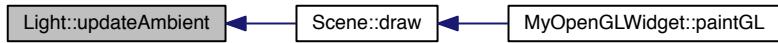
**5.4.3.8 void Light ::updateAmbient () [static]**

Met à jour la lumière ambient.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appels de cette fonction :



5.4.4 Documentation des données membres

5.4.4.1 `vec3 Light::_ambient [static], [private]`

La composante ambient est commune à toutes les lumières

5.4.4.2 `vec3 Light::_diffuse [private]`

5.4.4.3 `vec3 Light::_position [private]`

5.4.4.4 `vec3 Light::_specular [private]`

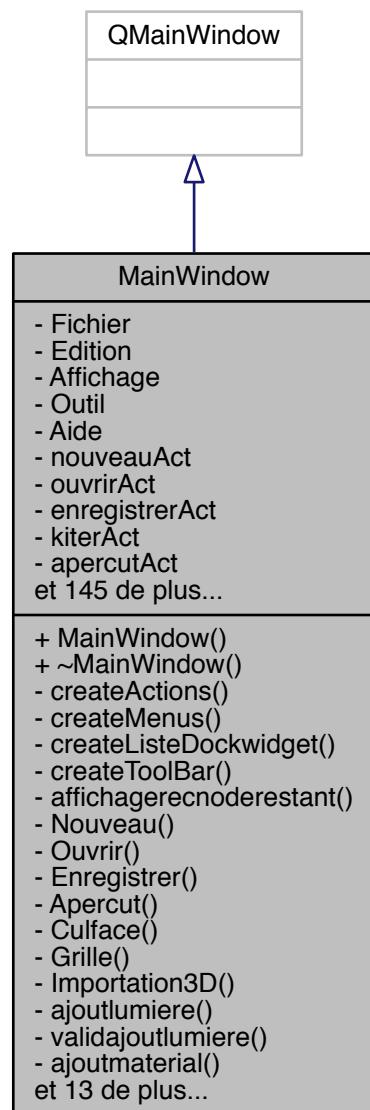
La documentation de cette classe a été générée à partir des fichiers suivants :

- [scene/light.hpp](#)
- [scene/light.cpp](#)

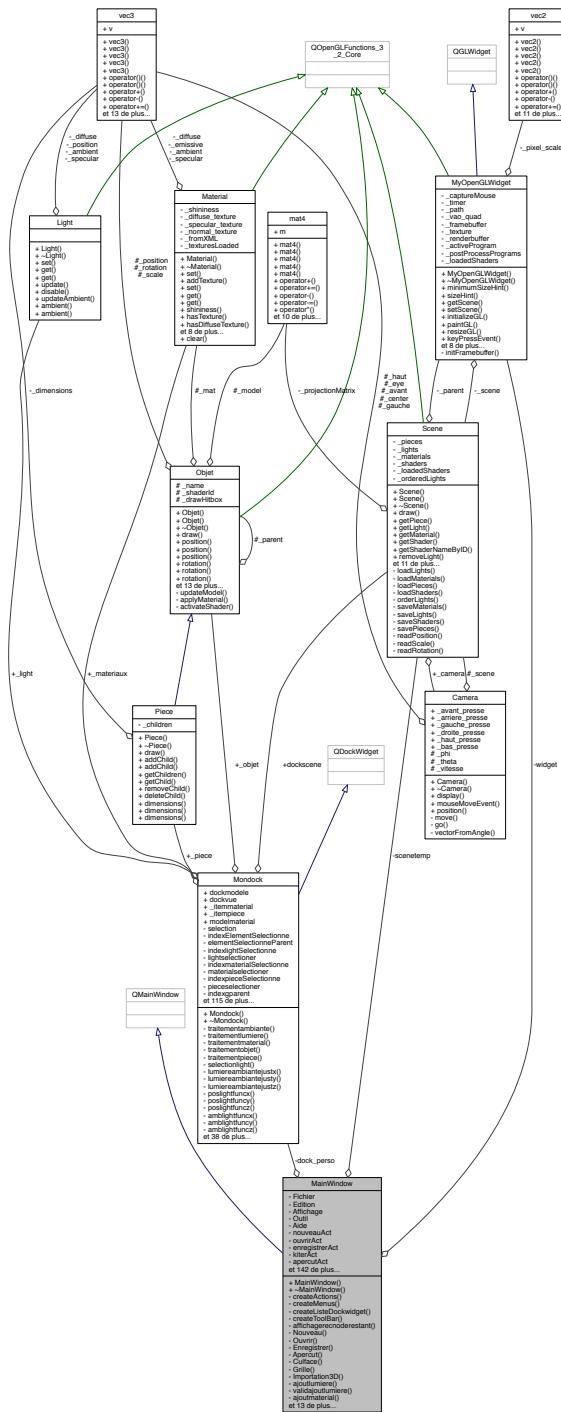
5.5 Référence de la classe MainWindow

```
#include <mainwindows.hpp>
```

Graphe d'héritage de MainWindow :



Graphe de collaboration de MainWindow :



Fonctions membres publiques

- [MainWindow \(const QString &path=""\)](#)
- [~MainWindow \(\)](#)

Connecteurs privés

- void [Nouveau \(\)](#)

```

— void Ouvrir ()
— void Enregistrer ()
— void Apercut ()
— void Culface ()
— void Grille ()
— void Importation3D ()
— void ajoutlumiere ()
— void validajoutlumiere ()
— void ajoutmaterial ()
— void validajoutmaterial ()
— void ajoutpiece ()
— void validajoutpiece ()
— void ajoutmur ()
— void ajoutmurcreatemodelmurcombo ()
— void validajoutmur ()
— void ajoutobjet ()
— void selectioncheminobj ()
— void validationajoutobjet ()
— void aboutassimp ()
— void supresonelement ()
— void reloadshader ()
— void changeshader (const QString &text)

```

Fonctions membres privées

```

— void createActions ()
— void createMenus ()
— void createListeDockwidget ()
— void createToolBar ()
— void affichagerecnoderestant (Node *a, QStandardItem *b)

```

Attributs privés

```

— QMenu * Fichier
— QMenu * Edition
— QMenu * Affichage
— QMenu * Outil
— QMenu * Aide
— QAction * nouveauAct
— QAction * ouvrirAct
— QAction * enregisterAct
— QAction * kiterAct
— QAction * apercutAct
— QAction * culfaceAct
— QAction * grilleAct
— QAction * importation3DAct
— QMenu * ajoutelement
— QAction * ajoutlumierAct
— QAction * ajoutmaterialAct
— QAction * ajoutpieceAct
— QAction * ajoutmurAct
— QAction * ajoutobjetAct
— QDockWidget * dockajoutlumiere
— QWidget * widgetajoutlumiere
— QPushButton * boutonajoutlumiere
— QLineEdit * lineeditnomajoutlumiere
— QLabel * nomajoutlumiere
— QDoubleSpinBox * spinpositionajoutlumierex
— QDoubleSpinBox * spinpositionajoutlumierey
— QDoubleSpinBox * spinpositionajoutlumierez
— QLabel * positionajoutlumiere
— QDoubleSpinBox * spindifajoutlumierex
— QDoubleSpinBox * spindifajoutlumierey
— QDoubleSpinBox * spindifajoutlumierez
— QLabel * difajoutlumiere
— QDoubleSpinBox * spinspeajoutlumierex
— QDoubleSpinBox * spinspeajoutlumierey
— QDoubleSpinBox * spinspeajoutlumierez
— QLabel * speajoutlumiere
— QHBoxLayout * layoutnomajoutlumiere
— QHBoxLayout * layoutposajoutlumiere
— QHBoxLayout * layoutdifajoutlumiere
— QHBoxLayout * layoutspeajoutlumiere
— QVBoxLayout * layoutajoutlumiere

```

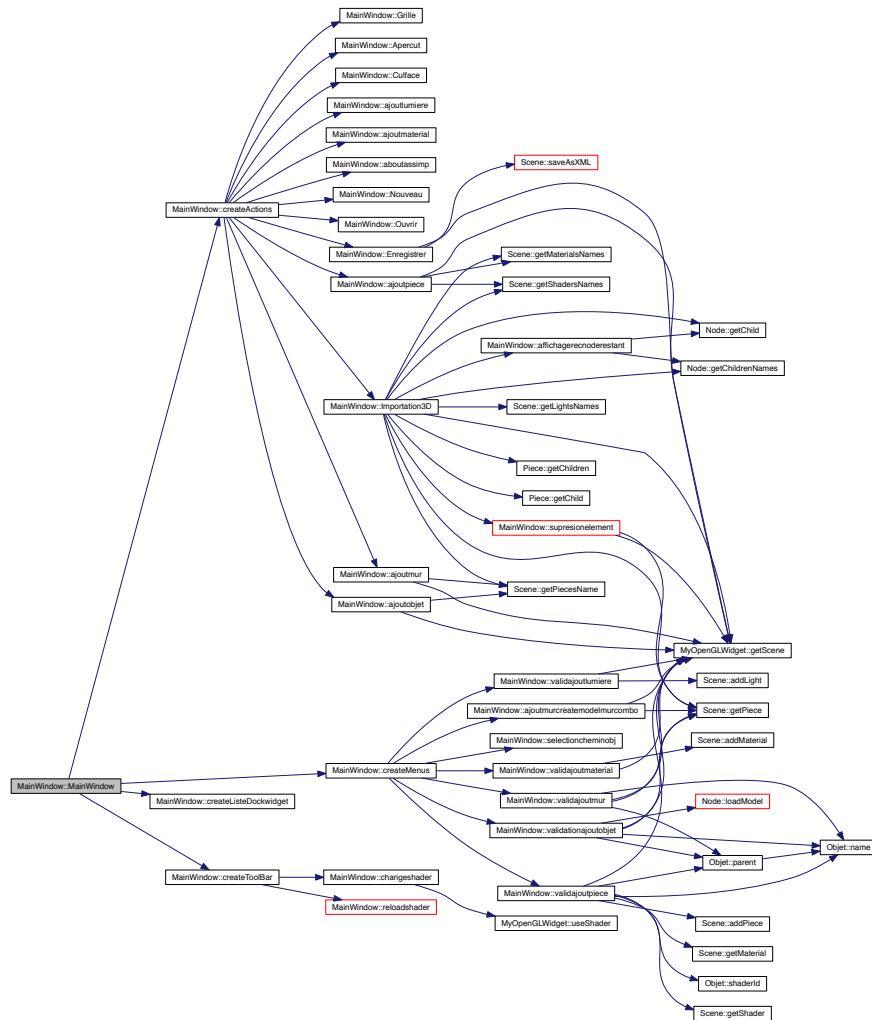
```
— QDockWidget * dockajoutmateriaux
— QWidget * widgetajoutmaterial
— QPushButton * boutonajoutmaterial
— QLineEdit * lineditnomajoutmaterial
— QLabel * nomajoutmaterial
— QDoubleSpinBox * spinambajoutmaterialx
— QDoubleSpinBox * spinambajoutmaterialy
— QDoubleSpinBox * spinambajoutmaterialz
— QLabel * ambajoutmaterial
— QDoubleSpinBox * spindifajoutmaterialx
— QDoubleSpinBox * spindifajoutmaterialy
— QDoubleSpinBox * spindifajoutmaterialz
— QLabel * difajoutmaterial
— QDoubleSpinBox * spinspeajoutmaterialx
— QDoubleSpinBox * spinspeajoutmaterialy
— QDoubleSpinBox * spinspeajoutmaterialz
— QDoubleSpinBox * spinspeajoutmateriala
— QLabel * speajoutmaterial
— QHBoxLayout * layoutnomajoutmaterial
— QHBoxLayout * layoutambajoutmaterial
— QHBoxLayout * layoutdifajoutmaterial
— QHBoxLayout * layoutspeajoutmaterial
— QVBoxLayout * layoutajoutmaterial
— QDockWidget * dockajoutpiece
— QWidget * widgetajoutpiece
— QPushButton * boutonajoutpiece
— QSpinBox * ajoutpiecedimx
— QSpinBox * ajoutpiecedimy
— QSpinBox * ajoutpiecedimz
— QComboBox * comboajoutpiece
— QStandardItemModel * modelelemateriaupiece
— QComboBox * comboajoutpieceshader
— QStandardItemModel * modelepieceeshader
— QLabel * labelcomboajoutpiece
— QLabel * labeledimajoutpiece
— QLabel * labelnomajoutpiece
— QLabel * labelmurajoutpiece
— QLabel * labelajoutpiececshader
— QLabel * labelmur1
— QLabel * labelmur2
— QLabel * labelmur3
— QLabel * labelmur4
— QLabel * labelmur5
— QLabel * labelmur6
— QCheckBox * checkmur1
— QCheckBox * checkmur2
— QCheckBox * checkmur3
— QCheckBox * checkmur4
— QCheckBox * checkmur5
— QCheckBox * checkmur6
— QLineEdit * lineditajoutpiece
— QVBoxLayout * mainlayoutajoutpiece
— QHBoxLayout * ajoutpiececelayoutdim
— QHBoxLayout * ajoutpiececelayoutnom
— QHBoxLayout * ajoutpiececelayoutcombo
— QHBoxLayout * ajoutpiececelabelmure
— QHBoxLayout * ajoutpiecepiececer1
— QHBoxLayout * ajoutpiecepiececer2
— QHBoxLayout * ajoutpiecepiececer3
— QHBoxLayout * ajoutpiececshaderlayout
— QDockWidget * dockajoutmur
— QWidget * widgetajoutmur
— QVBoxLayout * mainlayoutajoutmur
— QHBoxLayout * layoutcombotnompieceajoutmur
— QHBoxLayout * layoutcombotnommurajoutmur
— QComboBox * combopieceajoutmur
— QStandardItemModel * modelajoutmurpiece
— QLabel * labelpiececomboajoutmur
— QPushButton * boutonpieceajoutmur
— QComboBox * combomurajoutmur
— QStandardItemModel * modelajoutmurmur
— QLabel * labelmurcomboajoutmur
— QPushButton * boutonajoutmur
— QDockWidget * dockajoutobjet
— QWidget * widgetajoutobjet
— QVBoxLayout * mainlayoutajoutobjet
— QHBoxLayout * layoutajoutobjetnom
```

```
— QBoxLayout * layoutajoutobjetpiece
— QBoxLayout * layoutcheminajoutobjet
— QLineEdit * champnomajoutobjet
— QLabel * labelnomajoutobjet
— QComboBox * combopieceajoutobjet
— QStandardItemModel * modelajoutobjet
— QLabel * labelpieceajoutobjet
— QLabel * labelcheminobjajoutobj
— QLineEdit * lineeditobjajoutobj
— QPushButton * boutonselectionobjajoutobjet
— QPushButton * boutonajoutobjet
— QMenu * about
— QAction * aboutqtAct
— QAction * aboutassimpAct
— QToolBar * fileToolBar
— QComboBox * shaderpospro
— QStandardItemModel * modelshaderpostpro
— QAction * reloadshaderAct
— QDockWidget * dock_list_elements
— QPushButton * boutonlisteelement
— QPushButton * boutonlisteelementdelete
— Scene * scenetemp
— QStringList listtempobjet
— QStringList listtemplight
— QStringList listtempmaterial
— QStringList listtemshader
— QStandardItemModel * modele
— QStandardItem * light
— QStandardItem * material
— QStandardItem * objet
— QStandardItem * item
— QStandardItem * shader
— QTreeView * vue
— QVBoxLayout * layoutlistescene
— QWidget * widgetdocklistscape
— Mondock * dock_perso
— MyOpenGLWidget * widget
```

5.5.1 Documentation des constructeurs et destructeur

5.5.1.1 MainWindow : :MainWindow (const QString & path = " ")

Voici le graphe d'appel pour cette fonction :



5.5.1.2 MainWindow : :~MainWindow ()

5.5.2 Documentation des fonctions membres

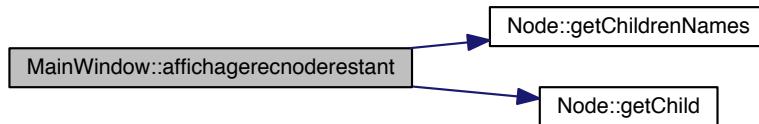
5.5.2.1 void MainWindow : :aboutassimp () [private], [slot]

Voici le graphe des appels de cette fonction :

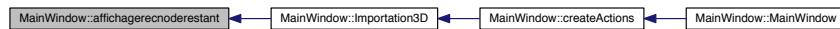


5.5.2.2 void MainWindow ::affichagerecnoderestant (Node * a, QStandardItem * b) [private]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



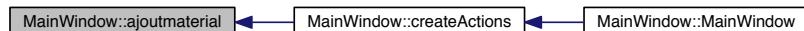
5.5.2.3 void MainWindow ::ajoutlumiere () [private], [slot]

Voici le graphe des appelants de cette fonction :



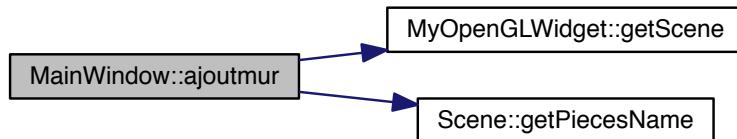
5.5.2.4 void MainWindow ::ajoutmaterial () [private], [slot]

Voici le graphe des appelants de cette fonction :



5.5.2.5 void MainWindow ::ajoutmur () [private], [slot]

Voici le graphe d'appel pour cette fonction :

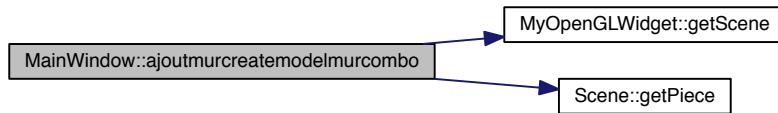


Voici le graphe des appelants de cette fonction :



5.5.2.6 void MainWindow ::ajoutmurmurcreatemodelmurcombo () [private], [slot]

Voici le graphe d'appel pour cette fonction :

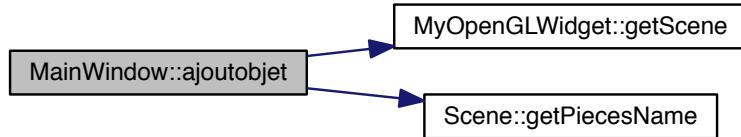


Voici le graphe des appelants de cette fonction :



5.5.2.7 void MainWindow ::ajoutobjet() [private], [slot]

Voici le graphe d'appel pour cette fonction :

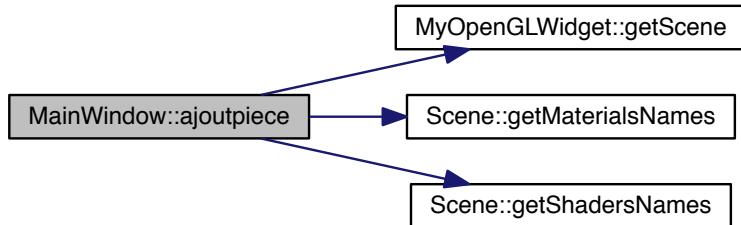


Voici le graphe des appelants de cette fonction :



5.5.2.8 void MainWindow ::ajoutpiece() [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



5.5.2.9 void MainWindow::Apercut() [private], [slot]

Voici le graphe des appelants de cette fonction :



5.5.2.10 void MainWindow::changeshader(const QString & text) [private], [slot]

Voici le graphe d'appel pour cette fonction :

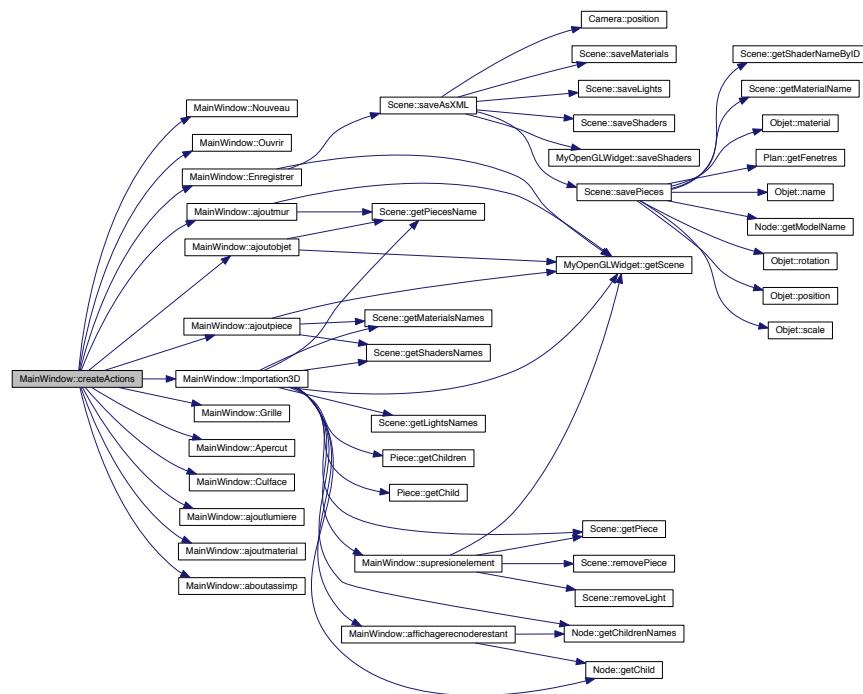


Voici le graphe des appelants de cette fonction :



5.5.2.11 void MainWindow ::createActions () [private]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appels de cette fonction :



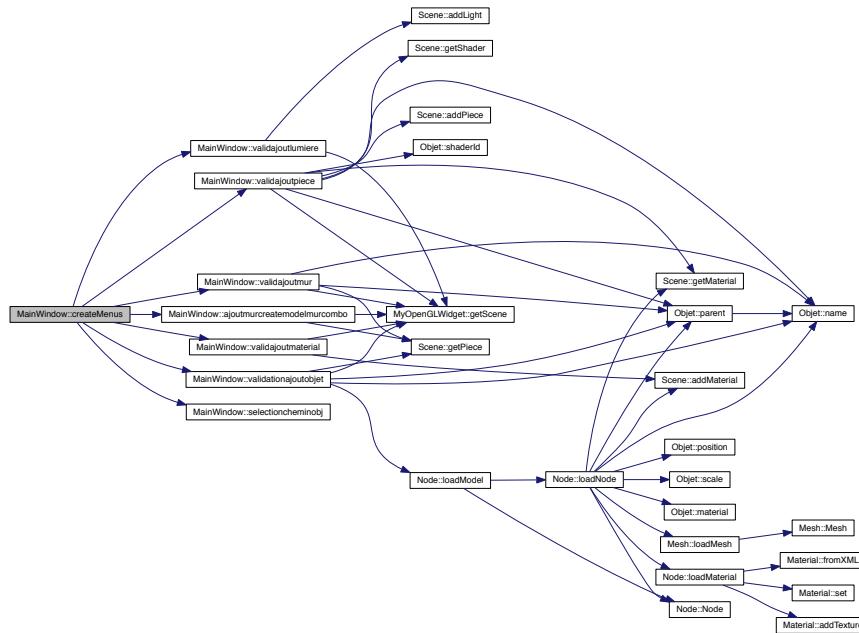
5.5.2.12 void MainWindow ::createListeDockwidget () [private]

Voici le graphe des appels de cette fonction :



5.5.2.13 void MainWindow ::createMenus () [private]

Voici le graphe d'appel pour cette fonction :

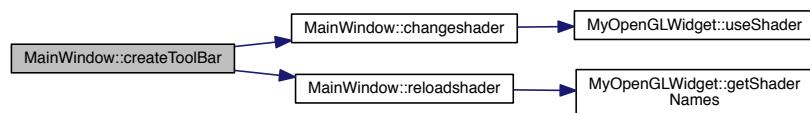


Voici le graphe des appels de cette fonction :



5.5.2.14 void MainWindow ::createToolBar () [private]

Voici le graphe d'appel pour cette fonction :

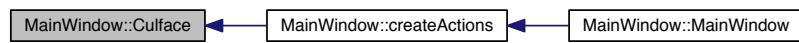


Voici le graphe des appelants de cette fonction :



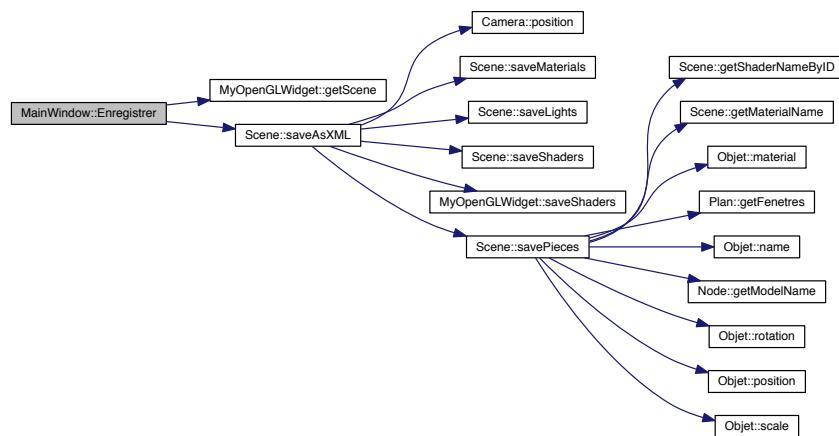
5.5.2.15 void MainWindow ::Culface () [private], [slot]

Voici le graphe des appelants de cette fonction :

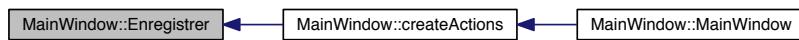


5.5.2.16 void MainWindow ::Enregistrer () [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



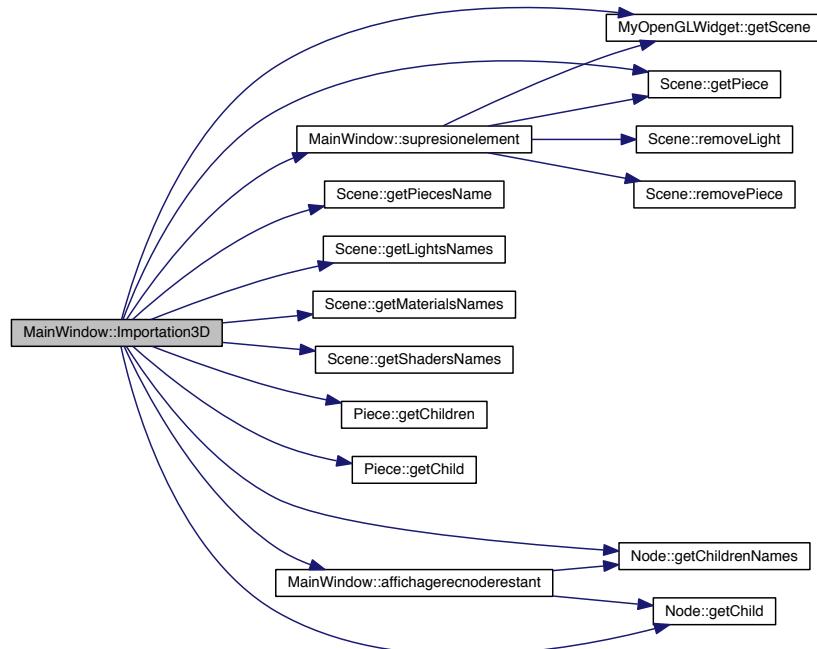
5.5.2.17 void MainWindow ::Grille() [private], [slot]

Voici le graphe des appelants de cette fonction :



5.5.2.18 void MainWindow ::Importation3D() [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

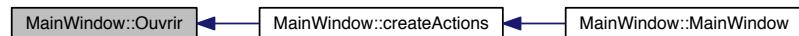


5.5.2.19 void MainWindow ::Nouveau() [private], [slot]

Voici le graphe des appelants de cette fonction :

**5.5.2.20 void MainWindow ::Ouvrir() [private], [slot]**

Voici le graphe des appelants de cette fonction :

**5.5.2.21 void MainWindow ::reloadshader() [private], [slot]**

Voici le graphe d'appel pour cette fonction :

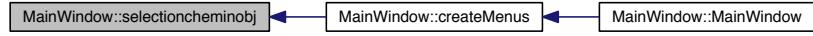


Voici le graphe des appelants de cette fonction :



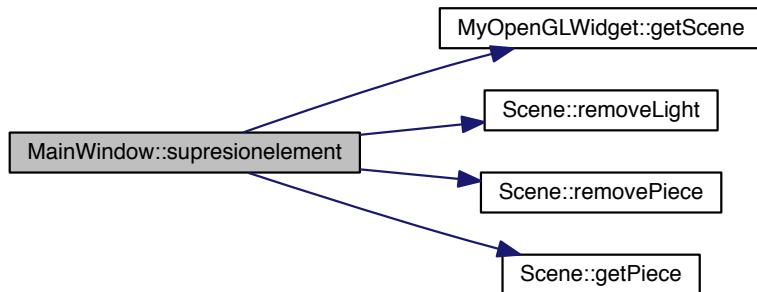
5.5.2.22 void MainWindow ::selectioncheminobj() [private], [slot]

Voici le graphe des appels de cette fonction :



5.5.2.23 void MainWindow ::supresionelement() [private], [slot]

Voici le graphe d'appel pour cette fonction :

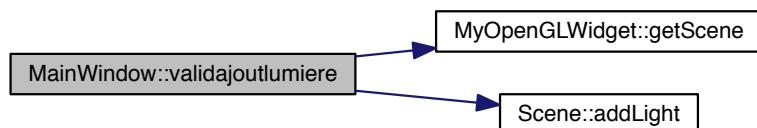


Voici le graphe des appels de cette fonction :

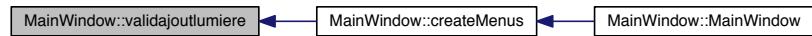


5.5.2.24 void MainWindow ::validajoutlumiere() [private], [slot]

Voici le graphe d'appel pour cette fonction :

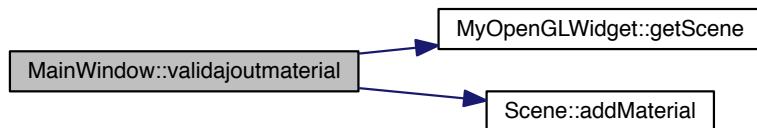


Voici le graphe des appelants de cette fonction :

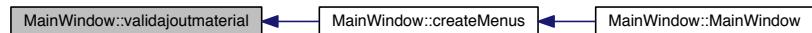


5.5.2.25 void MainWindow ::validajoutmaterial() [private], [slot]

Voici le graphe d'appel pour cette fonction :

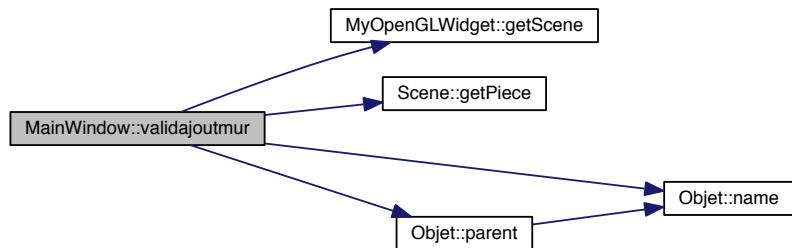


Voici le graphe des appelants de cette fonction :



5.5.2.26 void MainWindow ::validajoutmur() [private], [slot]

Voici le graphe d'appel pour cette fonction :

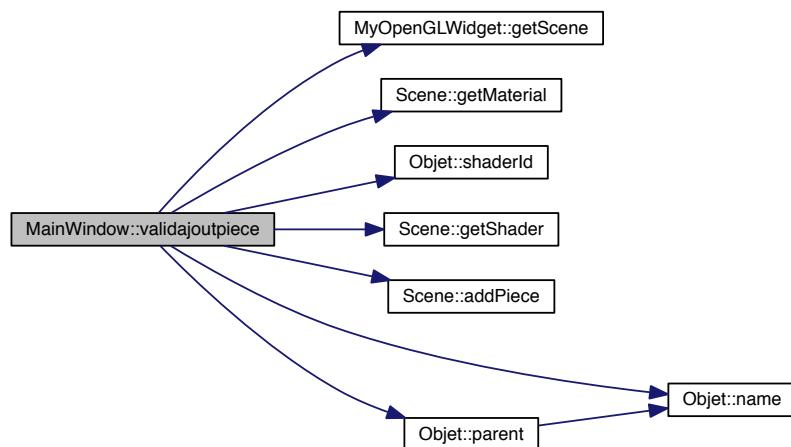


Voici le graphe des appelants de cette fonction :



5.5.2.27 void MainWindow ::validajoutpiece () [private], [slot]

Voici le graphe d'appel pour cette fonction :

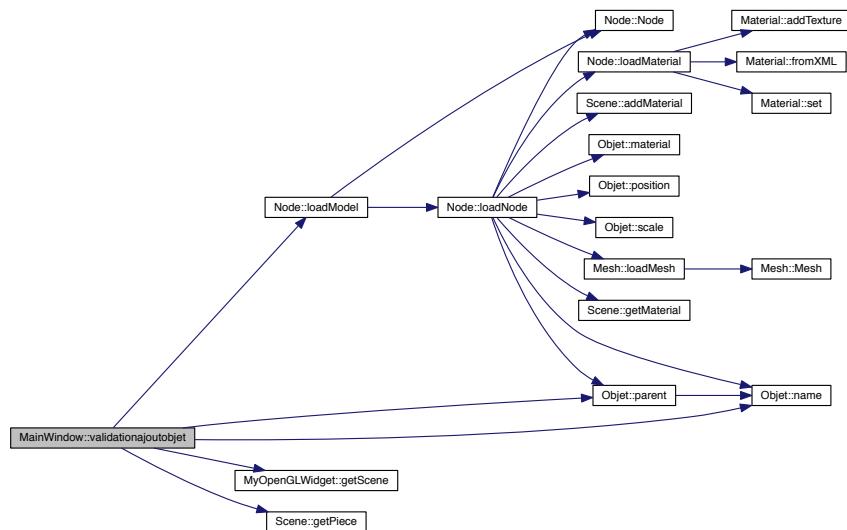


Voici le graphe des appelants de cette fonction :

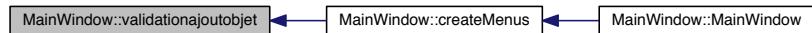


5.5.2.28 void MainWindow ::validationajoutobjet() [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appels de cette fonction :



5.5.3 Documentation des données membres

5.5.3.1 QAction* MainWindow ::about [private]

5.5.3.2 QAction* MainWindow ::aboutassimpAct [private]

5.5.3.3 QAction* MainWindow ::aboutqtAct [private]

5.5.3.4 QMenu* MainWindow ::Affichage [private]

5.5.3.5 QMenu* MainWindow ::Aide [private]

5.5.3.6 QMenu* MainWindow ::ajoutelement [private]

5.5.3.7 QAction* MainWindow ::ajoutlumierAct [private]

5.5.3.8 QAction* MainWindow ::ajoutmaterialAct [private]

5.5.3.9 QAction* MainWindow ::ajoutmurAct [private]

5.5.3.10 QAction* MainWindow ::ajoutobjetAct [private]

5.5.3.11 QAction* MainWindow::ajoutpieceAct [private]

5.5.3.12 QSpinBox* MainWindow::ajoutpiecedimx [private]

5.5.3.13 QSpinBox* MainWindow::ajoutpiecedimy [private]

5.5.3.14 QSpinBox* MainWindow::ajoutpiecedimz [private]

5.5.3.15 QHBoxLayout* MainWindow::ajoutpiececelabelmure [private]

5.5.3.16 QHBoxLayout* MainWindow::ajoutpiececelayoutcombo [private]

5.5.3.17 QHBoxLayout* MainWindow::ajoutpiececelayoutdim [private]

5.5.3.18 QHBoxLayout* MainWindow::ajoutpiececelayoutnom [private]

5.5.3.19 QHBoxLayout* MainWindow::ajoutpiecepiececer1 [private]

5.5.3.20 QHBoxLayout* MainWindow::ajoutpiecepiececer2 [private]

5.5.3.21 QHBoxLayout* MainWindow::ajoutpiecepiececer3 [private]

5.5.3.22 QHBoxLayout* MainWindow::ajoutpieceshaderlayout [private]

5.5.3.23 QLabel* MainWindow::ambajoutmaterial [private]

5.5.3.24 QAction* MainWindow::apercutAct [private]

5.5.3.25 QPushButton* MainWindow::boutonajoutlumiere [private]

5.5.3.26 QPushButton* MainWindow::boutonajoutmaterial [private]

5.5.3.27 QPushButton* MainWindow::boutonajoutmur [private]

5.5.3.28 QPushButton* MainWindow::boutonajoutobjet [private]

5.5.3.29 QPushButton* MainWindow::boutonajoutpiece [private]

5.5.3.30 QPushButton* MainWindow::boutonlisteelement [private]

5.5.3.31 QPushButton* MainWindow::boutonlisteelementdelete [private]

5.5.3.32 QPushButton* MainWindow::boutonpieceajoutmur [private]

5.5.3.33 QPushButton* MainWindow::boutonselectionobjajoutobjet [private]

5.5.3.34 QLineEdit* MainWindow::champnomajoutobjet [private]

5.5.3.35 QCheckBox* MainWindow::checkmur1 [private]

5.5.3.36 QCheckBox* MainWindow::checkmur2 [private]

5.5.3.37 QCheckBox* MainWindow::checkmur3 [private]

5.5.3.38 QCheckBox* MainWindow::checkmur4 [private]

- 5.5.3.39 **QCheckBox*** MainWindow : **:checkmur5** [private]
- 5.5.3.40 **QCheckBox*** MainWindow : **:checkmur6** [private]
- 5.5.3.41 **QComboBox*** MainWindow : **:comboajoutpiece** [private]
- 5.5.3.42 **QComboBox*** MainWindow : **:comboajoutpieceshader** [private]
- 5.5.3.43 **QComboBox*** MainWindow : **:combomurajoutmur** [private]
- 5.5.3.44 **QComboBox*** MainWindow : **:combopieceajoutmur** [private]
- 5.5.3.45 **QComboBox*** MainWindow : **:combopieceajoutobjet** [private]
- 5.5.3.46 **QAction*** MainWindow : **:culfaceAct** [private]
- 5.5.3.47 **QLabel*** MainWindow : **:difajoutlumiere** [private]
- 5.5.3.48 **QLabel*** MainWindow : **:difajoutmaterial** [private]
- 5.5.3.49 **QDockWidget*** MainWindow : **:dock_list_elements** [private]
- 5.5.3.50 **Mondock*** MainWindow : **:dock_perso** [private]
- 5.5.3.51 **QDockWidget*** MainWindow : **:dockajoutlumiere** [private]
- 5.5.3.52 **QDockWidget*** MainWindow : **:dockajoutmateriaux** [private]
- 5.5.3.53 **QDockWidget*** MainWindow : **:dockajoutmur** [private]
- 5.5.3.54 **QDockWidget*** MainWindow : **:dockajoutobjet** [private]
- 5.5.3.55 **QDockWidget*** MainWindow : **:dockajoutpiece** [private]
- 5.5.3.56 **QMenu*** MainWindow : **:Edition** [private]
- 5.5.3.57 **QAction*** MainWindow : **:enregistrerAct** [private]
- 5.5.3.58 **QMenu*** MainWindow : **:Fichier** [private]
- 5.5.3.59 **QToolBar*** MainWindow : **:fileToolBar** [private]
- 5.5.3.60 **QAction*** MainWindow : **:grilleAct** [private]
- 5.5.3.61 **QAction*** MainWindow : **:importation3DAct** [private]
- 5.5.3.62 **QStandardItem*** MainWindow : **:item** [private]
- 5.5.3.63 **QAction*** MainWindow : **:kiterAct** [private]
- 5.5.3.64 **QLabel*** MainWindow : **:labelajoutpieceshader** [private]
- 5.5.3.65 **QLabel*** MainWindow : **:labelcheminobjajoutobj** [private]
- 5.5.3.66 **QLabel*** MainWindow : **:labelcomboajoutpiece** [private]

5.5.3.67 QLabel* MainWindow ::labeldimajoutpiece [private]

5.5.3.68 QLabel* MainWindow ::labelmur1 [private]

5.5.3.69 QLabel* MainWindow ::labelmur2 [private]

5.5.3.70 QLabel* MainWindow ::labelmur3 [private]

5.5.3.71 QLabel* MainWindow ::labelmur4 [private]

5.5.3.72 QLabel* MainWindow ::labelmur5 [private]

5.5.3.73 QLabel* MainWindow ::labelmur6 [private]

5.5.3.74 QLabel* MainWindow ::labelmurajoutpiece [private]

5.5.3.75 QLabel* MainWindow ::labelmurcomboajoutmur [private]

5.5.3.76 QLabel* MainWindow ::labelnomajoutobjet [private]

5.5.3.77 QLabel* MainWindow ::labelnomajoutpiece [private]

5.5.3.78 QLabel* MainWindow ::labelpieceajoutobjet [private]

5.5.3.79 QLabel* MainWindow ::labelpiececomboajoutmur [private]

5.5.3.80 QVBoxLayout* MainWindow ::layoutajoutlumiere [private]

5.5.3.81 QVBoxLayout* MainWindow ::layoutajoutmaterial [private]

5.5.3.82 QHBoxLayout* MainWindow ::layoutajoutobjetnom [private]

5.5.3.83 QHBoxLayout* MainWindow ::layoutajoutobjetpiece [private]

5.5.3.84 QHBoxLayout* MainWindow ::layoutambajoutmaterial [private]

5.5.3.85 QHBoxLayout* MainWindow ::layoutcheminajoutobjet [private]

5.5.3.86 QHBoxLayout* MainWindow ::layoutcombotnommurajoutmur [private]

5.5.3.87 QHBoxLayout* MainWindow ::layoutcombotnompieceajoutmur [private]

5.5.3.88 QHBoxLayout* MainWindow ::layoutdifajoutlumiere [private]

5.5.3.89 QHBoxLayout* MainWindow ::layoutdifajoutmaterial [private]

5.5.3.90 QVBoxLayout* MainWindow ::layoutlistescene [private]

5.5.3.91 QHBoxLayout* MainWindow ::layoutnomajoutlumiere [private]

5.5.3.92 QHBoxLayout* MainWindow ::layoutnomajoutmaterial [private]

5.5.3.93 QHBoxLayout* MainWindow ::layoutposajoutlumiere [private]

5.5.3.94 QHBoxLayout* MainWindow ::layoutspeajoutlumiere [private]

- 5.5.3.95 **QHBoxLayout* MainWindow** :**:layoutspeajoutmaterial** [private]
- 5.5.3.96 **QStandardItem* MainWindow** :**:light** [private]
- 5.5.3.97 **QLineEdit* MainWindow** :**:lineeditajoutpiece** [private]
- 5.5.3.98 **QLineEdit* MainWindow** :**:lineeditnomajoutlumiere** [private]
- 5.5.3.99 **QLineEdit* MainWindow** :**:lineeditnomajoutmaterial** [private]
- 5.5.3.100 **QLineEdit* MainWindow** :**:lineeditobjajoutobj** [private]
- 5.5.3.101 **QStringList MainWindow** :**:listtemplight** [private]
- 5.5.3.102 **QStringList MainWindow** :**:listtempmaterial** [private]
- 5.5.3.103 **QStringList MainWindow** :**:listtempobjet** [private]
- 5.5.3.104 **QStringList MainWindow** :**:listtemphader** [private]
- 5.5.3.105 **QVBoxLayout* MainWindow** :**:mainlayoutajoutmur** [private]
- 5.5.3.106 **QVBoxLayout* MainWindow** :**:mainlayoutajoutobjet** [private]
- 5.5.3.107 **QVBoxLayout* MainWindow** :**:mainlayoutajoutpiece** [private]
- 5.5.3.108 **QStandardItem* MainWindow** :**:material** [private]
- 5.5.3.109 **QStandardItemModel* MainWindow** :**:modelajoutmurmur** [private]
- 5.5.3.110 **QStandardItemModel* MainWindow** :**:modelajoutmurpiece** [private]
- 5.5.3.111 **QStandardItemModel* MainWindow** :**:modelajoutobjet** [private]
- 5.5.3.112 **QStandardItemModel* MainWindow** :**:modele** [private]
- 5.5.3.113 **QStandardItemModel* MainWindow** :**:modelemateriaupiece** [private]
- 5.5.3.114 **QStandardItemModel* MainWindow** :**:modelepieceshader** [private]
- 5.5.3.115 **QStandardItemModel* MainWindow** :**:modelshaderpostpro** [private]
- 5.5.3.116 **QLabel* MainWindow** :**:nomajoutlumiere** [private]
- 5.5.3.117 **QLabel* MainWindow** :**:nomajoutmaterial** [private]
- 5.5.3.118 **QAction* MainWindow** :**:nouveauAct** [private]
- 5.5.3.119 **QStandardItem* MainWindow** :**:objet** [private]
- 5.5.3.120 **QMenu* MainWindow** :**:Outil** [private]
- 5.5.3.121 **QAction* MainWindow** :**:ouvrirAct** [private]
- 5.5.3.122 **QLabel* MainWindow** :**:positionajoutlumiere** [private]

5.5.3.123 **QAction* MainWindow::reloadshaderAct** [private]
5.5.3.124 **Scene* MainWindow::scenetemp** [private]
5.5.3.125 **QStandardItem* MainWindow::shader** [private]
5.5.3.126 **QComboBox* MainWindow::shaderpospro** [private]
5.5.3.127 **QLabel* MainWindow::speajoutlumiere** [private]
5.5.3.128 **QLabel* MainWindow::speajoutmaterial** [private]
5.5.3.129 **QDoubleSpinBox* MainWindow::spinambajoutmaterialx** [private]
5.5.3.130 **QDoubleSpinBox* MainWindow::spinambajoutmaterialy** [private]
5.5.3.131 **QDoubleSpinBox* MainWindow::spinambajoutmaterialz** [private]
5.5.3.132 **QDoubleSpinBox* MainWindow::spindifajoutlumierex** [private]
5.5.3.133 **QDoubleSpinBox* MainWindow::spindifajoutlumierey** [private]
5.5.3.134 **QDoubleSpinBox* MainWindow::spindifajoutlumierez** [private]
5.5.3.135 **QDoubleSpinBox* MainWindow::spindifajoutmaterialx** [private]
5.5.3.136 **QDoubleSpinBox* MainWindow::spindifajoutmaterialy** [private]
5.5.3.137 **QDoubleSpinBox* MainWindow::spindifajoutmaterialz** [private]
5.5.3.138 **QDoubleSpinBox* MainWindow::spinpositionajoutlumierex** [private]
5.5.3.139 **QDoubleSpinBox* MainWindow::spinpositionajoutlumierey** [private]
5.5.3.140 **QDoubleSpinBox* MainWindow::spinpositionajoutlumierez** [private]
5.5.3.141 **QDoubleSpinBox* MainWindow::spinspeajoutlumierex** [private]
5.5.3.142 **QDoubleSpinBox* MainWindow::spinspeajoutlumierey** [private]
5.5.3.143 **QDoubleSpinBox* MainWindow::spinspeajoutlumierez** [private]
5.5.3.144 **QDoubleSpinBox* MainWindow::spinspeajoutmateriala** [private]
5.5.3.145 **QDoubleSpinBox* MainWindow::spinspeajoutmaterialx** [private]
5.5.3.146 **QDoubleSpinBox* MainWindow::spinspeajoutmaterialy** [private]
5.5.3.147 **QDoubleSpinBox* MainWindow::spinspeajoutmaterialz** [private]
5.5.3.148 **QTreeView* MainWindow::vue** [private]
5.5.3.149 **MyOpenGLWidget* MainWindow::widget** [private]
5.5.3.150 **QWidget* MainWindow::widgetajoutlumiere** [private]

- 5.5.3.151 QWidget* MainWindow ::widgetajoutmaterial [private]
- 5.5.3.152 QWidget* MainWindow ::widgetajoutmur [private]
- 5.5.3.153 QWidget* MainWindow ::widgetajoutobjet [private]
- 5.5.3.154 QWidget* MainWindow ::widgetajoutpiece [private]
- 5.5.3.155 QWidget* MainWindow ::widgetdocklistscene [private]

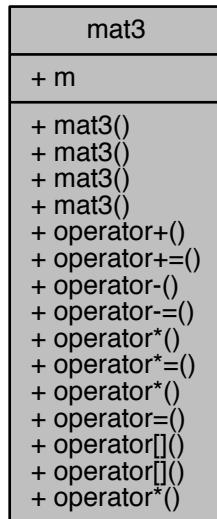
La documentation de cette classe a été générée à partir des fichiers suivants :

- [gui/mainwindows.hpp](#)
- [gui/mainwindows.cpp](#)

5.6 Référence de la structure mat3

```
#include <helper.hpp>
```

Graphe de collaboration de mat3 :



Fonctions membres publiques

- [mat3\(\)](#)
- [mat3\(int\)](#)
- [mat3\(float a, float b, float c, float d, float e, float f, float g, float h, float i\)](#)
- [mat3\(const mat3 &\)](#)
- [mat3 operator+\(const mat3 &\) const](#)
- [mat3 & operator+=\(const mat3 &\)](#)
- [mat3 operator-\(const mat3 &\) const](#)
- [mat3 & operator-=\(const mat3 &\)](#)
- [mat3 operator*\(const float\) const](#)
- [mat3 & operator*=\(const float\)](#)
- [mat3 operator*\(const mat3 &\) const](#)
- [mat3 & operator*=\(const mat3 &\)](#)
- [float & operator\[\]\(int\)](#)

- float `operator[]` (int) const
- `vec3 operator*(const vec3 &)` const

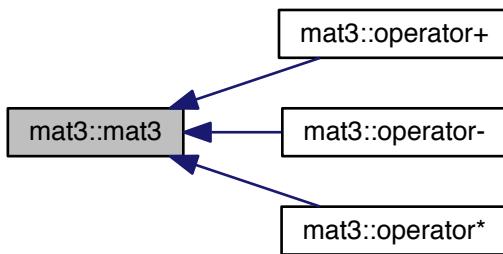
Attributs publics

- float `m` [9]

5.6.1 Documentation des constructeurs et destructeur

5.6.1.1 mat3 ::mat3()

Voici le graphe des appels de cette fonction :



5.6.1.2 mat3 ::mat3(int)

5.6.1.3 mat3 ::mat3(float a, float b, float c, float d, float e, float f, float g, float h, float i)

5.6.1.4 mat3 ::mat3(const mat3 & mm)

5.6.2 Documentation des fonctions membres

5.6.2.1 mat3 mat3 ::operator*(const float t) const

Voici le graphe d'appel pour cette fonction :



5.6.2.2 mat3 mat3 ::operator* (const mat3 & mm) const

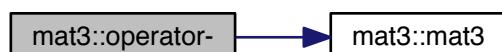
Voici le graphe d'appel pour cette fonction :

**5.6.2.3 vec3 mat3 ::operator* (const vec3 & vv) const****5.6.2.4 mat3 & mat3 ::operator*=(const float t)****5.6.2.5 mat3 mat3 ::operator+ (const mat3 & mm) const**

Voici le graphe d'appel pour cette fonction :

**5.6.2.6 mat3 & mat3 ::operator+= (const mat3 & mm)****5.6.2.7 mat3 mat3 ::operator- (const mat3 & mm) const**

Voici le graphe d'appel pour cette fonction :

**5.6.2.8 mat3 & mat3 ::operator-= (const mat3 & mm)****5.6.2.9 mat3 & mat3 ::operator= (const mat3 & mm)**

5.6.2.10 float & mat3 : :operator[] (int i)

5.6.2.11 float mat3 : :operator[] (int i) const

5.6.3 Documentation des données membres

5.6.3.1 float mat3 : :m[9]

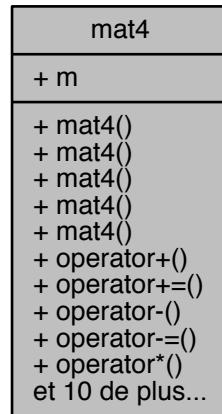
La documentation de cette structure a été générée à partir des fichiers suivants :

- [helper.hpp](#)
- [helper.cpp](#)

5.7 Référence de la structure mat4

```
#include <helper.hpp>
```

Graphe de collaboration de mat4 :



Fonctions membres publiques

- [mat4 \(\)](#)
- [mat4 \(int\)](#)
- [mat4 \(float a, float b, float c, float d, float e, float f, float g, float h, float i, float j, float k, float l, float m, float n, float o, float p\)](#)
- [mat4 \(float *v\)](#)
- [mat4 \(const mat4 &\)](#)
- [mat4 operator+ \(const mat4 &\) const](#)
- [mat4 & operator+= \(const mat4 &\)](#)
- [mat4 operator- \(const mat4 &\) const](#)
- [mat4 & operator-= \(const mat4 &\)](#)
- [mat4 operator* \(const float\) const](#)
- [mat4 & operator*= \(const float\)](#)
- [mat4 operator* \(const mat4 &\) const](#)
- [mat4 & operator*= \(const mat4 &\)](#)
- [mat4 & operator= \(const mat4 &\)](#)
- [float & operator\[\] \(int\)](#)
- [float operator\[\] \(int\) const](#)
- [vec4 operator* \(const vec4 &\) const](#)
- [mat4 transpose \(\) const](#)

- mat4 inverse () const
- float det () const

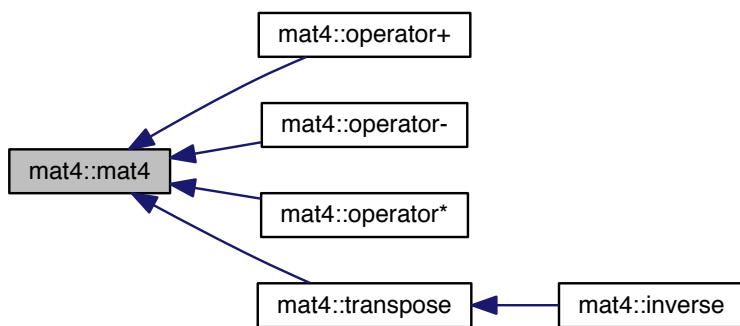
Attributs publics

- float m [16]

5.7.1 Documentation des constructeurs et destructeur

5.7.1.1 mat4 ::mat4 ()

Voici le graphe des appelants de cette fonction :



5.7.1.2 mat4 ::mat4 (int)

5.7.1.3 mat4 ::mat4 (float a, float b, float c, float d, float e, float f, float g, float h, float i, float j, float k, float l, float m, float n, float o, float p)

5.7.1.4 mat4 ::mat4 (float * v)

5.7.1.5 mat4 ::mat4 (const mat4 & mm)

5.7.2 Documentation des fonctions membres

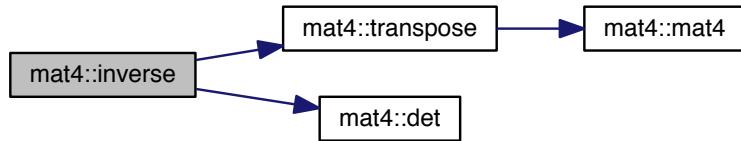
5.7.2.1 float mat4 ::det () const

Voici le graphe des appelants de cette fonction :

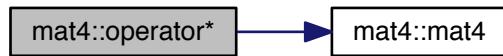


5.7.2.2 mat4 mat4 ::inverse () const

Voici le graphe d'appel pour cette fonction :

**5.7.2.3 mat4 mat4 ::operator* (const float t) const**

Voici le graphe d'appel pour cette fonction :

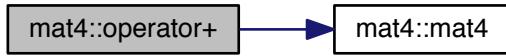
**5.7.2.4 mat4 mat4 ::operator* (const mat4 & mm) const**

Voici le graphe d'appel pour cette fonction :

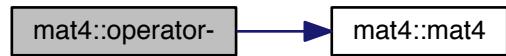
**5.7.2.5 vec4 mat4 ::operator* (const vec4 & vv) const****5.7.2.6 mat4 & mat4 ::operator*=(const float t)****5.7.2.7 mat4 & mat4 ::operator*=(const mat4 & mm)**

5.7.2.8 mat4 mat4 ::operator+ (const mat4 & mm) const

Voici le graphe d'appel pour cette fonction :

**5.7.2.9 mat4 & mat4 ::operator+= (const mat4 & mm)****5.7.2.10 mat4 mat4 ::operator- (const mat4 & mm) const**

Voici le graphe d'appel pour cette fonction :

**5.7.2.11 mat4 & mat4 ::operator-= (const mat4 & mm)****5.7.2.12 mat4 & mat4 ::operator= (const mat4 & mm)****5.7.2.13 float & mat4 ::operator[] (int i)****5.7.2.14 float mat4 ::operator[] (int i) const****5.7.2.15 mat4 mat4 ::transpose () const**

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appels de cette fonction :



5.7.3 Documentation des données membres

5.7.3.1 float mat4 ::m[16]

La documentation de cette structure a été générée à partir des fichiers suivants :

- [helper.hpp](#)
- [helper.cpp](#)

5.8 Référence de la classe Material

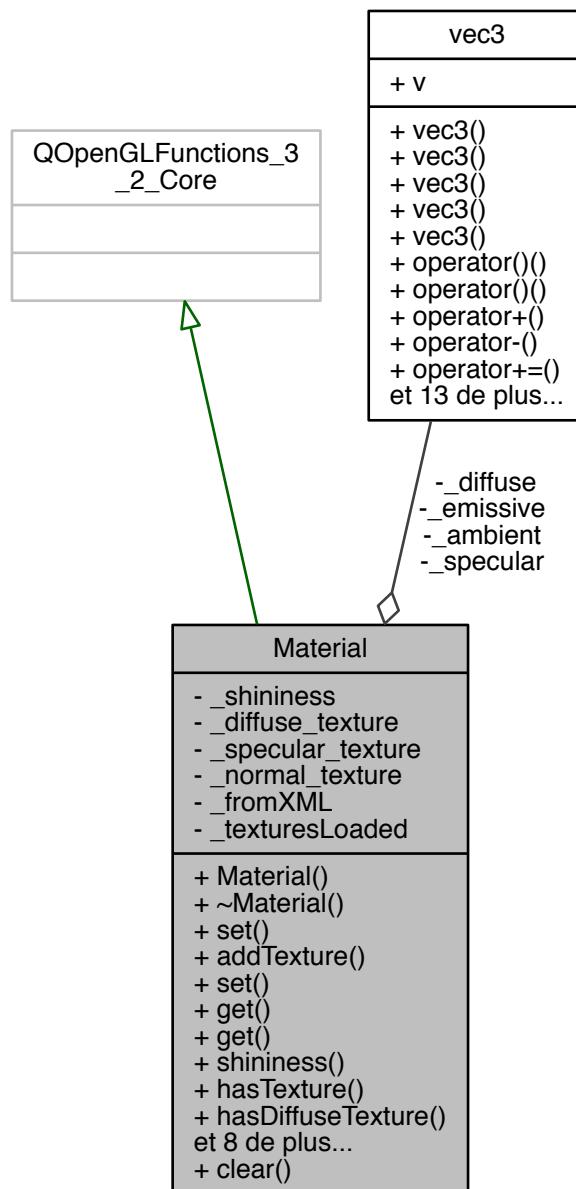
Matériaux pour la [Scene](#).

```
#include <material.hpp>
```

Graphe d'héritage de Material :



Graphe de collaboration de Material :



Fonctions membres publiques

- `Material (vec3 ambient=vec3(0.2, 0.2, 0.2), vec3 diffuse=vec3(0.8, 0.8, 0.8), vec3 specular=vec3(0.8, 0.8, 0.8), float shininess=0.0f, vec3 emissive=vec3(0.0, 0.0, 0.0))`
Constructeur.
- `~Material ()`
Destructeur.
- `void set (GLenum type, vec4 value)`
Met à jour la composante du matériaux.
- `void addTexture (const QString &texFile, const QString &type)`
Charge une texture et l'associe à un slot.

- void **set** (float shininess)
Met à jour la shininess du Material.
- **vec3 get** (GLenum type) const
Récupère la valeur d'un champ.
- **vec3 & get** (GLenum type)
Récupère la valeur d'un champ.
- float **shininess** () const
Valeur du shininess.
- bool **hasTexture** (unsigned char type) const
- bool **hasDiffuseTexture** () const
- bool **hasSpecularTexture** () const
- bool **hasNormalTexture** () const
- QString **getDiffuseTextureName** () const
Retourne le path de la texture diffuse.
- QString **getSpecularTextureName** () const
Retourne le path de la texture spéculaire.
- QString **getNormalTextureName** () const
Retourne le path de la normal map.
- void **update** ()
Met à jour le Material courant dans le shader actif.
- bool **isFromXML** () const
Indique si le Material a été chargé depuis le XML ou pas.
- void **fromXML** (bool b)
Indique si le Material est issu ou non du XML.

Fonctions membres publiques statiques

- static void **clear** ()
Libère les textures chargées.

Attributs privés

- **vec3 _ambient**
- **vec3 _diffuse**
- **vec3 _specular**
- float **_shininess**
- **vec3 _emissive**
- QOpenGLTexture * **_diffuse_texture**
- QOpenGLTexture * **_specular_texture**
- QOpenGLTexture * **_normal_texture**
- bool **_fromXML**

Attributs privés statiques

- static QMap< QString,
QOpenGLTexture * > **_texturesLoaded**

Amis

- QDebug **operator<<** (QDebug dbg, const Material &m)

5.8.1 Description détaillée

Matériaux pour la Scene.

Contient les informations nécessaire à la gestion d'un material pour la scène

5.8.2 Documentation des constructeurs et destructeur

```
5.8.2.1 Material ::Material ( vec3 ambient = vec3(0.2, 0.2, 0.2), vec3 diffuse = vec3(0.8, 0.8, 0.8),
    vec3 specular = vec3(0.8, 0.8, 0.8), float shininess = 0.0f, vec3 emissive =
    vec3(0.0, 0.0, 0.0) )
```

Constructeur.

Avertissement

composante emissive pas encore gérée

```
5.8.2.2 Material ::~Material ( )
```

Destructeur.

A faire Gérer libération de la texture quand plus utilisée

5.8.3 Documentation des fonctions membres

```
5.8.3.1 void Material ::addTexture ( const QString & texFile, const QString & type )
```

Charge une texture et l'associe à un slot.

Charge une texture et l'associe à un slot pour le passage au shader

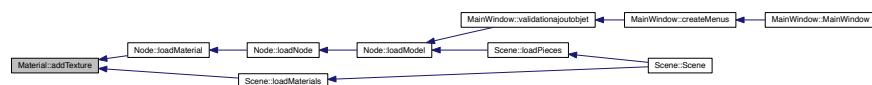
Paramètres

<i>texFile</i>	fichier de texture à charger
<i>int</i>	indice que la texture doit occuper

Avertissement

Ecrase la texture à l'indice choisi

Voici le graphe des appelants de cette fonction :



```
5.8.3.2 void Material ::clear ( ) [static]
```

Libère les textures chargées.

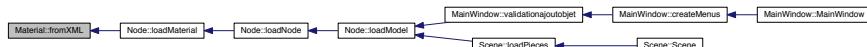
Voici le graphe des appelants de cette fonction :



5.8.3.3 void Material ::fromXML (bool b) [inline]

Indique si le [Material](#) est issu ou non du XML.

Voici le graphe des appelants de cette fonction :



5.8.3.4 vec3 Material ::get (GLenum type) const

Récupère la valeur d'un champ.

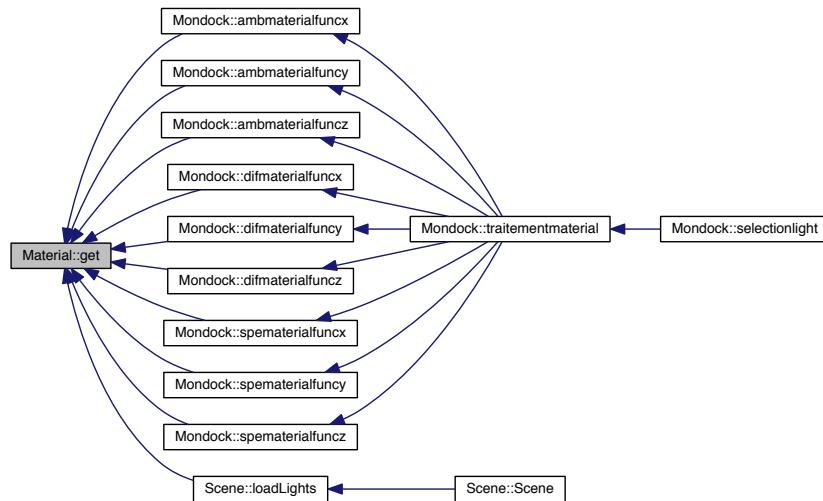
Paramètres

<code>type</code>	doit être parmi GL_AMBIENT, GL_DIFFUSE, GL_SPECULAR
-------------------	---

Renvoie

dans le cas de GL_AMBIENT, GL_DIFFUSE, GL_SPECULAR retourne un [vec3](#) correspondant à la composante, retourne [vec4\(\)](#) autrement

Voici le graphe des appelants de cette fonction :



5.8.3.5 vec3 & Material ::get (GLenum type)

Récupère la valeur d'un champ.

Paramètres

<i>type</i>	doit être parmi GL_AMBIENT, GL_DIFFUSE, GL_SPECULAR
-------------	---

Renvoie

dans le cas de GL_AMBIENT, GL_DIFFUSE, GL_SPECULAR retourne un `vec3` correspondant à la composante, retourne `vec4()` autrement

5.8.3.6 QString Material::getDiffuseTextureName() const

Retourne le path de la texture diffuse.

Renvoie

nom de la texture diffuse

5.8.3.7 QString Material::getNormalTextureName() const

Retourne le path de la normal map.

Renvoie

nom de la normal map

5.8.3.8 QString Material::getSpecularTextureName() const

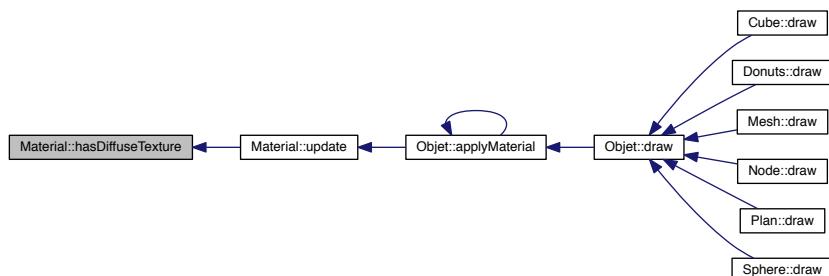
Retourne le path de la texture spéculaire.

Renvoie

nom de la texture spéculaire

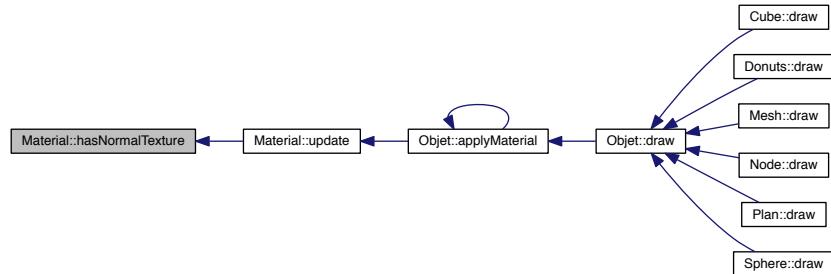
5.8.3.9 bool Material::hasDiffuseTexture() const

Voici le graphe des appels de cette fonction :



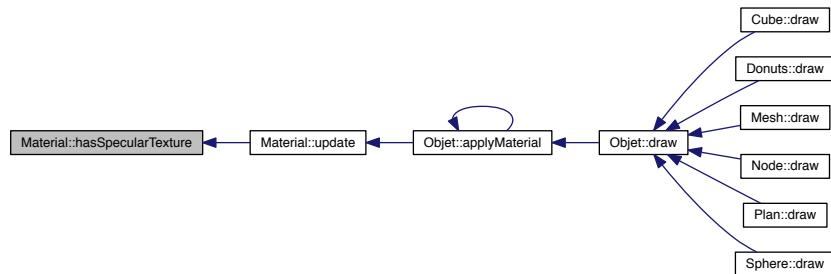
5.8.3.10 bool Material::hasNormalTexture() const

Voici le graphe des appelants de cette fonction :



5.8.3.11 bool Material::hasSpecularTexture() const

Voici le graphe des appelants de cette fonction :



5.8.3.12 bool Material::hasTexture(unsigned char type) const

5.8.3.13 bool Material::isFromXML() const [inline]

Indique si le **Material** a été chargé depuis le XML ou pas.

5.8.3.14 void Material::set(GLenum type, vec4 value)

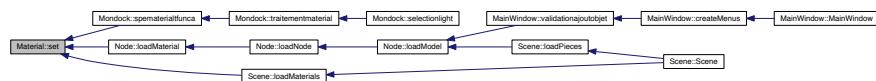
Met à jour la composante du matériaux.

Paramètres

<i>type</i>	la composante qui sera mise à jour, doit être parmis GL_AMBIENT, GL_DIFFUSE, GL_SPECULAR, GL_EMISSION
-------------	---

<i>value</i>	valeur
--------------	--------

Voici le graphe des appelants de cette fonction :



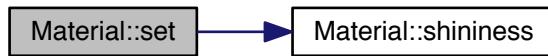
5.8.3.15 void Material ::set (float shininess)

Met à jour la shininess du [Material](#).

Paramètres

<i>shininess</i>	valeur
------------------	--------

Voici le graphe d'appel pour cette fonction :



5.8.3.16 float Material ::shininess () const

Valeur du shininess.

Renvoie

valeur du GL_SHININESS

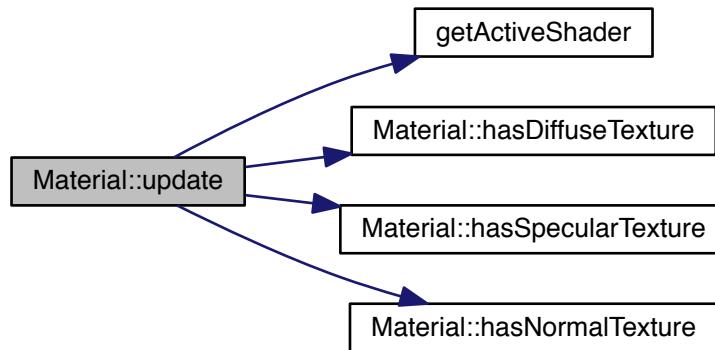
Voici le graphe des appelants de cette fonction :



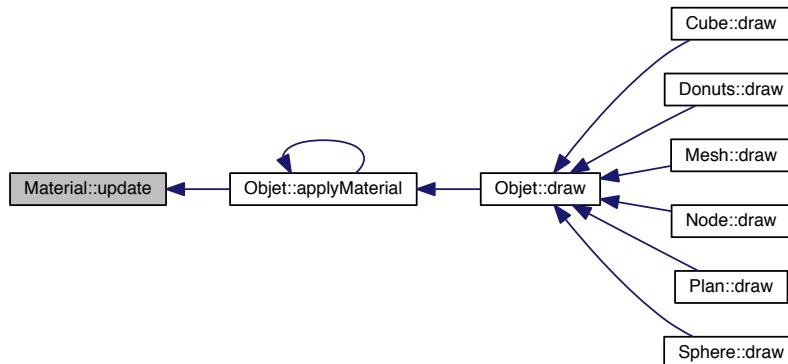
5.8.3.17 void Material ::update ()

Met à jour le [Material](#) courant dans le shader actif.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



5.8.4 Documentation des fonctions amies et associées

5.8.4.1 `QDebug operator<< (QDebug dbg, const Material & m)` [friend]

5.8.5 Documentation des données membres

5.8.5.1 `vec3 Material::_ambient [private]`

5.8.5.2 `vec3 Material::_diffuse [private]`

5.8.5.3 `QOpenGLTexture* Material::_diffuse_texture [private]`

5.8.5.4 `vec3 Material::_emissive [private]`

5.8.5.5 `bool Material::_fromXML [private]`

5.8.5.6 **QOpenGLTexture* Material::_normal_texture** [private]

5.8.5.7 **float Material::_shininess** [private]

5.8.5.8 **vec3 Material::_specular** [private]

5.8.5.9 **QOpenGLTexture* Material::_specular_texture** [private]

5.8.5.10 **QMap<QString, QOpenGLTexture * > Material::_texturesLoaded** [static], [private]

Ensemble des textures déjà chargées pour réutilisation

La documentation de cette classe a été générée à partir des fichiers suivants :

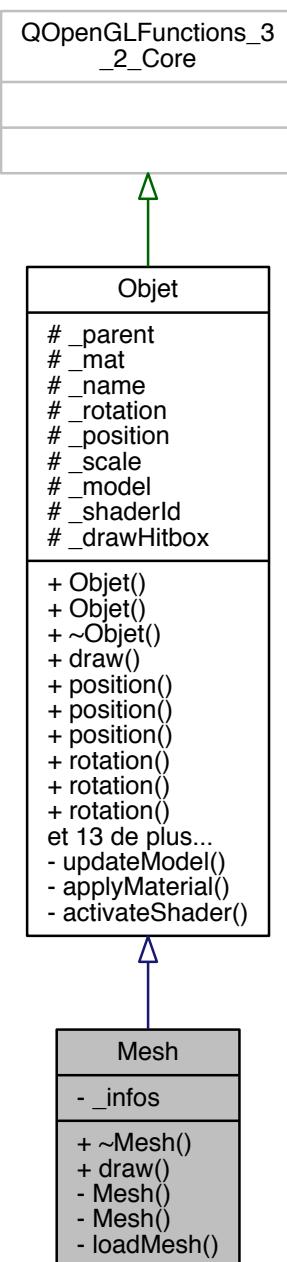
- [scene/material.hpp](#)
- [scene/material.cpp](#)

5.9 Référence de la classe Mesh

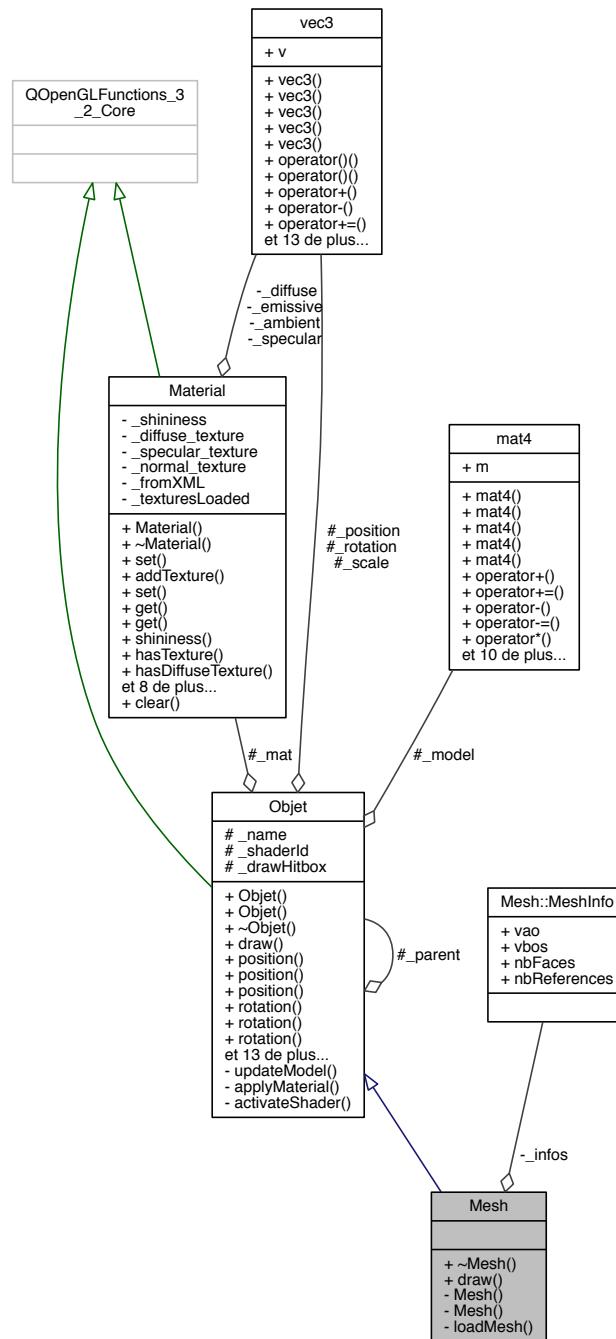
Charge un modèle 3D.

```
#include <mesh.hpp>
```

Graphe d'héritage de Mesh :



Graphe de collaboration de Mesh :



Classes

- struct `MeshInfo`

Fonctions membres publiques

- `~Mesh ()`
Destructeur.

- void [draw \(\)](#)
Affiche le modèle.

Fonctions membres privées

- [Mesh \(\)](#)
Constructeur privé
- [Mesh \(const Mesh &m\)](#)
Constructeur par recopie.

Fonctions membres privées statiques

- static [Mesh * loadMesh \(const aiMesh *scene\)](#)
Charge un mesh.

Attributs privés

- [MeshInfo * _infos](#)

Amis

- class [Node](#)

Membres hérités additionnels

5.9.1 Description détaillée

Charge un modèle 3D.

Permet la lecture de fichiers contenant des modèles 3D

Avertissement

Tout les formats de textures ne sont pas supportés, se référer au support de QImage

5.9.2 Documentation des constructeurs et destructeur

5.9.2.1 Mesh ::Mesh () [private]

Constructeur privé

Utiliser Node ::load(const QString &fileName, Scene * scene)

Voir également

[Node ::loadModel\(\)](#)

Voici le graphe des appels de cette fonction :



5.9.2.2 Mesh ::Mesh (const Mesh & m) [private]

Constructeur par recopie.

5.9.2.3 Mesh ::~Mesh ()

Destructeur.

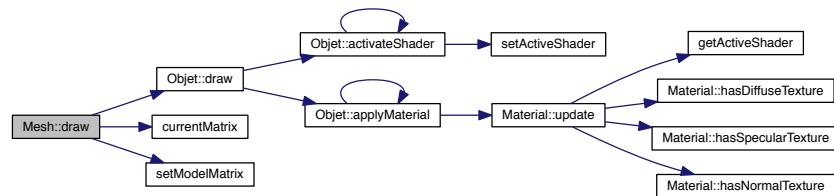
5.9.3 Documentation des fonctions membres

5.9.3.1 void Mesh ::draw () [virtual]

Affiche le modèle.

Réimplémentée à partir de [Objet](#).

Voici le graphe d'appel pour cette fonction :



5.9.3.2 Mesh * Mesh ::loadMesh (const aiMesh * scene) [static], [private]

Charge un mesh.

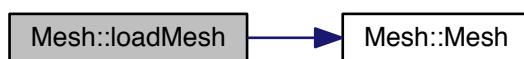
Paramètres

<code>scene</code>	scène générée par assimp
--------------------	--------------------------

Renvoie

un pointeur sur [Mesh](#), NULL si erreur

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



5.9.4 Documentation des fonctions amies et associées

5.9.4.1 friend class Node [friend]

5.9.5 Documentation des données membres

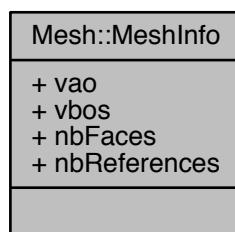
5.9.5.1 MeshInfo* Mesh::infos [private]

La documentation de cette classe a été générée à partir des fichiers suivants :

- objets/mesh.hpp
- objets/mesh.cpp

5.10 Référence de la structure Mesh ::MeshInfo

Graphe de collaboration de Mesh ::MeshInfo :



Attributs publics

- GLuint **vao**
- QList< GLuint > **vbos**
- GLuint **nbFaces**
- unsigned int **nbReferences**

5.10.1 Documentation des données membres

5.10.1.1 GLuint Mesh ::MeshInfo ::nbFaces

5.10.1.2 unsigned int Mesh ::MeshInfo ::nbReferences

5.10.1.3 GLuint Mesh ::MeshInfo ::vao

5.10.1.4 QList<GLuint> Mesh::MeshInfo::vbos

La documentation de cette structure a été générée à partir du fichier suivant :
— [objets/mesh.hpp](#)

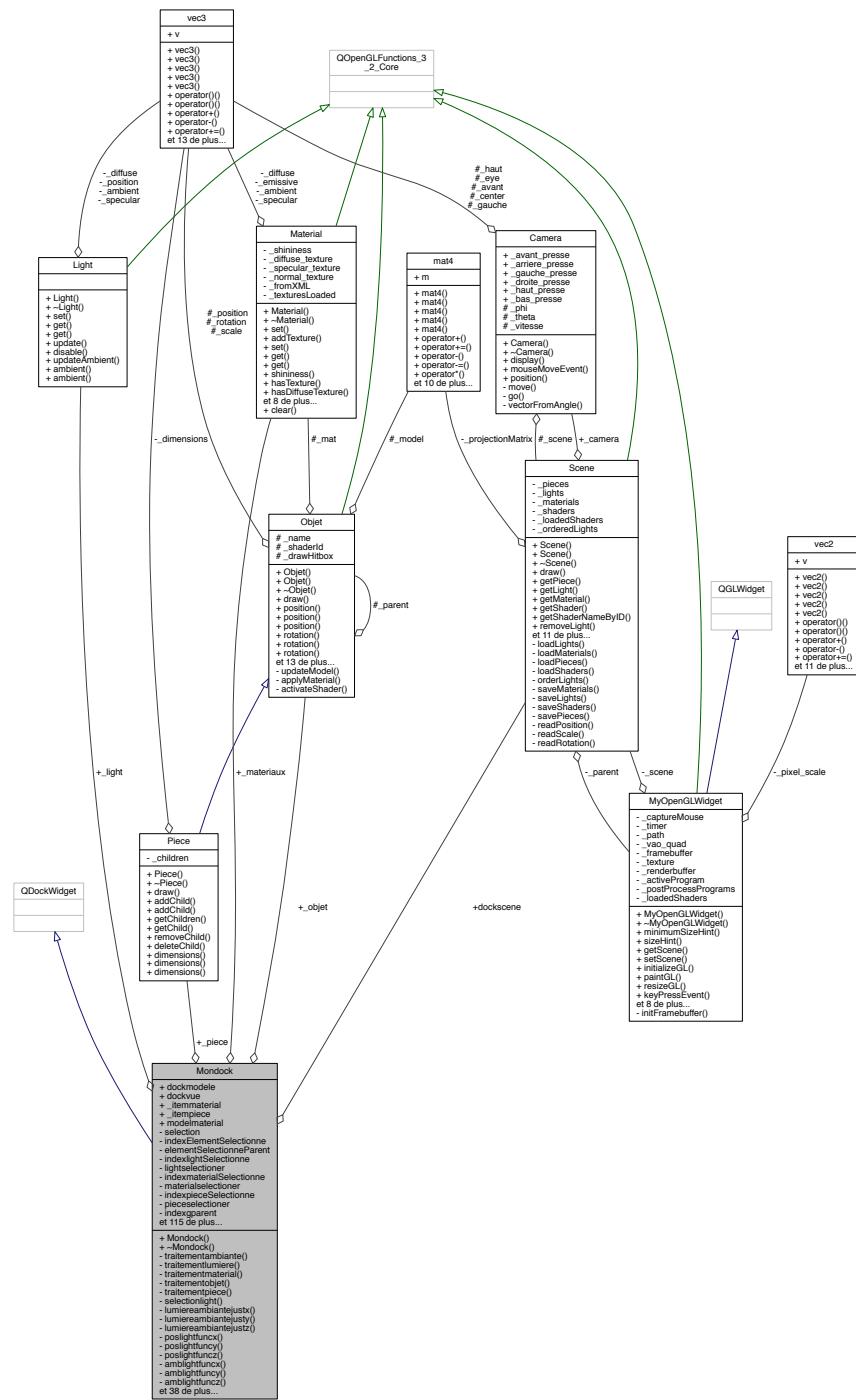
5.11 Référence de la classe Mondock

```
#include <mondock.hpp>
```

Graphe d'héritage de Mondock :



Graphe de collaboration de Mondock :



Fonctions membres publiques

- **Mondock** (const QString &title, QWidget *parent=0, Qt::WindowFlags flags=0)
- **~Mondock** ()

Attributs publics

- **Scene *** dockscene

- Light * _light
- Material * _materiaux
- Objet * _objet
- Piece * _piece
- QStandardItemModel * dockmodele
- QTreeView * dockvue
- QStandardItem * _itemmaterial
- QStandardItem * _itempiece
- QStandardItemModel * modelmaterial

Connecteurs privés

- void selectionlight ()
- void lumiereambiantejustx (int x)
- void lumiereambiantejusty (int x)
- void lumiereambiantejustz (int x)
- void poslightfuncx (double x)
- void poslightfuncy (double x)
- void poslightfuncz (double x)
- void amblightfuncx (int x)
- void amblightfuncy (int x)
- void amblightfuncz (int x)
- void diflightfuncx (int x)
- void diflightfuncy (int x)
- void diflightfuncz (int x)
- void spelightfuncx (int x)
- void spelightfuncy (int x)
- void spelightfuncz (int x)
- void ambmaterialfuncx (int x)
- void ambmaterialfuncy (int x)
- void ambmaterialfuncz (int x)
- void difmaterialfuncx (int x)
- void difmaterialfuncy (int x)
- void difmaterialfuncz (int x)
- void spematerialfuncx (int x)
- void spematerialfuncy (int x)
- void spematerialfuncz (int x)
- void spematerialfunca (int x)
- void matobjet (const QString &text)
- void pereobjet (const QString &text)
- void rotobjectx (double x)
- void rotobjecty (double x)
- void rotobjectz (double x)
- void transobjectx (double x)
- void transobjecty (double x)
- void transobjectz (double x)
- void scaleobjectx (double x)
- void scaleobjecty (double x)
- void scaleobjectz (double x)
- void scalesliderobject (int x)
- void slotpositionpiecex (int x)
- void slotpositionpiecey (int x)
- void slotpositionpiecez (int x)
- void scalepiecex (double x)
- void scalepiecey (double x)
- void scalepiecez (double x)
- void scalesliderpiece (int x)
- void rotpiecex (double x)
- void rotpiecey (double x)
- void rotpiecez (double x)

Fonctions membres privées

- void traitementambiante ()
- void traitementlumiere ()
- void traitementmaterial ()
- void traitementobjet ()
- void traitementpiece ()

Attributs privés

- QItemSelectionModel * selection

```
— QModelIndex indexElementSelectionne
— QVariant elementSelectionneParent
— QModelIndex indexlightSelectionne
— QVariant lightselectioner
— QModelIndex indexmaterialSelectionne
— QVariant materialselectioner
— QModelIndex indexpieceSelectionne
— QVariant pieceselectioner
— QModelIndex indexparent
— QVariant elementSelectionneGParent
— QModelIndex indexobjetSelectionne
— QVariant objetselectioner
— QStandardItemModel * modelpiece
— QWidget * editionambiante
— QSpinBox * spinboxlumiereambiantex
— QSpinBox * spinboxlumiereambiantey
— QSpinBox * spinboxlumiereambiantez
— QBoxLayout * layouteditionlighamb
— QTabWidget * tablight
— QWidget * tablightpos
— QBoxLayout * tablightposlayout
— QDoubleSpinBox * lightdoublespinboxx
— QDoubleSpinBox * lightdoublespinboxy
— QDoubleSpinBox * lightdoublespinboxz
— QWidget * tablightdf
— QVBoxLayout * tabdiflayout
— QHBoxLayout * lightdiflayout
— QSlider * sliderdfx
— QSlider * sliderdfy
— QSlider * sliderdfz
— QHBoxLayout * labeldiflayout
— QLabel * labelfdfx
— QLabel * labelfdfy
— QLabel * labelfdfz
— QWidget * tablightspe
— QVBoxLayout * tabspeplayout
— QHBoxLayout * lightspeplayout
— QSlider * sliderspe
— QSlider * sliderspey
— QSlider * sliderspez
— QHBoxLayout * labelspeplayout
— QLabel * labelspe
— QLabel * labelspey
— QLabel * labelspez
— QWidget * tablightamb
— QVBoxLayout * tabamblayout
— QHBoxLayout * lightamblayout
— QSlider * sliderambx
— QSlider * slideramby
— QSlider * sliderambz
— QHBoxLayout * labelamblayout
— QLabel * labelambx
— QLabel * labelamby
— QLabel * labelambz
— QTabWidget * tabmaterial
— QWidget * tabmaterialamb
— QHBoxLayout * tabmaterialamblayout
— QSpinBox * materialspinboxambx
— QSpinBox * materialspinboxamby
— QSpinBox * materialspinboxambz
— QWidget * tabmaterialdf
— QHBoxLayout * tabmaterialdflayout
— QSpinBox * materialspinboxdfx
— QSpinBox * materialspinboxdfy
— QSpinBox * materialspinboxdfz
— QWidget * tabmaterialspe
— QHBoxLayout * tabmaterialspeplayout
— QSpinBox * materialspinboxspe
— QSpinBox * materialspinboxspey
— QSpinBox * materialspinboxspez
— QSpinBox * materialspinboxspea
— QTabWidget * tabobjet
— QWidget * tabobjetprop
— QVBoxLayout * layouttabobjetprop
— QHBoxLayout * layouttabobjetproplabel
— QHBoxLayout * layouttabobjetpropcombo
— QComboBox * combomaterial
```

```
— QComboBox * comboparent
— QLabel * labelobjetproprmaterial
— QLabel * labelobjetproprparent
— QWidget * tabobjetrotation
— QHBoxLayout * layouttabobjetrotation
— QDoubleSpinBox * boxobjetrotationx
— QDoubleSpinBox * boxobjetrotationy
— QDoubleSpinBox * boxobjetrotationz
— QWidget * tabobjetrans
— QHBoxLayout * layouttabobjetrans
— QDoubleSpinBox * boxobjetransx
— QDoubleSpinBox * boxobjetransy
— QDoubleSpinBox * boxobjetransz
— QWidget * tabobjetscale
— QVBoxLayout * layoutglobalscaleobjet
— QHBoxLayout * layouttabobjetscale
— QHBoxLayout * layoutobjetscaleall
— QDoubleSpinBox * boxobjetscalex
— QDoubleSpinBox * boxobjetscaley
— QDoubleSpinBox * boxobjetscalez
— QLabel * labelsliderobjetscale
— QSlider * sliderobjetscale
— QTabWidget * tabpiece
— QWidget * widgetpiececdim
— QHBoxLayout * piecedimlayout
— QSpinBox * dimentionpiecex
— QSpinBox * dimentionpiecey
— QSpinBox * dimentionpiecez
— QWidget * widgetpieceposi
— QHBoxLayout * pieceposilayout
— QSpinBox * positionpiecex
— QSpinBox * positionpiecey
— QSpinBox * positionpiecez
— QWidget * widgetpiecescale
— QVBoxLayout * piecescalelayout
— QHBoxLayout * scalesliderlayout
— QHBoxLayout * scalespinboxlayout
— QDoubleSpinBox * scalespinboxx
— QDoubleSpinBox * scalespinboxy
— QDoubleSpinBox * scalespinboxz
— QLabel * scalelabel
— QSlider * scaleslider
— QWidget * widgetpiecerotate
— QHBoxLayout * piicerotatelayout
— QDoubleSpinBox * rotspinboxx
— QDoubleSpinBox * rotspinboxy
— QDoubleSpinBox * rotspinboxz
```

5.11.1 Documentation des constructeurs et destructeur

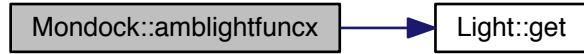
5.11.1.1 Mondock :**Mondock (const QString & title, QWidget * parent = 0, Qt::WindowFlags flags = 0)**

5.11.1.2 Mondock :**~Mondock ()**

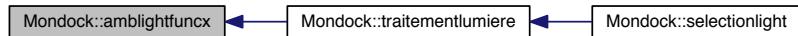
5.11.2 Documentation des fonctions membres

5.11.2.1 void Mondock::amblightfuncx(int x) [private], [slot]

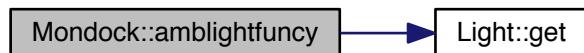
Voici le graphe d'appel pour cette fonction :



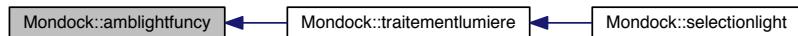
Voici le graphe des appels de cette fonction :

**5.11.2.2 void Mondock::amblightfuncy(int x) [private], [slot]**

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appels de cette fonction :

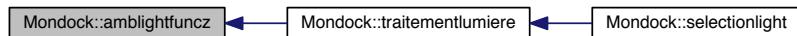


5.11.2.3 void Mondock : :amblightfuncz(int x) [private], [slot]

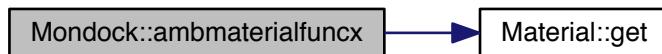
Voici le graphe d'appel pour cette fonction :



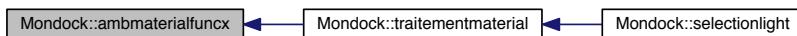
Voici le graphe des appelants de cette fonction :

**5.11.2.4 void Mondock : :ambmaterialfuncx(int x) [private], [slot]**

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

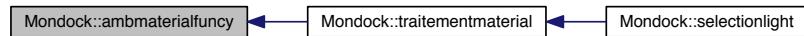


5.11.2.5 void Mondock::ambmaterialfuncy (int x) [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

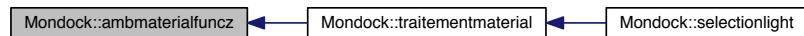


5.11.2.6 void Mondock::ambmaterialfuncz (int x) [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

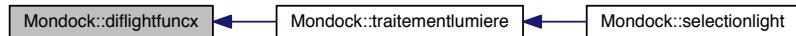


5.11.2.7 void Mondock ::diflightfuncx (int x) [private], [slot]

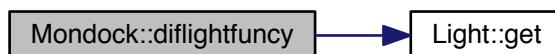
Voici le graphe d'appel pour cette fonction :



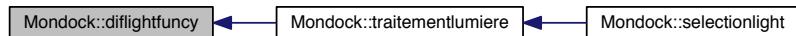
Voici le graphe des appelants de cette fonction :

**5.11.2.8 void Mondock ::diflightfuncy (int x) [private], [slot]**

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

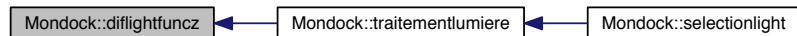


5.11.2.9 void Mondock : :diflightfuncz(int x) [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

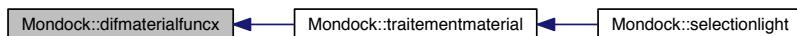


5.11.2.10 void Mondock : :difmaterialfuncx(int x) [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

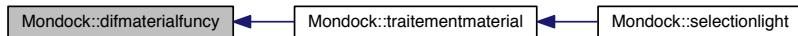


5.11.2.11 void Mondock ::difmaterialfuncy (int x) [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

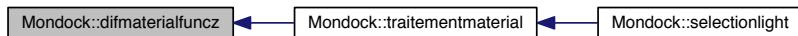


5.11.2.12 void Mondock ::difmaterialfuncz (int x) [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



5.11.2.13 void Mondock ::lumiereambiantejustx (int x) [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



5.11.2.14 void Mondock ::lumiereambiantejusty (int x) [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



5.11.2.15 void Mondock ::lumiereambiantejustz (int x) [private], [slot]

Voici le graphe d'appel pour cette fonction :

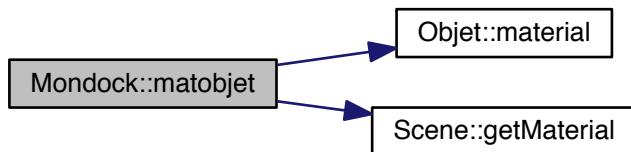


Voici le graphe des appelants de cette fonction :

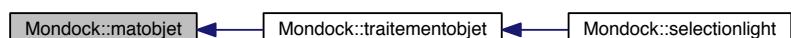


5.11.2.16 void Mondock ::matobjet (const QString & text) [private], [slot]

Voici le graphe d'appel pour cette fonction :

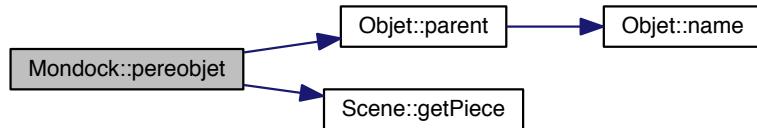


Voici le graphe des appelants de cette fonction :

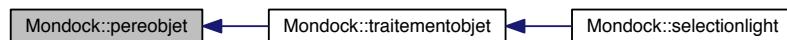


5.11.2.17 void Mondock ::pereobjet (const QString & text) [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



5.11.2.18 void Mondock ::poslightfuncx (double x) [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

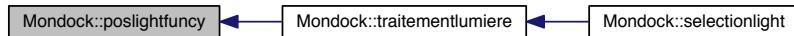


5.11.2.19 void Mondock ::poslightfuncy (double x) [private], [slot]

Voici le graphe d'appel pour cette fonction :

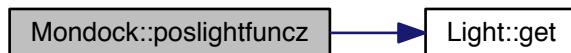


Voici le graphe des appels de cette fonction :

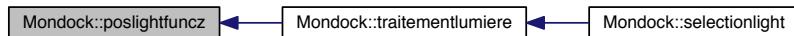


5.11.2.20 void Mondock ::poslightfuncz (double x) [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appels de cette fonction :

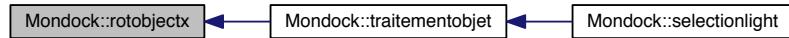


5.11.2.21 void Mondock ::rotobjectx (double x) [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appels de cette fonction :

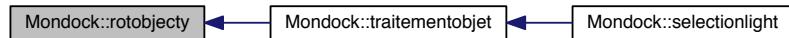


5.11.2.22 void Mondock ::rotobjecty (double x) [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appels de cette fonction :

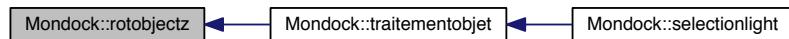


5.11.2.23 void Mondock ::rotobjectz (double x) [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appels de cette fonction :

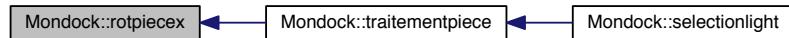


5.11.2.24 void Mondock ::rotpiececx (double x) [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appels de cette fonction :



5.11.2.25 void Mondock ::rotpiecey (double x) [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appels de cette fonction :

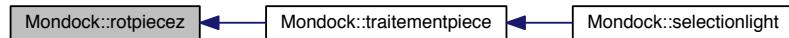


5.11.2.26 void Mondock ::rotpiecez (double x) [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appels de cette fonction :

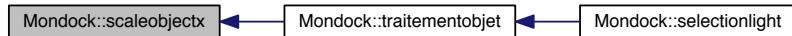


5.11.2.27 void Mondock::scaleobjectx(double x) [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appels de cette fonction :

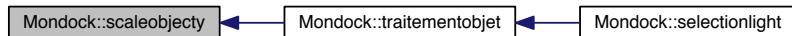


5.11.2.28 void Mondock::scaleobjecty(double x) [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appels de cette fonction :

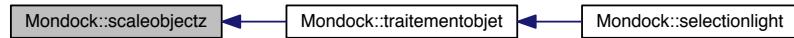


5.11.2.29 void Mondock ::scaleobjectz (double x) [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

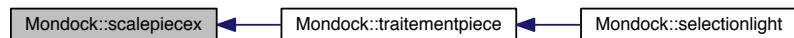


5.11.2.30 void Mondock ::scalepiececx (double x) [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

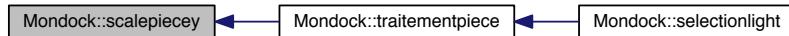


5.11.2.31 void Mondock ::scalepiecey (double x) [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appels de cette fonction :

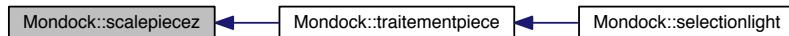


5.11.2.32 void Mondock ::scalepiecez (double x) [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appels de cette fonction :

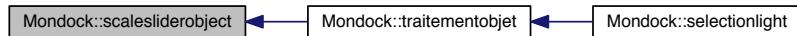


5.11.2.33 void Mondock ::scalesliderobject (int x) [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appels de cette fonction :

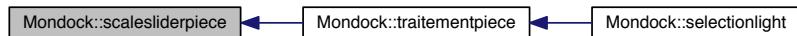


5.11.2.34 void Mondock ::scalesliderpiece (int x) [private], [slot]

Voici le graphe d'appel pour cette fonction :

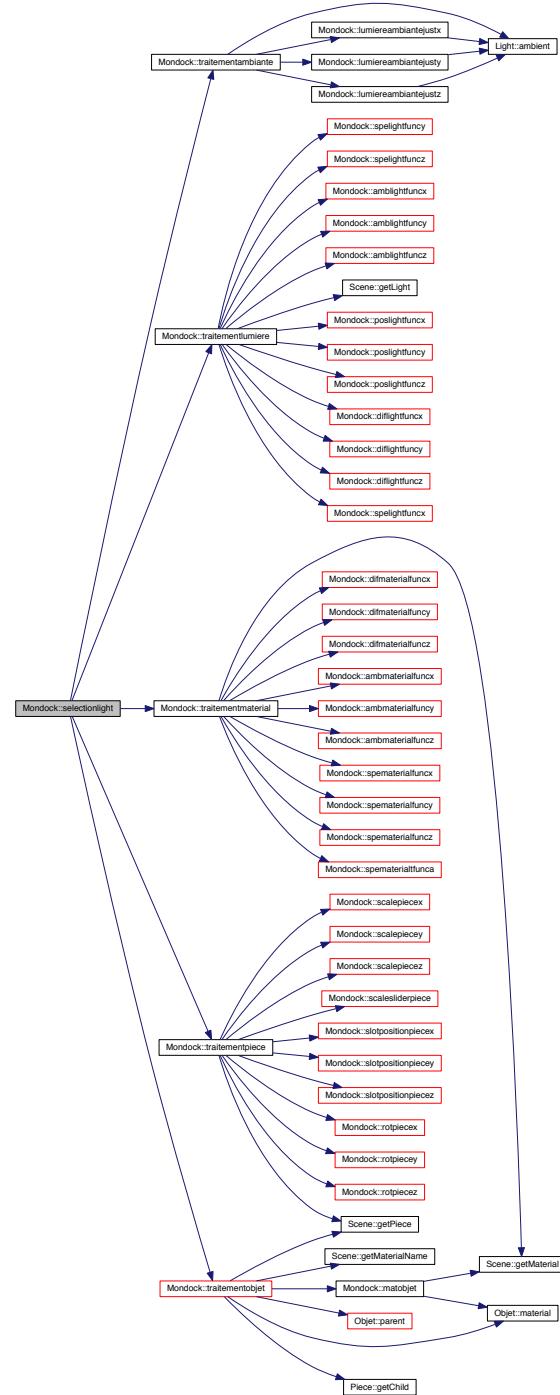


Voici le graphe des appels de cette fonction :



5.11.2.35 void Mondock::selectionlight() [private], [slot]

Voici le graphe d'appel pour cette fonction :



5.11.2.36 void Mondock ::slotpositionpiecex (int x) [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



5.11.2.37 void Mondock ::slotpositionpiecey (int x) [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



5.11.2.38 void Mondock ::slotpositionpiecez (int x) [private], [slot]

Voici le graphe d'appel pour cette fonction :

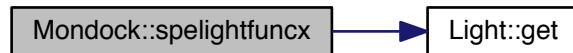


Voici le graphe des appels de cette fonction :

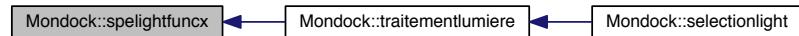


5.11.2.39 void Mondock ::spelightfuncx (int x) [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appels de cette fonction :

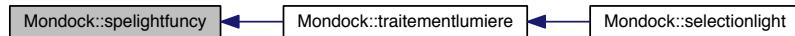


5.11.2.40 void Mondock::spelightfuncy (int x) [private], [slot]

Voici le graphe d'appel pour cette fonction :

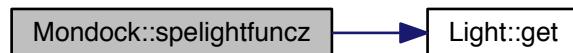


Voici le graphe des appelsants de cette fonction :

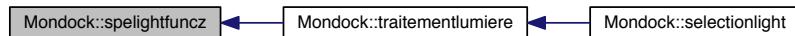


5.11.2.41 void Mondock::spelightfuncz (int x) [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelsants de cette fonction :

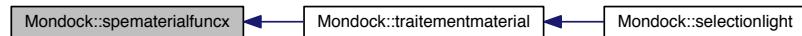


5.11.2.42 void Mondock::spematerialfuncx (int x) [private], [slot]

Voici le graphe d'appel pour cette fonction :

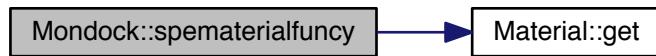


Voici le graphe des appelants de cette fonction :

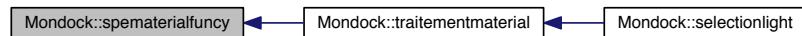


5.11.2.43 void Mondock::spematerialfuncy (int x) [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

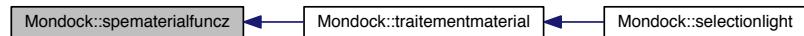


5.11.2.44 void Mondock ::spematerialfuncz (int x) [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



5.11.2.45 void Mondock ::spematerialfunca (int x) [private], [slot]

Voici le graphe d'appel pour cette fonction :

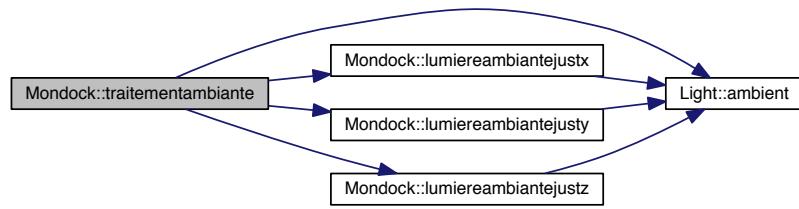


Voici le graphe des appelants de cette fonction :

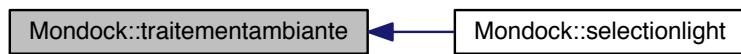


5.11.2.46 void Mondock ::traitementambiante() [private]

Voici le graphe d'appel pour cette fonction :

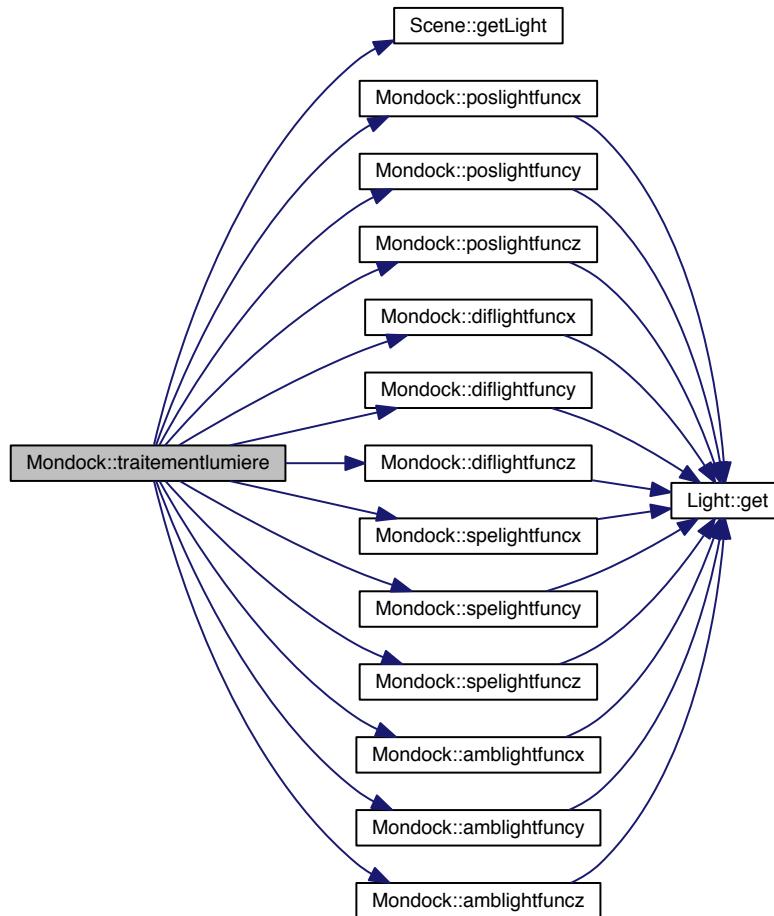


Voici le graphe des appelants de cette fonction :



5.11.2.47 void Mondock ::traitementlumiere() [private]

Voici le graphe d'appel pour cette fonction :

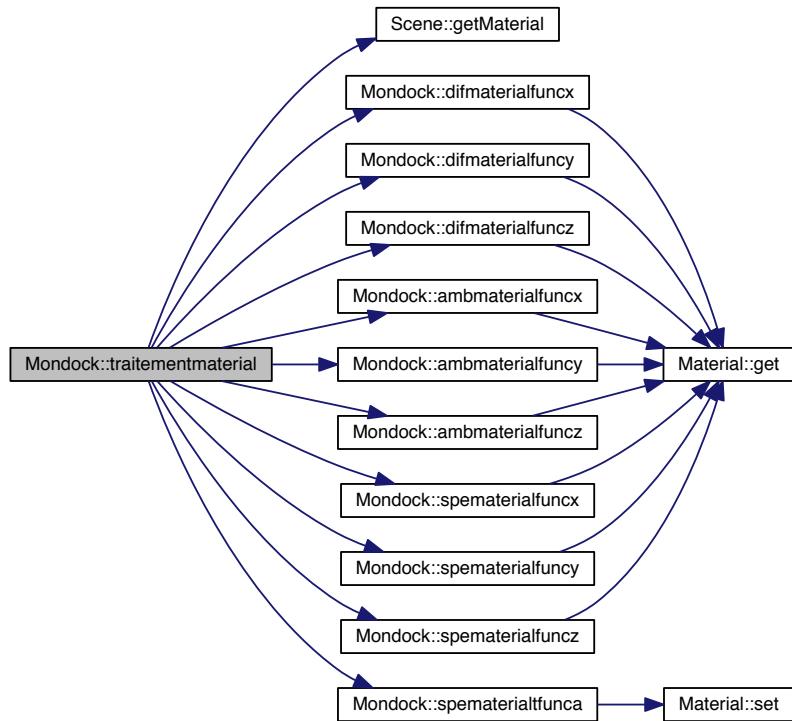


Voici le graphe des appels de cette fonction :



5.11.2.48 void Mondock ::traitementmaterial () [private]

Voici le graphe d'appel pour cette fonction :

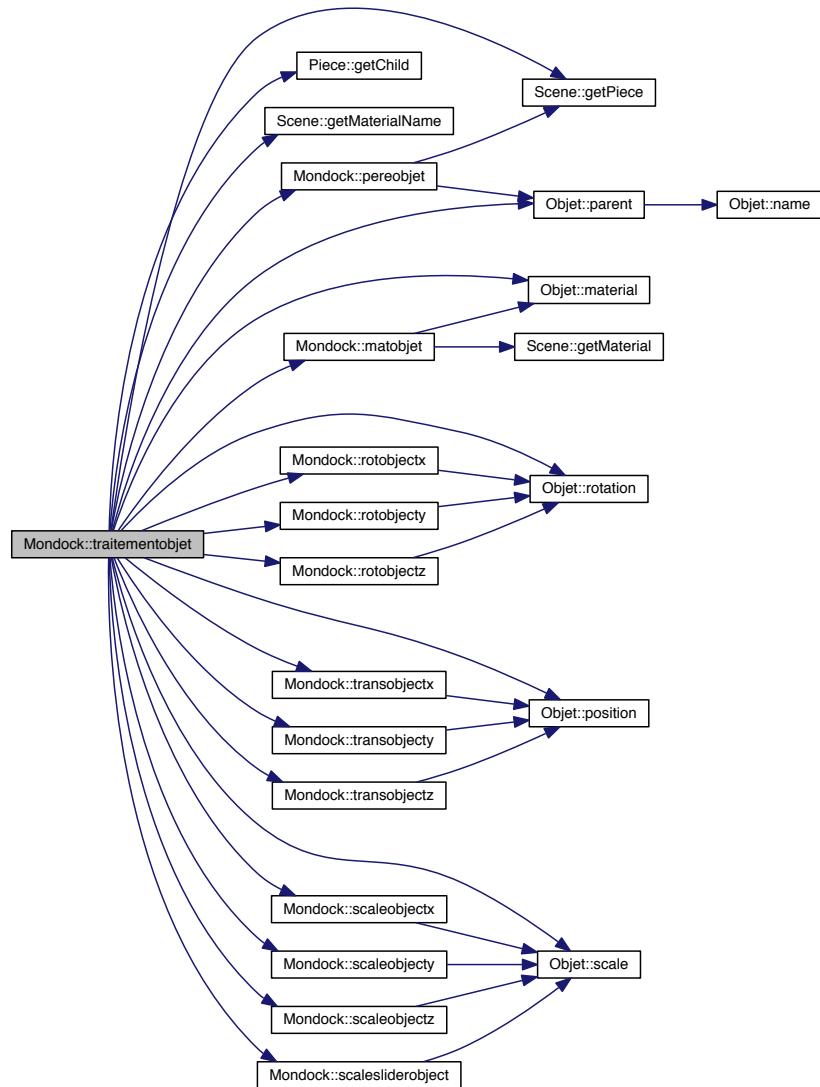


Voici le graphe des appels de cette fonction :

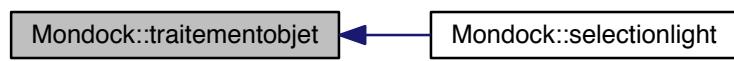


5.11.2.49 void Mondock ::traitementobjet() [private]

Voici le graphe d'appel pour cette fonction :

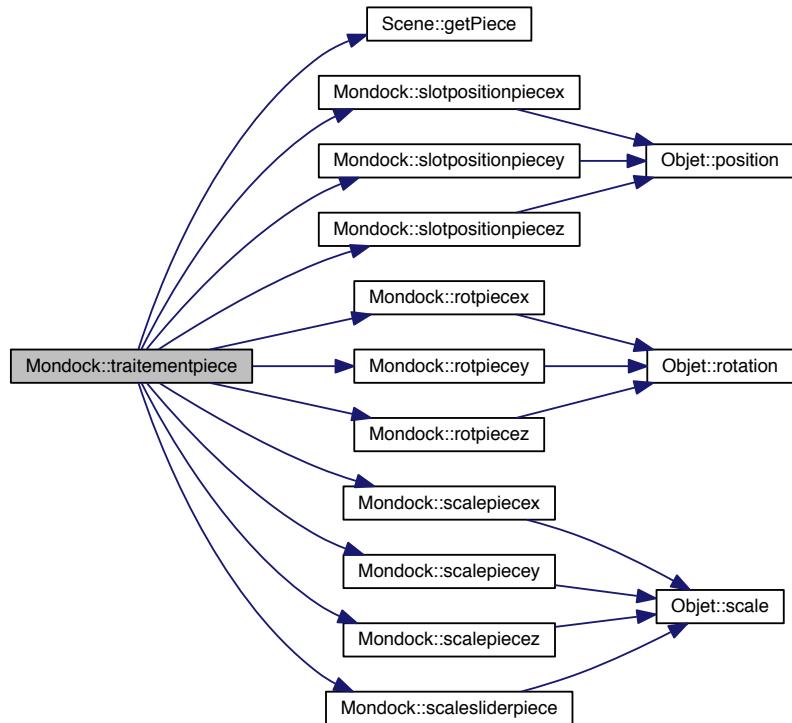


Voici le graphe des appels de cette fonction :



5.11.2.50 void Mondock ::traitementpiece () [private]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appels de cette fonction :

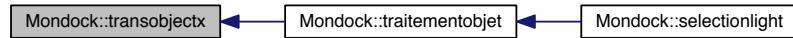


5.11.2.51 void Mondock ::transobjectx (double x) [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

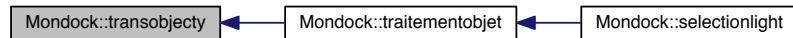


5.11.2.52 void Mondock ::transobjecty (double x) [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

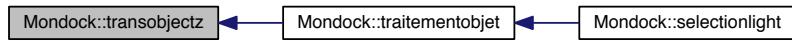


5.11.2.53 void Mondock ::transobjectz(double x) [private], [slot]

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



5.11.3 Documentation des données membres

5.11.3.1 QStandardItem* Mondock ::_itemmaterial

5.11.3.2 QStandardItem* Mondock ::_itempiece

5.11.3.3 Light* Mondock ::_light

5.11.3.4 Material* Mondock ::_materiaux

5.11.3.5 Objet* Mondock ::_objet

5.11.3.6 Piece* Mondock ::_piece

5.11.3.7 QDoubleSpinBox* Mondock ::boxobjetrotationx [private]

5.11.3.8 QDoubleSpinBox* Mondock ::boxobjetrotationy [private]

5.11.3.9 QDoubleSpinBox* Mondock ::boxobjetrotationz [private]

5.11.3.10 QDoubleSpinBox* Mondock ::boxobjetscalex [private]

5.11.3.11 QDoubleSpinBox* Mondock ::boxobjetscaley [private]

5.11.3.12 QDoubleSpinBox* Mondock ::boxobjetscalez [private]

5.11.3.13 QDoubleSpinBox* Mondock ::boxobjettransx [private]

5.11.3.14 QDoubleSpinBox* Mondock ::boxobjettransy [private]

5.11.3.15 QDoubleSpinBox* Mondock ::boxobjettransz [private]

- 5.11.3.16 **QComboBox*** Mondock : **:combomaterial** [private]
- 5.11.3.17 **QComboBox*** Mondock : **:comboparent** [private]
- 5.11.3.18 **QSpinBox*** Mondock : **:dimentionpiecex** [private]
- 5.11.3.19 **QSpinBox*** Mondock : **:dimentionpiecey** [private]
- 5.11.3.20 **QSpinBox*** Mondock : **:dimentionpiecez** [private]
- 5.11.3.21 **QStandardItemModel*** Mondock : **:dockmodele**
- 5.11.3.22 **Scene*** Mondock : **:dockscene**
- 5.11.3.23 **QTreeView*** Mondock : **:dockvue**
- 5.11.3.24 **QWidget*** Mondock : **:editionambiante** [private]
- 5.11.3.25 **QVariant** Mondock : **:elementSelectionneGParent** [private]
- 5.11.3.26 **QVariant** Mondock : **:elementSelectionneParent** [private]
- 5.11.3.27 **QModelIndex** Mondock : **:indexElementSelectionne** [private]
- 5.11.3.28 **QModelIndex** Mondock : **:indexgparent** [private]
- 5.11.3.29 **QModelIndex** Mondock : **:indexlightSelectionne** [private]
- 5.11.3.30 **QModelIndex** Mondock : **:indexmaterialSelectionne** [private]
- 5.11.3.31 **QModelIndex** Mondock : **:indexobjetSelectionne** [private]
- 5.11.3.32 **QModelIndex** Mondock : **:indexpieceSelectionne** [private]
- 5.11.3.33 **QHBoxLayout*** Mondock : **:labelamblayout** [private]
- 5.11.3.34 **QLabel*** Mondock : **:labelambx** [private]
- 5.11.3.35 **QLabel*** Mondock : **:labelamby** [private]
- 5.11.3.36 **QLabel*** Mondock : **:labelambz** [private]
- 5.11.3.37 **QHBoxLayout*** Mondock : **:labeldiflayout** [private]
- 5.11.3.38 **QLabel*** Mondock : **:labeldifx** [private]
- 5.11.3.39 **QLabel*** Mondock : **:labelidify** [private]
- 5.11.3.40 **QLabel*** Mondock : **:labelidifz** [private]
- 5.11.3.41 **QLabel*** Mondock : **:labelobjetproprmaterial** [private]
- 5.11.3.42 **QLabel*** Mondock : **:labelobjetproprparent** [private]
- 5.11.3.43 **QLabel*** Mondock : **:labelsliderobjetscale** [private]

- 5.11.3.44 QHBoxLayout* Mondock :**:labelspelayout** [private]
- 5.11.3.45 QLabel* Mondock :**:labelspex** [private]
- 5.11.3.46 QLabel* Mondock :**:labelspey** [private]
- 5.11.3.47 QLabel* Mondock :**:labelspez** [private]
- 5.11.3.48 QHBoxLayout* Mondock :**:layouteditionlighamb** [private]
- 5.11.3.49 QVBoxLayout* Mondock :**:layoutglobalscaleobjet** [private]
- 5.11.3.50 QHBoxLayout* Mondock :**:layoutobjetscaleall** [private]
- 5.11.3.51 QVBoxLayout* Mondock :**:layoutheadobjetprop** [private]
- 5.11.3.52 QHBoxLayout* Mondock :**:layouttabobjetpropcombo** [private]
- 5.11.3.53 QHBoxLayout* Mondock :**:layouttabobjetproplabel** [private]
- 5.11.3.54 QHBoxLayout* Mondock :**:layouttabobjetrotation** [private]
- 5.11.3.55 QHBoxLayout* Mondock :**:layouttabobjetscale** [private]
- 5.11.3.56 QHBoxLayout* Mondock :**:layouttabobjettrans** [private]
- 5.11.3.57 QHBoxLayout* Mondock :**:lightamblayout** [private]
- 5.11.3.58 QHBoxLayout* Mondock :**:lightdiflayout** [private]
- 5.11.3.59 QDoubleSpinBox* Mondock :**:lightdoublespinboxx** [private]
- 5.11.3.60 QDoubleSpinBox* Mondock :**:lightdoublespinboxy** [private]
- 5.11.3.61 QDoubleSpinBox* Mondock :**:lightdoublespinboxz** [private]
- 5.11.3.62 QVariant Mondock :**:lightselectioner** [private]
- 5.11.3.63 QHBoxLayout* Mondock :**:lightspelayout** [private]
- 5.11.3.64 QVariant Mondock :**:materialselectioner** [private]
- 5.11.3.65 QSpinBox* Mondock :**:materialspinboxambx** [private]
- 5.11.3.66 QSpinBox* Mondock :**:materialspinboxamby** [private]
- 5.11.3.67 QSpinBox* Mondock :**:materialspinboxambz** [private]
- 5.11.3.68 QSpinBox* Mondock :**:materialspinboxdfix** [private]
- 5.11.3.69 QSpinBox* Mondock :**:materialspinboxdify** [private]
- 5.11.3.70 QSpinBox* Mondock :**:materialspinboxdifz** [private]
- 5.11.3.71 QSpinBox* Mondock :**:materialspinboxspea** [private]

- 5.11.3.72 **QSpinBox* Mondock::materialspinboxspex** [private]
- 5.11.3.73 **QSpinBox* Mondock::materialspinboxspey** [private]
- 5.11.3.74 **QSpinBox* Mondock::materialspinboxspez** [private]
- 5.11.3.75 **QStandardItemModel* Mondock::modelmaterial**
- 5.11.3.76 **QStandardItemModel* Mondock::modelpiece** [private]
- 5.11.3.77 **QVariant Mondock::objetselectioner** [private]
- 5.11.3.78 **QHBoxLayout* Mondock::piecedimlayout** [private]
- 5.11.3.79 **QHBoxLayout* Mondock::pieceposilayout** [private]
- 5.11.3.80 **QHBoxLayout* Mondock::piecerotatelayout** [private]
- 5.11.3.81 **QVBoxLayout* Mondock::piecescalelayout** [private]
- 5.11.3.82 **QVariant Mondock::pieceselectioner** [private]
- 5.11.3.83 **QSpinBox* Mondock::positionpiecex** [private]
- 5.11.3.84 **QSpinBox* Mondock::positionpiecey** [private]
- 5.11.3.85 **QSpinBox* Mondock::positionpiecez** [private]
- 5.11.3.86 **QDoubleSpinBox* Mondock::rotspinboxx** [private]
- 5.11.3.87 **QDoubleSpinBox* Mondock::rotspinboxy** [private]
- 5.11.3.88 **QDoubleSpinBox* Mondock::rotspinboxz** [private]
- 5.11.3.89 **QLabel* Mondock::scalelabel** [private]
- 5.11.3.90 **QSlider* Mondock::scaleslider** [private]
- 5.11.3.91 **QHBoxLayout* Mondock::scalesliderlayout** [private]
- 5.11.3.92 **QHBoxLayout* Mondock::scalespinboxlayout** [private]
- 5.11.3.93 **QDoubleSpinBox* Mondock::scalespinboxx** [private]
- 5.11.3.94 **QDoubleSpinBox* Mondock::scalespinboxy** [private]
- 5.11.3.95 **QDoubleSpinBox* Mondock::scalespinboxz** [private]
- 5.11.3.96 **QItemSelectionModel* Mondock::selection** [private]
- 5.11.3.97 **QSlider* Mondock::sliderambx** [private]
- 5.11.3.98 **QSlider* Mondock::slideramby** [private]
- 5.11.3.99 **QSlider* Mondock::sliderambz** [private]

- 5.11.3.100 **QSlider* Mondock::sliderdifx** [private]
- 5.11.3.101 **QSlider* Mondock::sliderdify** [private]
- 5.11.3.102 **QSlider* Mondock::sliderdifz** [private]
- 5.11.3.103 **QSlider* Mondock::sliderobjetscale** [private]
- 5.11.3.104 **QSlider* Mondock::sliderspez** [private]
- 5.11.3.105 **QSlider* Mondock::sliderspey** [private]
- 5.11.3.106 **QSlider* Mondock::sliderspez** [private]
- 5.11.3.107 **QSpinBox* Mondock::spinboxlumiereambiantex** [private]
- 5.11.3.108 **QSpinBox* Mondock::spinboxlumiereambiantey** [private]
- 5.11.3.109 **QSpinBox* Mondock::spinboxlumiereambiantez** [private]
- 5.11.3.110 **QVBoxLayout* Mondock::tabamblayout** [private]
- 5.11.3.111 **QVBoxLayout* Mondock::tabdiflayout** [private]
- 5.11.3.112 **QTabWidget* Mondock::tablight** [private]
- 5.11.3.113 **QWidget* Mondock::tablightamb** [private]
- 5.11.3.114 **QWidget* Mondock::tablightdif** [private]
- 5.11.3.115 **QWidget* Mondock::tablightpos** [private]
- 5.11.3.116 **QHBoxLayout* Mondock::tablightposlayout** [private]
- 5.11.3.117 **QWidget* Mondock::tablightspe** [private]
- 5.11.3.118 **QTabWidget* Mondock::tabmaterial** [private]
- 5.11.3.119 **QWidget* Mondock::tabmaterialamb** [private]
- 5.11.3.120 **QHBoxLayout* Mondock::tabmaterialamblayout** [private]
- 5.11.3.121 **QWidget* Mondock::tabmaterialdif** [private]
- 5.11.3.122 **QHBoxLayout* Mondock::tabmaterialdiflayout** [private]
- 5.11.3.123 **QWidget* Mondock::tabmaterialspe** [private]
- 5.11.3.124 **QHBoxLayout* Mondock::tabmaterialsplayout** [private]
- 5.11.3.125 **QTabWidget* Mondock::tabobjet** [private]
- 5.11.3.126 **QWidget* Mondock::tabobjetprop** [private]
- 5.11.3.127 **QWidget* Mondock::tabobjetrotation** [private]

5.11.3.128 QWidget* Mondock::tabobjetscale [private]

5.11.3.129 QWidget* Mondock::tabobjettrans [private]

5.11.3.130 QTabWidget* Mondock::tabpiece [private]

5.11.3.131 QVBoxLayout* Mondock::tabsplayout [private]

5.11.3.132 QWidget* Mondock::widgetpiecedim [private]

5.11.3.133 QWidget* Mondock::widgetpieceposi [private]

5.11.3.134 QWidget* Mondock::widgetpiicerotate [private]

5.11.3.135 QWidget* Mondock::widgetpiecescale [private]

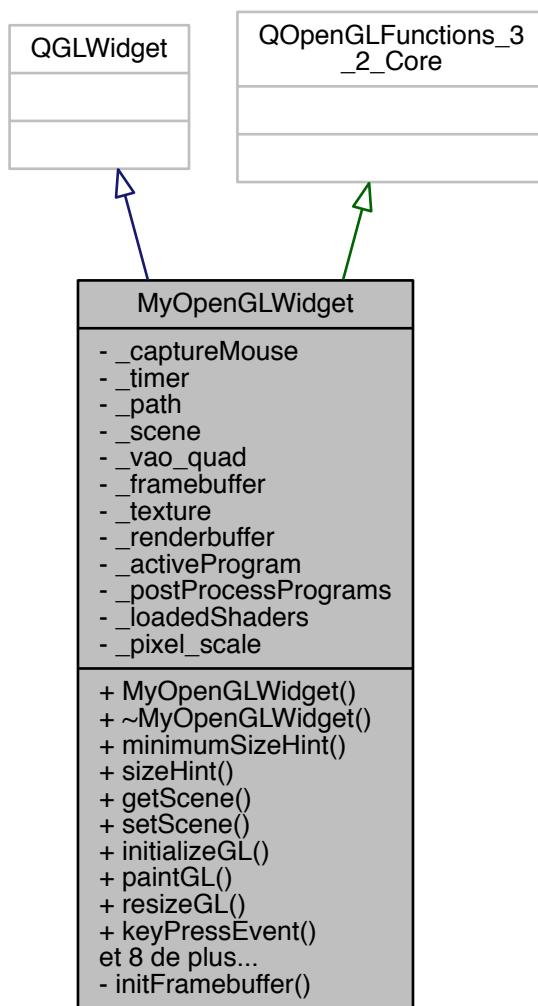
La documentation de cette classe a été générée à partir des fichiers suivants :

- gui/[mondock.hpp](#)
- gui/[mondock.cpp](#)

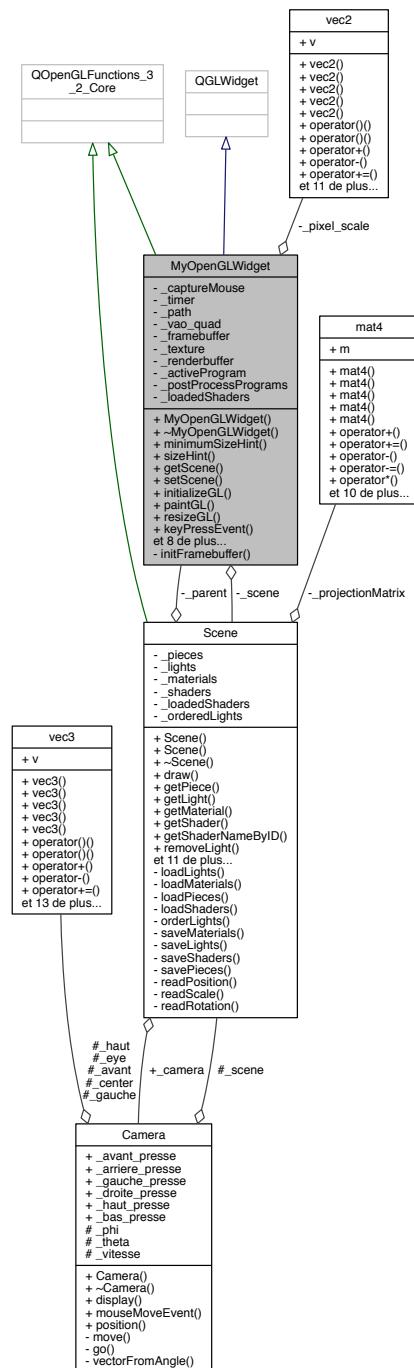
5.12 Référence de la classe MyOpenGLWidget

```
#include <MyOpenGLWidget.hpp>
```

Graphe d'héritage de MyOpenGLWidget :



Graphe de collaboration de MyOpenGLWidget :



Fonctions membres publiques

- [MyOpenGLWidget](#) (const QGLFormat &format, QWidget *parent=0, const QString &path="", const QGLWidget *shareWidget=0, Qt::WindowFlags f=0)
- [~MyOpenGLWidget \(\)](#)
- [QSize minimumSizeHint \(\)](#) const
- [QSize sizeHint \(\)](#) const
- [Scene * getScene \(\)](#)
- [void setScene \(Scene *scene\)](#)
- [virtual void initializeGL \(\)](#)

```

— virtual void paintGL ()
— virtual void resizeGL (int width, int height)
— virtual void keyPressEvent (QKeyEvent *event)
— virtual void keyReleaseEvent (QKeyEvent *event)
— virtual void mouseMoveEvent (QMouseEvent *event)
— virtual void mousePressEvent (QMouseEvent *event)
— void useShader (const QString &name)
— void addShader (const QString &name, const QString &vertex, const QString &fragment)
— QStringList getShaderNames ()
— void loadShaders (const QDomElement &postProcess)
— void saveShaders (QDomElement &root, QDomDocument &doc) const

```

Fonctions membres privées

- void **initFramebuffer** (int width, int height)

Attributs privés

```

— bool _captureMouse
— QTimer * _timer
— QString _path
— Scene * _scene
— GLuint _vao_quad
— GLuint _framebuffer
— GLuint _texture
— GLuint _renderbuffer
— QOpenGLShaderProgram * _activeProgram
— QMap<QString,
      QOpenGLShaderProgram *> _postProcessPrograms
— QMap<QString, QOpenGLShader *> _loadedShaders
— vec2 _pixel_scale

```

5.12.1 Documentation des constructeurs et destructeur

5.12.1.1 MyOpenGLWidget : :MyOpenGLWidget (const QGLFormat & *format*, QWidget * *parent* = 0, const QString & *path* = " ", const QGLWidget * *shareWidget* = 0, Qt : :WindowFlags *f* = 0)

5.12.1.2 MyOpenGLWidget : :~MyOpenGLWidget ()

5.12.2 Documentation des fonctions membres

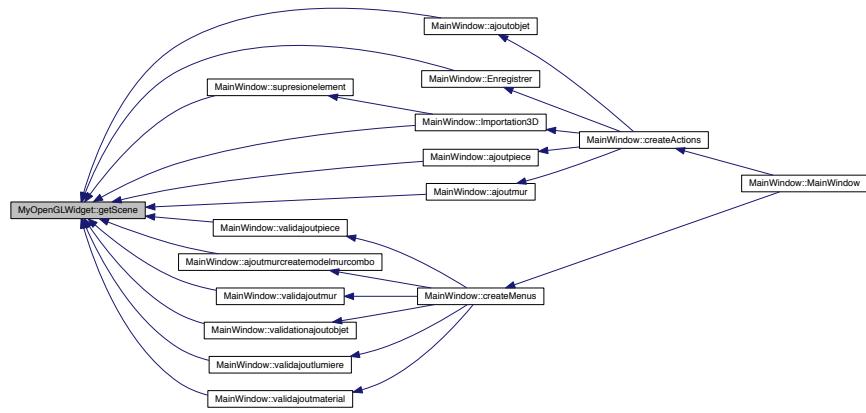
5.12.2.1 void MyOpenGLWidget : :addShader (const QString & *name*, const QString & *vertex*, const QString & *fragment*)

Voici le graphe des appels de cette fonction :



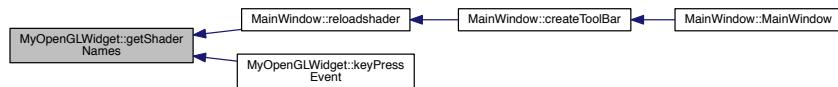
5.12.2.2 Scene * MyOpenGLWidget ::getScene ()

Voici le graphe des appels de cette fonction :



5.12.2.3 QStringList MyOpenGLWidget ::getShaderNames ()

Voici le graphe des appels de cette fonction :



5.12.2.4 void MyOpenGLWidget : initFramebuffer (int width, int height) [private]

Voici le graphe des appels de cette fonction :



5.12.2.5 void MyOpenGLWidget :: initializeGL() [virtual]

5.12.2.6 void MyOpenGLWidget : :keyPressEvent (QKeyEvent * event) [virtual]

Voici le graphe d'appel pour cette fonction :



5.12.2.7 void MyOpenGLWidget : :keyReleaseEvent (QKeyEvent * event) [virtual]

5.12.2.8 void MyOpenGLWidget : :loadShaders (const QDomElement & postProcess)

Voici le graphe d'appel pour cette fonction :



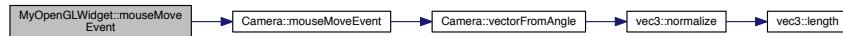
Voici le graphe des appels de cette fonction :



5.12.2.9 QSize MyOpenGLWidget : :minimumSizeHint () const

5.12.2.10 void MyOpenGLWidget : :mouseMoveEvent (QMouseEvent * event) [virtual]

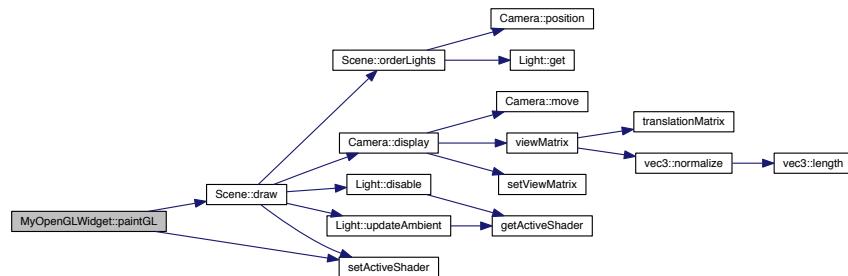
Voici le graphe d'appel pour cette fonction :



5.12.2.11 void MyOpenGLWidget ::mousePressEvent (QMouseEvent * event) [virtual]

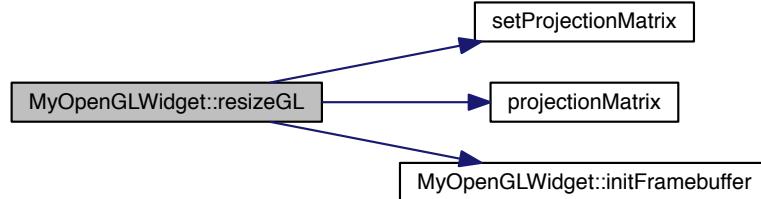
5.12.2.12 void MyOpenGLWidget ::paintGL () [virtual]

Voici le graphe d'appel pour cette fonction :



5.12.2.13 void MyOpenGLWidget ::resizeGL (int width, int height) [virtual]

Voici le graphe d'appel pour cette fonction :



5.12.2.14 void MyOpenGLWidget ::saveShaders (QDomElement & root, QDomDocument & doc) const

Voici le graphe des appels de cette fonction :



5.12.2.15 void MyOpenGLWidget ::setScene (Scene * scene)

5.12.2.16 QSize MyOpenGLWidget ::sizeHint () const

5.12.2.17 void MyOpenGLWidget ::useShader (const QString & name)

Voici le graphe des appels de cette fonction :



5.12.3 Documentation des données membres

5.12.3.1 QOpenGLShaderProgram* MyOpenGLWidget ::_activeProgram [private]

5.12.3.2 bool MyOpenGLWidget ::_captureMouse [private]

5.12.3.3 GLuint MyOpenGLWidget ::_framebuffer [private]

5.12.3.4 QMap<QString, QOpenGLShader *> MyOpenGLWidget ::_loadedShaders [private]

5.12.3.5 QString MyOpenGLWidget ::_path [private]

5.12.3.6 vec2 MyOpenGLWidget ::_pixel_scale [private]

5.12.3.7 QMap<QString, QOpenGLShaderProgram *> MyOpenGLWidget ::_postProcessPrograms [private]

5.12.3.8 GLuint MyOpenGLWidget ::_renderbuffer [private]

5.12.3.9 Scene* MyOpenGLWidget ::_scene [private]

5.12.3.10 GLuint MyOpenGLWidget ::_texture [private]

5.12.3.11 QTimer* MyOpenGLWidget ::_timer [private]

5.12.3.12 GLuint MyOpenGLWidget ::_vao_quad [private]

La documentation de cette classe a été générée à partir des fichiers suivants :

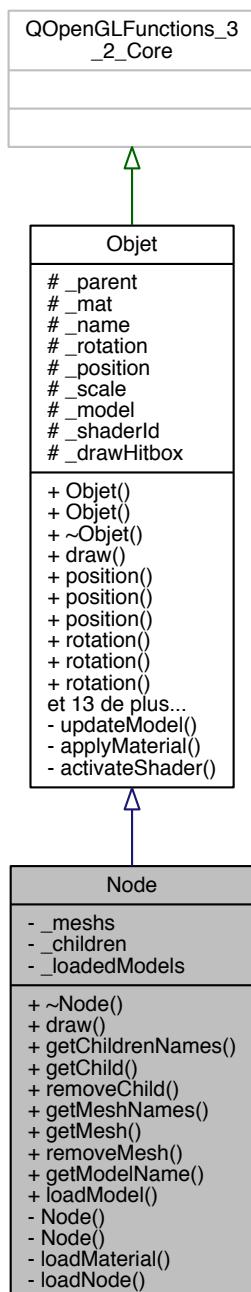
- gui/[MyOpenGLWidget.hpp](#)
- gui/[MyOpenGLWidget.cpp](#)

5.13 Référence de la classe Node

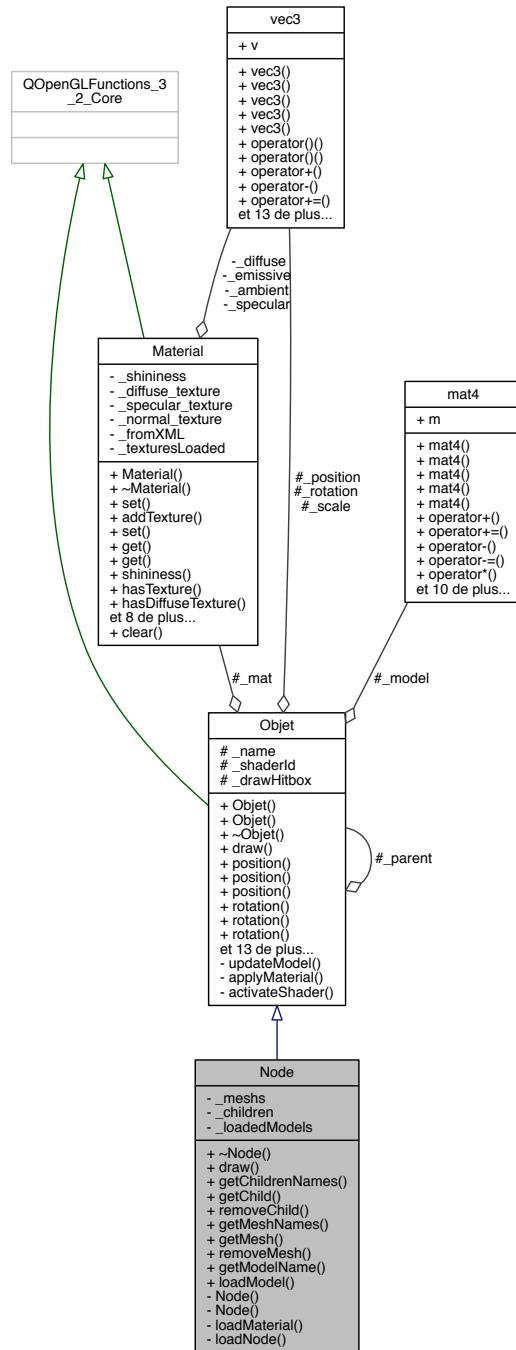
Contient un node de modèle 3D.

```
#include <node.hpp>
```

Graphe d'héritage de Node :



Graphe de collaboration de Node :



Fonctions membres publiques

- `~Node ()`
Destructeur.
- `void draw ()`
Affichage de l'Objet.
- `QStringList getChildrenNames () const`
Retourne la liste des noms des Nodes fils.
- `Node * getChild (const QString &name)`

- Retourne le `Node` enfant à partir de son nom.
- void `removeChild` (const QString &`name`)
supprime un `Node` fils
- QStringList `getMeshNames` () const
Retourne la liste des noms des `Mesh` de ce `Node`.
- Mesh * `getMesh` (const QString &`name`)
Retourne le `Mesh` ayant le nom donné
- void `removeMesh` (const QString &`name`)
Supprime le `Mesh` portant le nom.
- QString `getModelName` () const
Récupère le nom du modèle chargé

Fonctions membres publiques statiques

- static Node * `loadModel` (const QString &path, Scene *`scene`)
Charge le modèle.

Fonctions membres privées

- Node ()
Constructeur par défaut.
- Node (const Node &`node`)
Constructeur par recopie.

Fonctions membres privées statiques

- static Material * `loadMaterial` (const aiMaterial *`mtl`)
Charge un `Material` depuis une `aiMaterial` d'ASSIMP.
- static Node * `loadNode` (const aiNode *`node`, const aiScene *`pScene`, Scene *`scene`)
Charge un `aiNode`.

Attributs privés

- QList< Mesh * > _meshes
- QMap< QString, Node * > _children

Attributs privés statiques

- static QMultiMap< QString, Node * > _loadedModels

Membres hérités additionnels

5.13.1 Description détaillée

Contient un node de modèle 3D.

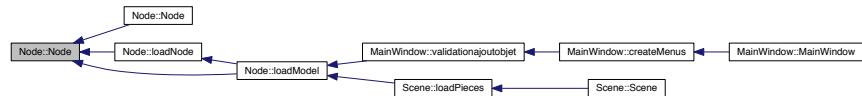
Un node possède un ensemble de meshes et de `Node` fils

5.13.2 Documentation des constructeurs et destructeur

5.13.2.1 Node : Node () [private]

Constructeur par défaut.

Voici le graphe des appels de cette fonction :



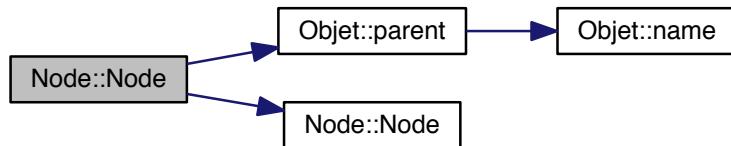
5.13.2.2 Node::Node (const Node & node) [private]

Constructeur par recopie.

Paramètres

<code>node</code>	Node à recopier
-------------------	-----------------

Voici le graphe d'appel pour cette fonction :



5.13.2.3 Node::~Node ()

Destructeur.

Détruit les Nodes et Mesh fils

5.13.3 Documentation des fonctions membres

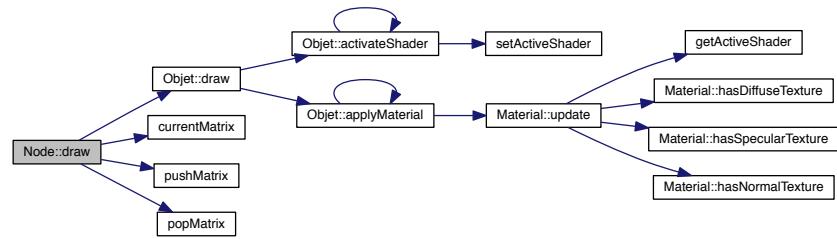
5.13.3.1 void Node::draw() [virtual]

Affichage de l'Objet.

Active le shader de l'Objet et applique le material

Réimplémentée à partir de [Objet](#).

Voici le graphe d'appel pour cette fonction :



5.13.3.2 Node * Node :: getChild (const QString & name)

Retourne le **Node** enfant à partir de son nom.

Paramètres

<i>name</i>	nom du Node enfant
-------------	---------------------------

Renvoie

retourne NULL si aucun enfant n'existe avec ce nom

Voici le graphe des appels de cette fonction :



5.13.3.3 QStringList Node :: getChildrenNames () const

Retourne la liste des noms des Nodes fils.

Renvoie

liste des noms des Nodes fils

Voici le graphe des appels de cette fonction :



5.13.3.4 Mesh * Node :: getMesh (const QString & name)

Retourne le **Mesh** ayant le nom donné

Paramètres

<i>name</i>	nom du Mesh
-------------	-----------------------------

Renvoie

NULL si le [Mesh](#) n'appartient pas à ce [Node](#)

Voici le graphe d'appel pour cette fonction :

**5.13.3.5 QStringList Node ::getMeshNames () const**

Retourne la liste des noms des [Mesh](#) de ce [Node](#).

Renvoie

liste des noms des [Mesh](#)

5.13.3.6 QString Node ::getModelName () const

Récupère le nom du modèle chargé

Paramètres

<i>node</i>	Node dont on veut le nom du modèle
-------------	--

Renvoie

nom du fichier charger

Avertissement

Ne fonctionne que pour les [Node](#) racines

Voici le graphe des appels de cette fonction :

**5.13.3.7 Material * Node ::loadMaterial (const aiMaterial * mtl) [static], [private]**

Charge un [Material](#) depuis une aiMaterial d'ASSIMP.

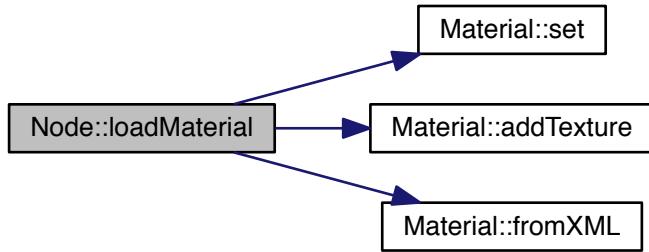
Paramètres

<i>mtl</i>	material à charger
------------	--------------------

Renvoie

Material créé

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appels de cette fonction :



5.13.3.8 Node * Node ::loadModel (const QString & path, Scene * scene) [static]

Charge le modèle.

Retourne le `Node` racine représentant le modèle et ajoute les matériaux à la scène

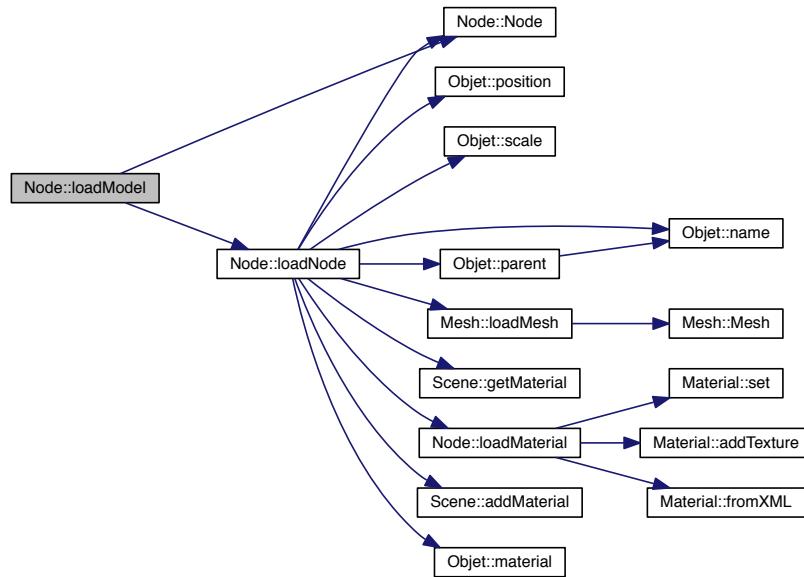
Paramètres

<i>path</i>	nom du fichier à charger
<i>scene</i>	Scene qui recevra les matériaux chargés

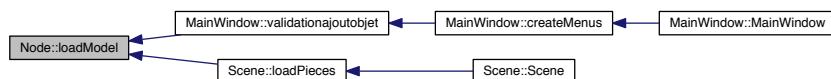
Renvoie

[Node](#) racine du modèle

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appels de cette fonction :



5.13.3.9 `Node * Node ::loadNode (const aiNode * node, const aiScene * pScene, Scene * scene) [static], [private]`

Charge un aiNode.

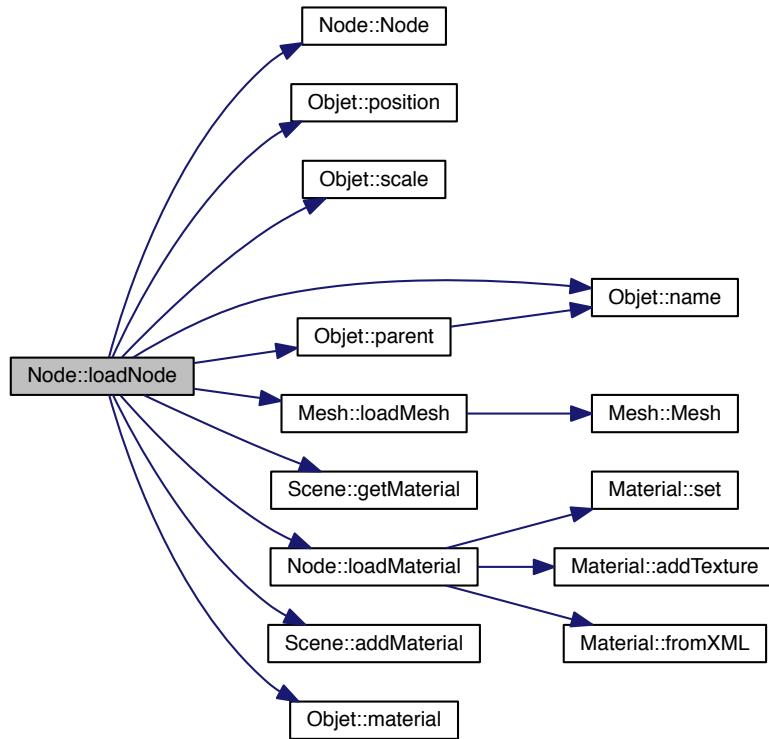
Paramètres

<code>node</code>	aiNode à charger
<code>pScene</code>	aiScene contenant les textures, matériaux et mesh
<code>scene</code>	Scene à laquelle ajouter les matériaux chargés

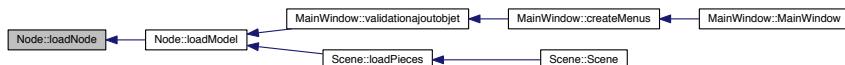
Renvoie

[Node](#) proprement chargé

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appels de cette fonction :



5.13.3.10 void Node ::removeChild (const QString & name)

supprime un [Node](#) fils

Paramètres

<i>name</i>	nom du Node à supprimer
-------------	---

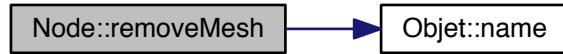
5.13.3.11 void Node ::removeMesh (const QString & name)

Supprime le [Mesh](#) portant le nom.

Paramètres

<i>name</i>	nom du Mesh à supprimer
-------------	---

Voici le graphe d'appel pour cette fonction :



5.13.4 Documentation des données membres

5.13.4.1 QMap<QString, Node *> Node ::_children [private]

Fils du [Node](#)

5.13.4.2 QMultiMap< QString, Node * > Node ::_loadedModels [static], [private]

Modèles déjà chargés, reconstruit un ensemble de [Node](#) par recopie

5.13.4.3 QList<Mesh *> Node ::_meshs [private]

Liste des [Mesh](#) de ce node

La documentation de cette classe a été générée à partir des fichiers suivants :

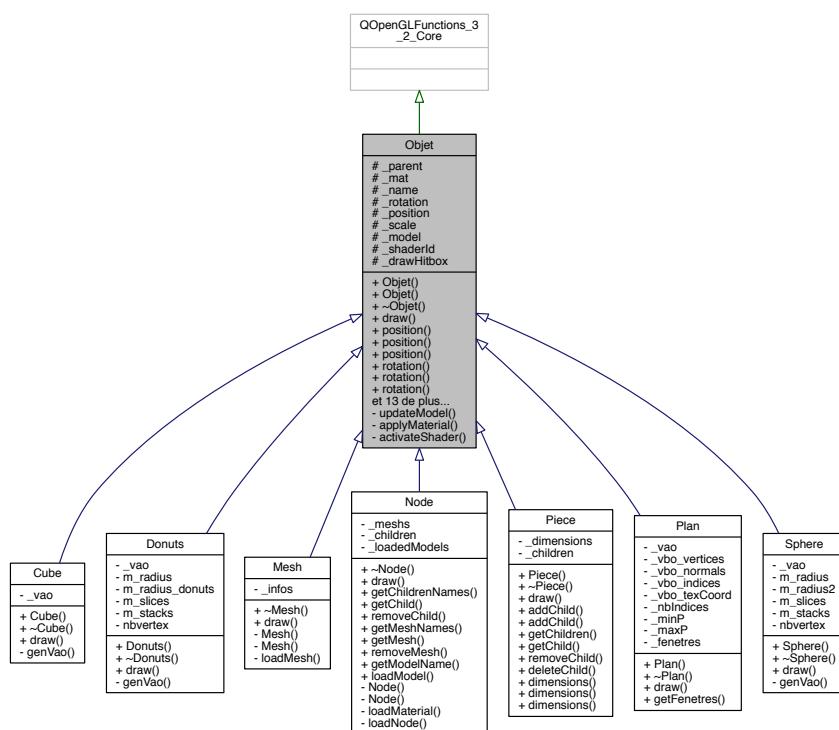
- objets/node.hpp
- objets/node.cpp

5.14 Référence de la classe Objet

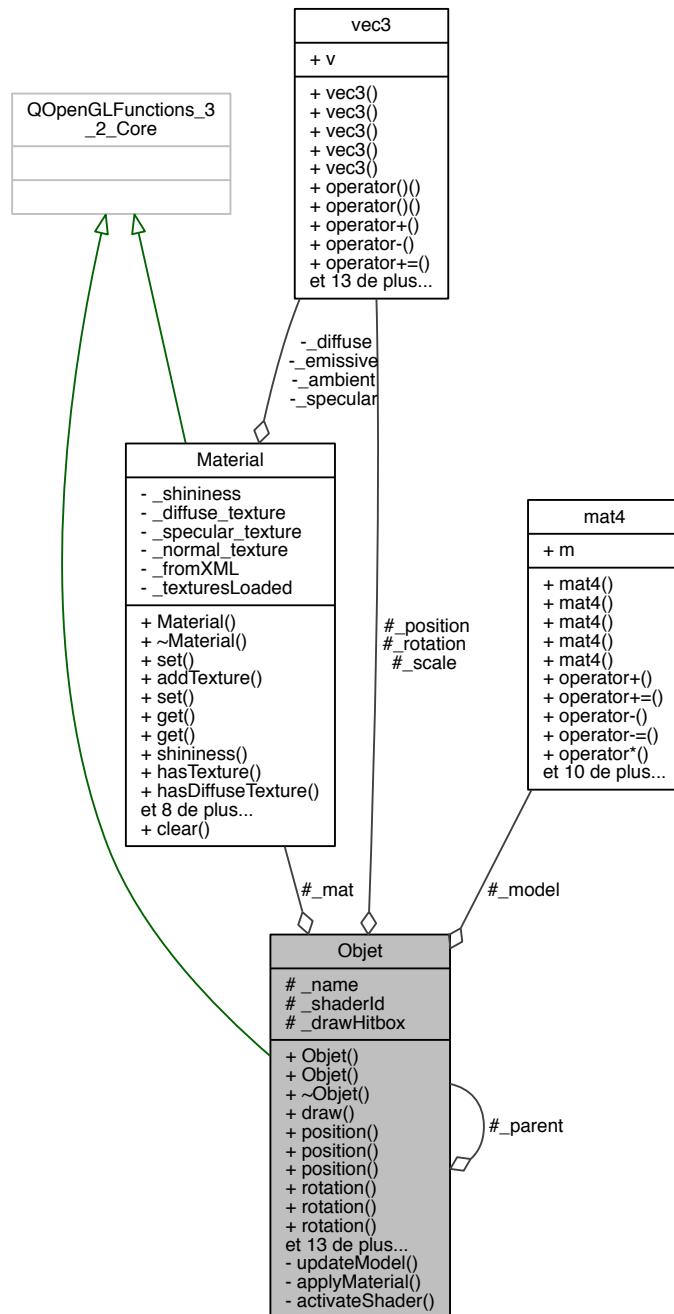
Classe de base affichable.

```
#include <objet.hpp>
```

Graphe d'héritage de Objet :



Graphe de collaboration de Objet :



Fonctions membres publiques

- `Objet (const QString &name=QString(), Material *mat=NULL, vec3 rotation=vec3(), vec3 position=vec3(), Objet *parent=NULL)`
Constructeur.
- `Objet (const Objet &o)`
Constructeur par recopie.
- `virtual ~Objet ()`
Destructeur.

- virtual void **draw** ()

Affichage de l'Objet.
- **vec3 & position** ()

Récupère la position de l'Objet.
- **vec3 position** () const

Récupère la position de l'Objet.
- void **position** (**vec3** p)

Change la position de l'Objet.
- **vec3 & rotation** ()

Récupère la rotation de l'Objet.
- **vec3 rotation** () const

Récupère la rotation de l'Objet.
- void **rotation** (**vec3** r)

Change la rotation de l'Objet.
- **vec3 & scale** ()

Récupère les facteurs de scale de l'objet.
- **vec3 scale** () const

Récupère les facteurs de scale de l'objet.
- void **scale** (**vec3** scale)

Modifie le scale de l'objet.
- void **shaderId** (GLuint s)

Change l'id du shader.
- GLuint **shaderId** () const

Retourne l'id du shader qu'utilise l'Objet.
- void **drawHitbox** (bool b)

Affiche les Hitbox des objets capables d'en afficher.
- bool **drawHitbox** () const

Affiche la Hitbox de l'objet.
- **Material * material** ()

Récupère le material de l'Objet.
- void **material** (**Material** *m)

Change le Material de l'Objet.
- void **parent** (Objet *o)

Change l'Objet père.
- **Objet * parent** ()

Récupère l'Objet père.
- const QString & **name** () const

Récupère le nom de l'Objet.
- void **name** (const QString &n)

Change le nom de l'Objet.

Attributs protégés

- **Objet * __parent**
- **Material * __mat**
- **QString __name**
- **vec3 __rotation**
- **vec3 __position**
- **vec3 __scale**
- **mat4 __model**
- **GLuint __shaderId**
- **bool __drawHitbox**

Fonctions membres privées

- void **updateModel** ()

Met à jour la matrice de model.
- void **applyMaterial** ()

Transmet les information du Material au shader.
- void **activateShader** ()

Change le shader actif.

5.14.1 Description détaillée

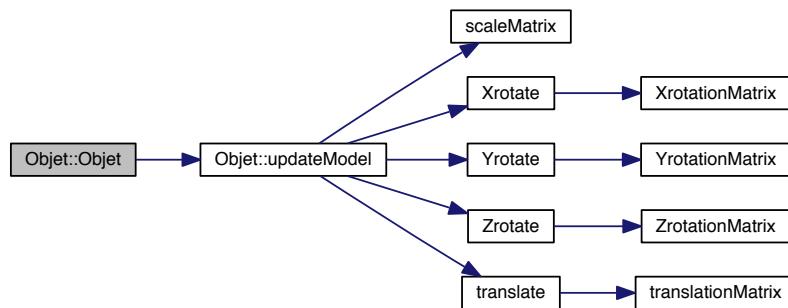
Classe de base affichable.

5.14.2 Documentation des constructeurs et destructeur

5.14.2.1 `Objet : :Objet (const QString & name = QString () , Material * mat = NULL, vec3 rotation = vec3 () , vec3 position = vec3 () , Objet * parent = NULL)`

Constructeur.

Voici le graphe d'appel pour cette fonction :



5.14.2.2 `Objet : :Objet (const Objet & o)`

Constructeur par recopie.

Paramètres

<code>o</code>	objet à copier
----------------	----------------

5.14.2.3 `virtual Objet : :~Objet () [inline], [virtual]`

Destructeur.

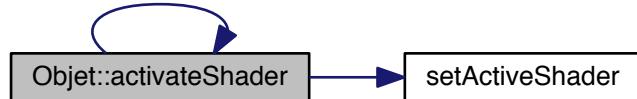
5.14.3 Documentation des fonctions membres

5.14.3.1 `void Objet : :activateShader () [private]`

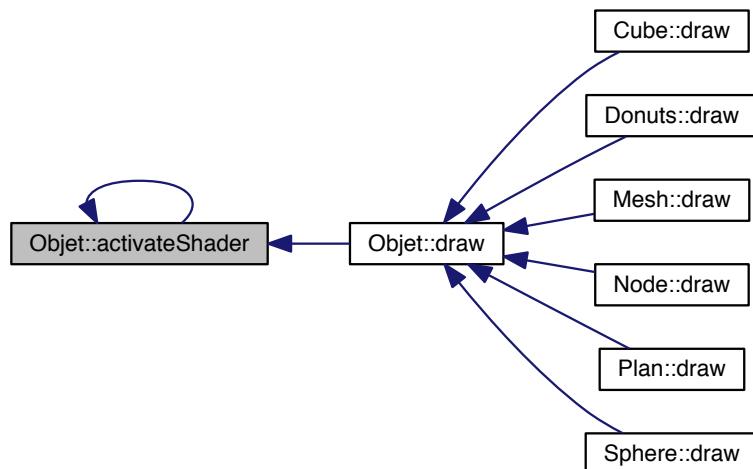
Change le shader actif.

Active le shader de l'[Objet](#), reporte les matrices de view et projection

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

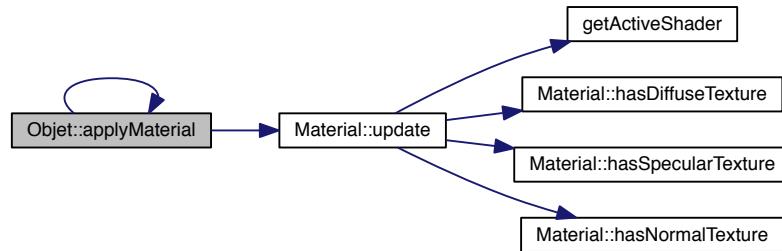


5.14.3.2 void Objet::applyMaterial() [private]

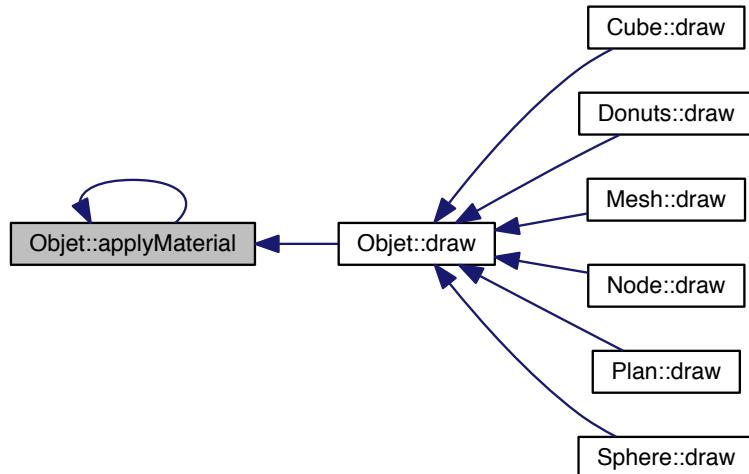
Transmet les informations du [Material](#) au shader.

Si l'[Objet](#) ne possède pas de [Material](#) celui de son père est appliqué, s'il ne possède pas de père le dernier [Material](#) est utilisé

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



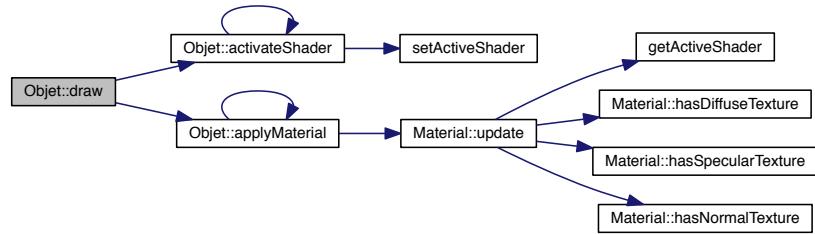
5.14.3.3 void `Objet::draw() [virtual]`

Affichage de l'[Objet](#).

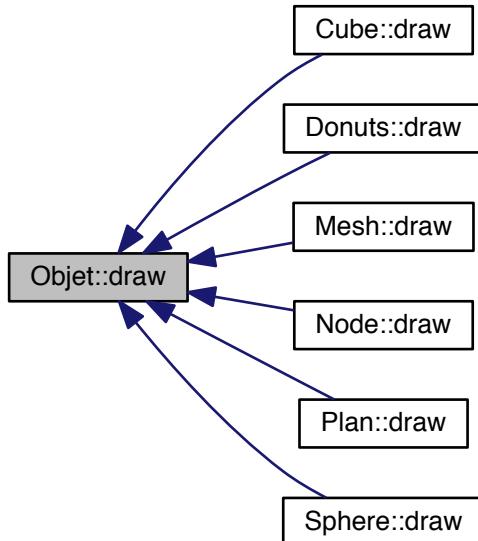
Active le shader de l'[Objet](#) et applique le material

Réimplémentée dans [Mesh](#), [Node](#), [Plan](#), [Cube](#), [Piece](#), [Donuts](#), et [Sphere](#).

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appels de cette fonction :



5.14.3.4 void Objet ::drawHitbox (bool b) [inline]

Affiche les Hitbox des objets capables d'en afficher.

Paramètres

<i>b</i>	valeur
----------	--------

5.14.3.5 bool Objet ::drawHitbox () const [inline]

Affiche la Hitbox de l'objet.

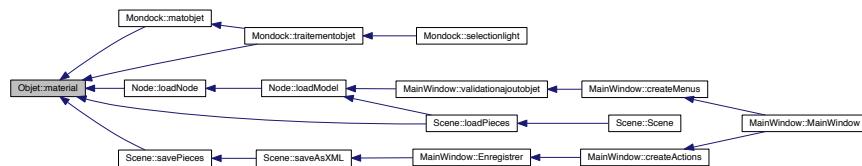
5.14.3.6 Material* Objet ::material () [inline]

Récupère le material de l'[Objet](#).

Renvoie

un pointeur valide ou NULL

Voici le graphe des appels de cette fonction :



5.14.3.7 void Objet ::material (Material * m) [inline]

Change le [Material](#) de l'[Objet](#).

Paramètres

<i>m</i>	pointeur sur le nouveau Material
----------	--

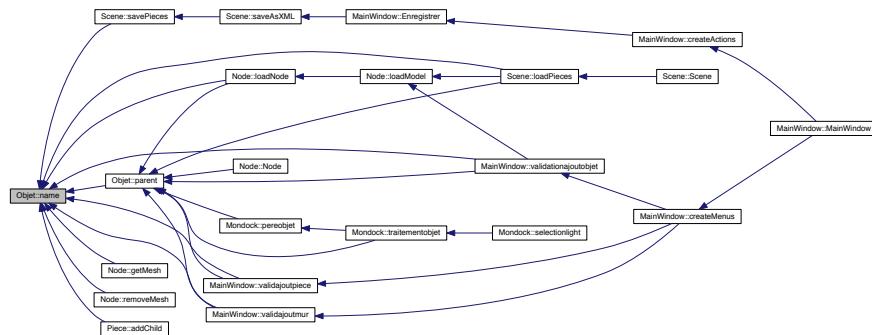
5.14.3.8 const QString& Objet ::name () const [inline]

Récupère le nom de l'[Objet](#).

Renvoie

nom de l'[Objet](#)

Voici le graphe des appels de cette fonction :



5.14.3.9 void Objet ::name (const QString & n) [inline]

Change le nom de l'[Objet](#).

Paramètres

<i>n</i>	nouveau nom de l' Objet
----------	---

5.14.3.10 void Objet::parent([Objet](#) * *o*)

Change l'[Objet](#) père.

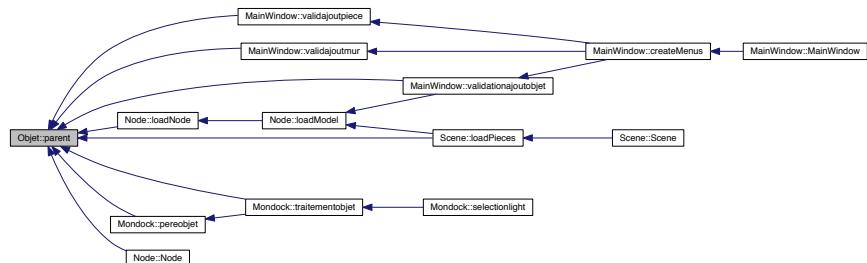
Paramètres

<i>o</i>	père
----------	------

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appels de cette fonction :

5.14.3.11 [Objet](#)* Objet::parent() [inline]

Récupère l'[Objet](#) père.

Renvoie

[Objet](#) père ou NULL

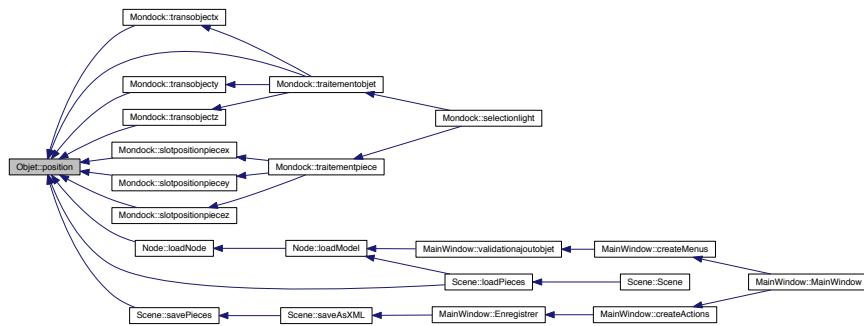
5.14.3.12 vec3& Objet::position() [inline]

Récupère la position de l'[Objet](#).

Renvoie

position de l'Objet

Voici le graphe des appels de cette fonction :



5.14.3.13 vec3 Objet::position() const [inline]

Récupère la position de l'Objet.

Renvoie

position de l'Objet

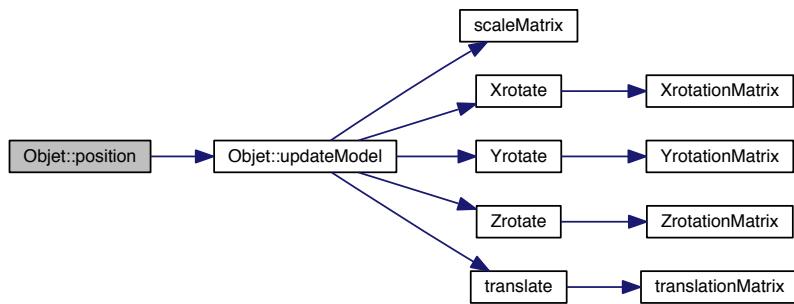
5.14.3.14 void Objet::position(vec3 p)

Change la position de l'Objet.

Paramètres

<i>p</i>	nouvelle position
----------	-------------------

Voici le graphe d'appel pour cette fonction :



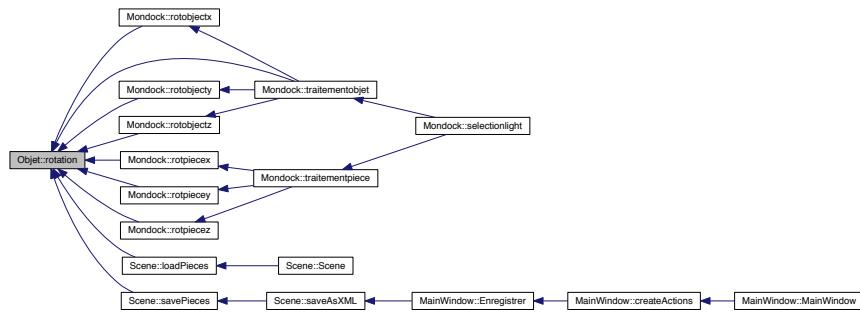
5.14.3.15 vec3& Objet::rotation() [inline]

Récupère la rotation de l'Objet.

Renvoie

rotation de l'Objet

Voici le graphe des appels de cette fonction :



5.14.3.16 vec3 Objet::rotation() const [inline]

Récupère la rotation de l'Objet.

Renvoie

rotation de l'Objet

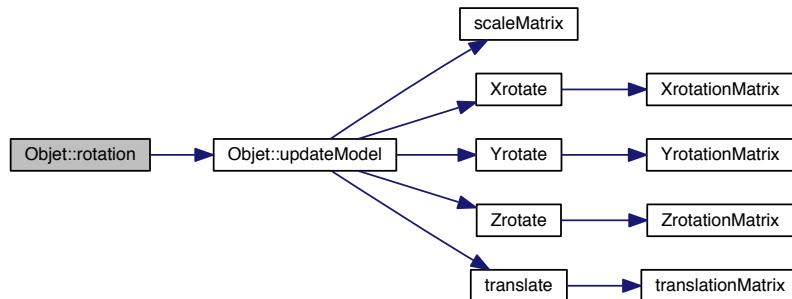
5.14.3.17 void Objet::rotation(vec3 r)

Change la rotation de l'Objet.

Paramètres

<i>r</i>	nouvelle rotation de l'Objet
----------	------------------------------

Voici le graphe d'appel pour cette fonction :



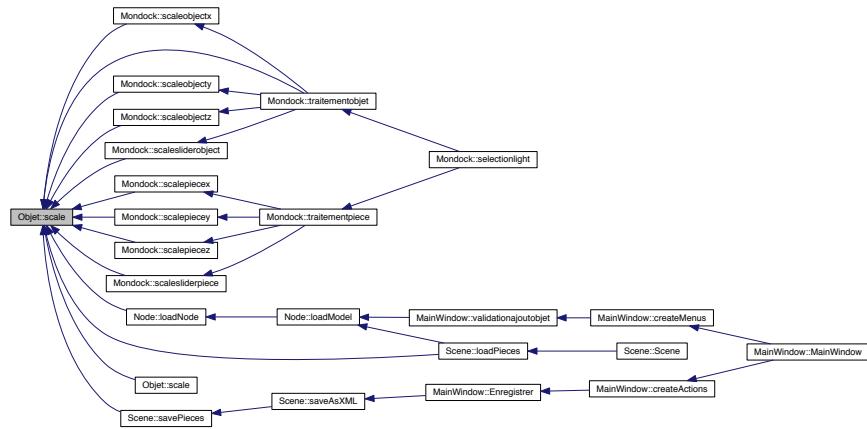
5.14.3.18 vec3& Objet::scale() [inline]

Récupère les facteurs de scale de l'objet.

Renvoie

scale de l'objet

Voici le graphe des appelants de cette fonction :



5.14.3.19 vec3 Objet::scale() const [inline]

Récupère les facteurs de scale de l'objet.

Renvoie

scale de l'objet

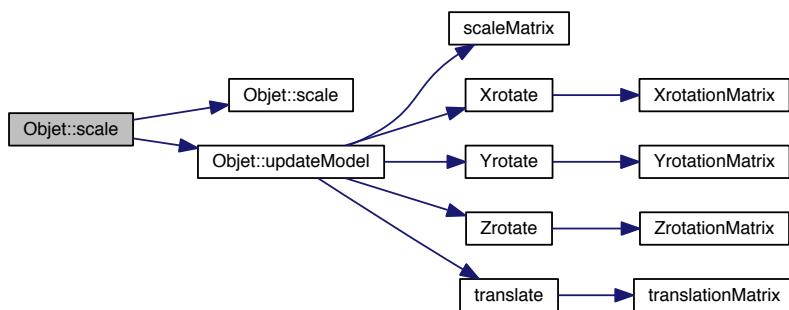
5.14.3.20 void Objet::scale(vec3 scale)

Modifie le scale de l'objet.

Paramètres

<i>scale</i>	nouveau vecteur de scale
--------------	--------------------------

Voici le graphe d'appel pour cette fonction :



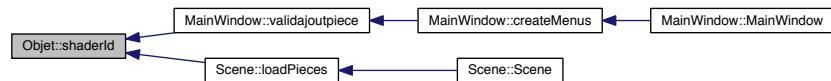
5.14.3.21 void Objet::shaderId(GLuint s) [inline]

Change l'id du shader.

Paramètres

s	nouvel id du shader de l' Objet
---	---

Voici le graphe des appels de cette fonction :

**5.14.3.22 GLuint Objet::shaderId() const [inline]**

Retourne l'id du shader qu'utilise l'[Objet](#).

Renvoie

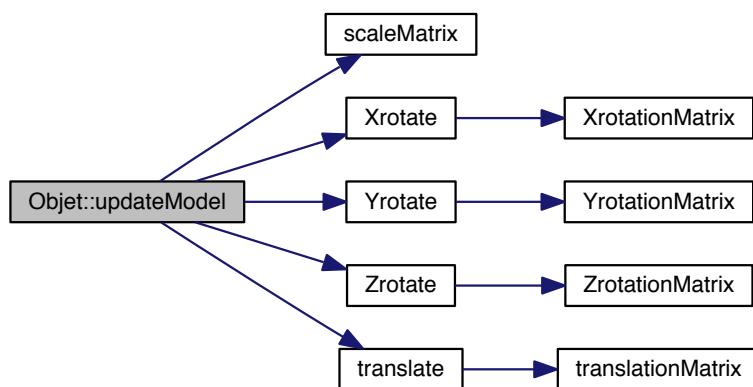
identifiant OpenGL du shader

5.14.3.23 void Objet::updateModel() [private]

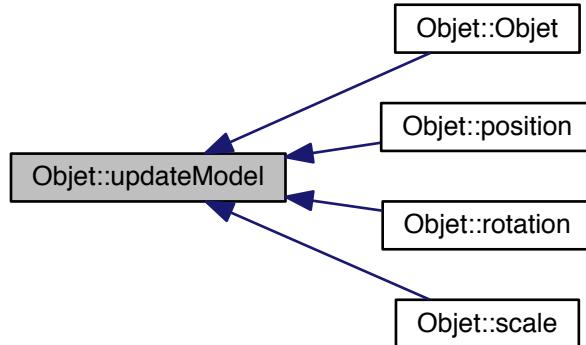
Met à jour la matrice de model.

Appelée lors d'une modification des vecteurs position ou rotation

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



5.14.4 Documentation des données membres

5.14.4.1 `bool Objet::_drawHitbox [protected]`

5.14.4.2 `Material* Objet::_mat [protected]`

5.14.4.3 `mat4 Objet::_model [protected]`

5.14.4.4 `QString Objet::_name [protected]`

5.14.4.5 `Objet* Objet::_parent [protected]`

5.14.4.6 `vec3 Objet::_position [protected]`

5.14.4.7 `vec3 Objet::_rotation [protected]`

5.14.4.8 `vec3 Objet::_scale [protected]`

5.14.4.9 `GLuint Objet::_shaderId [protected]`

La documentation de cette classe a été générée à partir des fichiers suivants :

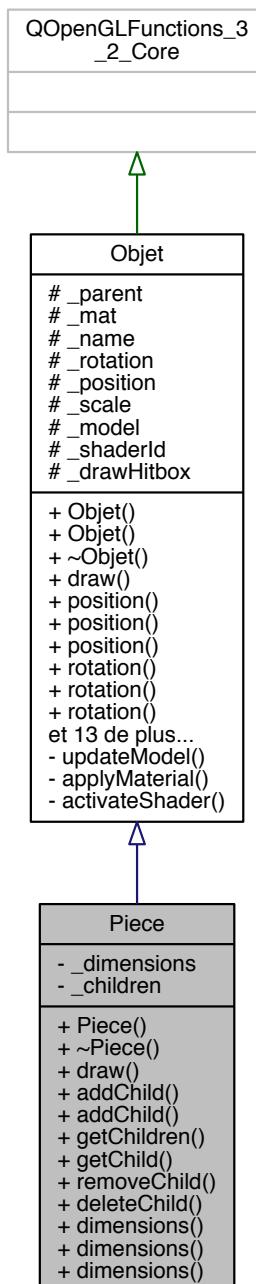
- [objets/objet.hpp](#)
- [objets/objet.cpp](#)

5.15 Référence de la classe Piece

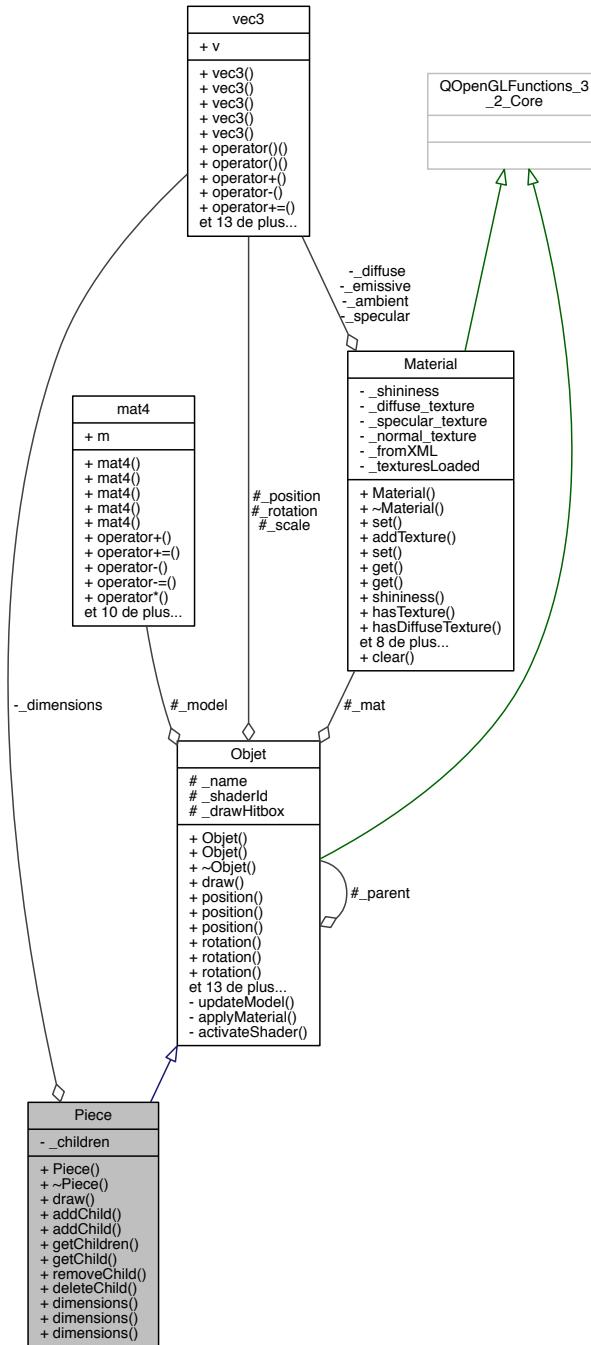
Pièce, contient un ensemble d'objet qu'elle contient.

```
#include <piece.hpp>
```

Graphe d'héritage de Piece :



Graphe de collaboration de Piece :



Fonctions membres publiques

- **Piece** (`vec3 dimension=vec3(1, 1, 1), vec3 rotation=vec3(), vec3 position=vec3(), Material *mat=NULL`)
Constructeur.
- **~Piece ()**
Destructeur.
- **void draw ()**
Affiche l'ensemble des fils.
- **void addChild (const QString &name, Objet *objet)**

- Ajoute un fils à la pièce.
- void **addChild** (*Objet* **o*)
 - Ajoute un fils à la pièce.
- QStringList **getChildren** () const
 - Récupère la liste des noms des objets fils.
- *Objet* * **getChild** (const QString &*name*)
 - Renvoie un pointeur sur un objet fils.
- void **removeChild** (const QString &*name*)
 - enlève un *Objet* fils de la pièce
- void **deleteChild** (const QString &*name*)
 - supprime un *Objet* fils de la *Pièce*
- const **vec3 & dimensions** () const
 - Retourne les dimensions de la pièce.
- **vec3 & dimensions** ()
 - Retourne les dimensions de la pièce.
- void **dimensions** (const *vec3* *v*)
 - Edite les dimensions de la pièce.

Attributs privés

- *vec3 _dimensions*
- QMap<QString, *Objet* *> *_children*

Membres hérités additionnels

5.15.1 Description détaillée

Pièce, contient un ensemble d'objet qu'elle contient.

5.15.2 Documentation des constructeurs et destructeur

5.15.2.1 Piece ::Piece (vec3 dimension = vec3 (1, 1, 1), vec3 rotation = vec3 (), vec3 position = vec3 (), Material * mat = NULL)

Constructeur.

Liste des objets fils de la pièces, contient au moins les murs

5.15.2.2 Piece ::~Piece ()

Destructeur.

5.15.3 Documentation des fonctions membres

5.15.3.1 void Piece ::addChild (const QString & name, Objet * objet)

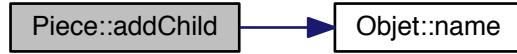
Ajoute un fils à la pièce.

Se charge de la destruction de l'objet

Paramètres

<i>name</i>	nom de l'objet
<i>objet</i>	objet à ajouter

Voici le graphe d'appel pour cette fonction :



5.15.3.2 void Piece ::addChild (Objet * o)

Ajoute un fils à la pièce.

Paramètres

<code>o</code>	objet à ajouter
----------------	-----------------

Voici le graphe d'appel pour cette fonction :



5.15.3.3 void Piece ::deleteChild (const QString & name)

supprime un [Objet](#) fils de la [Piece](#)

supprime l'[Objet](#) et libère la mémoire associé

Paramètres

<code>name</code>	nom de l' Objet
-------------------	---------------------------------

5.15.3.4 const vec3 & Piece ::dimensions () const

Retourne les dimensions de la pièce.

Renvoie

[vec3](#) contenant les dimensions de la pièce

5.15.3.5 vec3 & Piece ::dimensions ()

Retourne les dimensions de la pièce.

Renvoie

`vec3` contenant les dimensions de la pièce

5.15.3.6 void Piece ::dimensions (const vec3 v)

Edite les dimensions de la pièce.

Paramètres

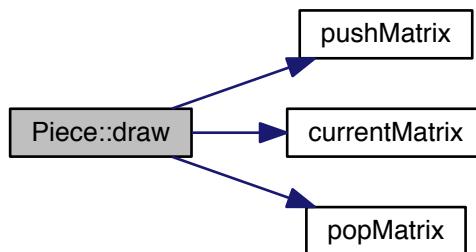
v	nouvelles dimensions de la pièce
---	----------------------------------

5.15.3.7 void Piece ::draw() [virtual]

Affiche l'ensemble des fils.

Réimplémentée à partir de [Objet](#).

Voici le graphe d'appel pour cette fonction :



5.15.3.8 Objet * Piece ::getChild (const QString & name)

Renvoie un pointeur sur un objet fils.

Récupère un pointeur sur un objet fils de la pièce

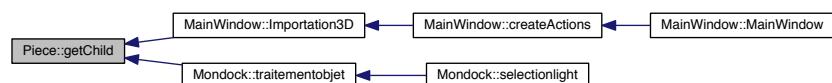
Paramètres

name	nom de l'objet
------	----------------

Renvoie

un pointeur valide si l'objet est un fils de la pièce, NULL sinon

Voici le graphe des appels de cette fonction :



5.15.3.9 QStringList Piece ::getChildren () const

Récupère la liste des noms des objets fils.

A faire A revoir !

Renvoie

liste des noms des objets

Voici le graphe des appelants de cette fonction :



5.15.3.10 void Piece ::removeChild (const QString & name)

enlève un **Objet** fils de la pièce

Paramètres

<i>name</i>	nom de l' Objet à supprimer
-------------	------------------------------------

5.15.4 Documentation des données membres

5.15.4.1 QMap<QString, Objet *> Piece ::_children [private]

5.15.4.2 vec3 Piece ::_dimensions [private]

Contient les dimensions de la pièces

La documentation de cette classe a été générée à partir des fichiers suivants :

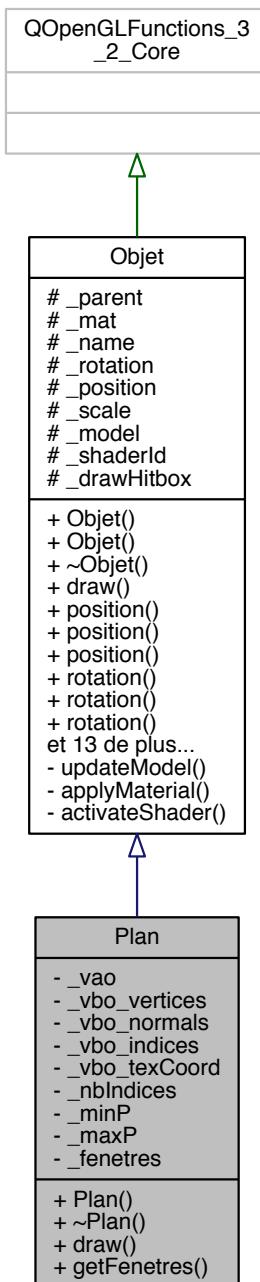
- [scene/piece.hpp](#)
- [scene/piece.cpp](#)

5.16 Référence de la classe Plan

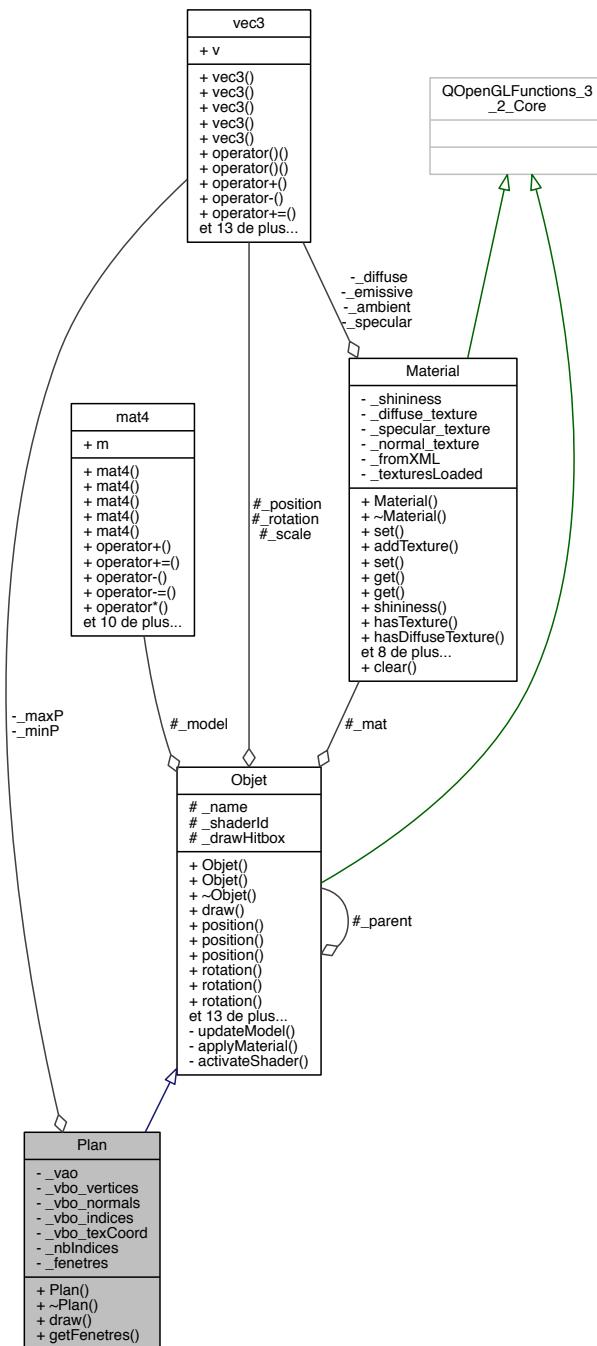
Plan utilisé par la **Piece**.

```
#include <plan.hpp>
```

Graphe d'héritage de Plan :



Graphe de collaboration de Plan :



Fonctions membres publiques

- `Plan` (int width=1, int height=1, int widthDivision=1, int heightDivision=1, const QList< QRectF > rects=QList< QRectF >(), Material *mat=NULL, vec3 rotation=vec3(), vec3 position=vec3())
 - Constructeur.*
- `~Plan ()`
 - Destructeur.*
- void `draw ()`
 - Affiche le plan.*

- const QList< QRectF > & [getFenetres \(\)](#) const
Renvoie la liste des rectangles définissant les fenêtres de la pièce.

Attributs privés

- GLuint [_vao](#)
- GLuint [_vbo_vertices](#)
- GLuint [_vbo_normals](#)
- GLuint [_vbo_indices](#)
- GLuint [_vbo_texCoord](#)
- unsigned int [_nblIndices](#)
- [vec3 _minP](#)
- [vec3 _maxP](#)
- QList< QRectF > [_fenetres](#)

Membres hérités additionnels

5.16.1 Description détaillée

[Plan](#) utilisé par la [Piece](#).

Gère un plan, avec des trous pour les fenêtres

5.16.2 Documentation des constructeurs et destructeur

5.16.2.1 Plan ::Plan (int *width* = 1, int *height* = 1, int *widthDivision* = 1, int *heightDivision* = 1, const QList< QRectF > *rects* = QList<QRectF>(), Material * *mat* = NULL, vec3 *rotation* = vec3(), vec3 *position* = vec3())

Constructeur.

Paramètres

<i>width</i>	largeur du plan
<i>height</i>	hauteur du plan
<i>widthDivision</i>	nombre de division selon width
<i>heightDivision</i>	nombre de division selon height
<i>rects</i>	QRectF rectangles définissant les fenêtres du plan
<i>mat</i>	Material à utiliser
<i>rotation</i>	rotation à appliquer au plan
<i>position</i>	position du plan dans le monde

5.16.2.2 Plan ::~Plan ()

Destructeur.

Détruit le plan et libère les vbo/vba

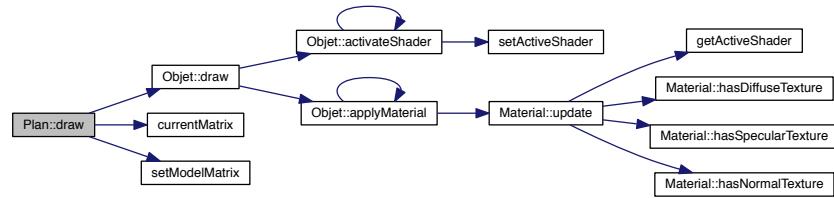
5.16.3 Documentation des fonctions membres

5.16.3.1 void Plan ::draw () [virtual]

Affiche le plan.

Réimplémentée à partir de [Objet](#).

Voici le graphe d'appel pour cette fonction :



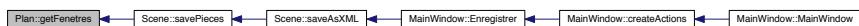
5.16.3.2 const QList<QRectF>& Plan ::getFenetres () const [inline]

Renvoie la liste des rectangles définissant les fenêtres de la pièce.

Renvoie

liste des fenêtres

Voici le graphe des appels de cette fonction :



5.16.4 Documentation des données membres

5.16.4.1 QList<QRectF> Plan ::_fenetres [private]

5.16.4.2 vec3 Plan ::_maxP [private]

5.16.4.3 vec3 Plan ::_minP [private]

5.16.4.4 unsigned int Plan ::_nbIndices [private]

5.16.4.5 GLuint Plan ::_vao [private]

5.16.4.6 GLuint Plan ::_vbo_indices [private]

5.16.4.7 GLuint Plan ::_vbo_normals [private]

5.16.4.8 GLuint Plan ::_vboTexCoord [private]

5.16.4.9 GLuint Plan ::_vbo_vertices [private]

La documentation de cette classe a été générée à partir des fichiers suivants :

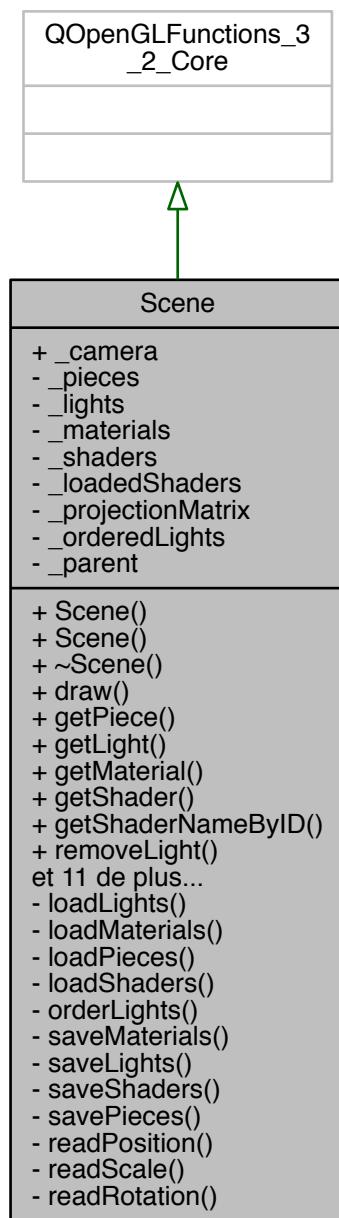
- objets/plan.hpp
- objets/plan.cpp

5.17 Référence de la classe Scene

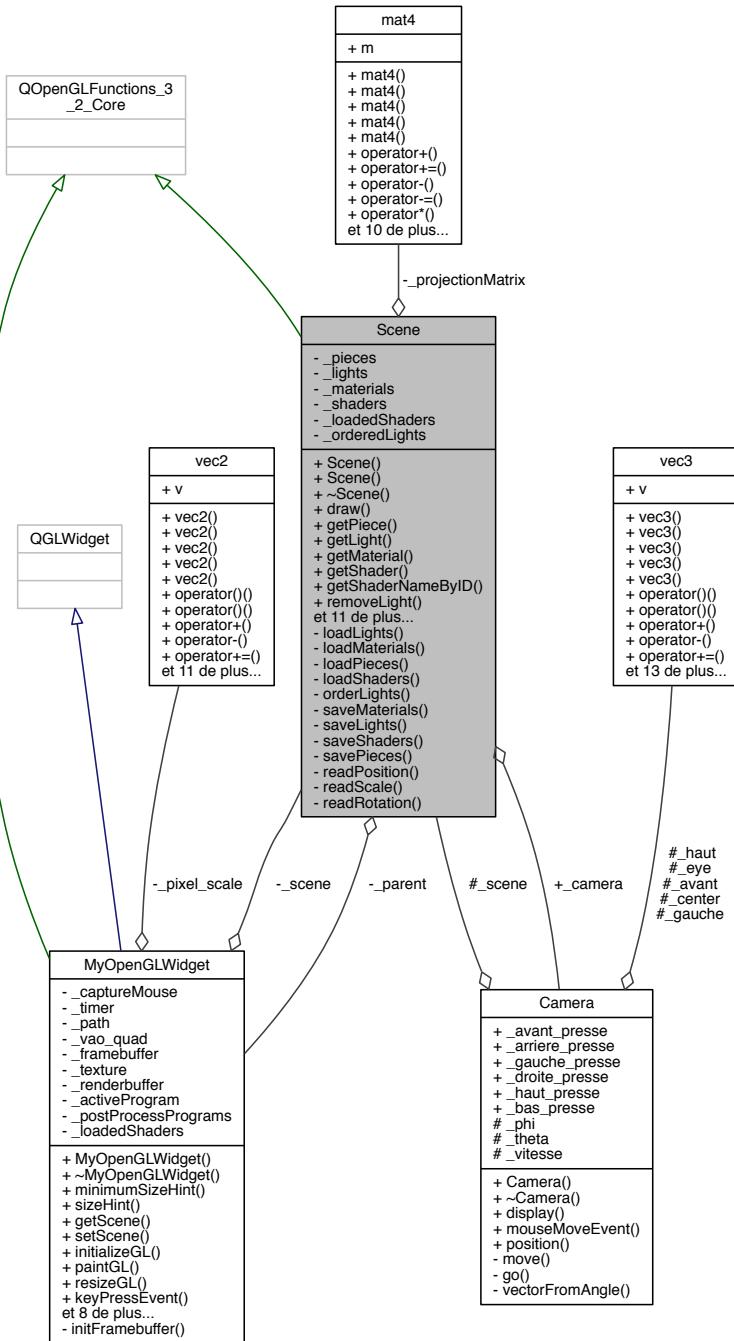
Classe principale.

```
#include <scene.hpp>
```

Graphe d'héritage de Scene :



Graphe de collaboration de Scene :



Fonctions membres publiques

- **Scene (MyOpenGLWidget *parent)**
Constructeur vide.
- **Scene (MyOpenGLWidget *parent, const QString &fileName)**
Charge une scène à partir d'un fichier xml la décrivant.
- **~Scene ()**
Destructeur de la scène.
- **void draw ()**

- **Affiche la scène.**
- **Piece * getPiece (const QString &name) const**
Récupère un pointeur sur une pièce.
- **Light * getLight (const QString &name) const**
Récupère un pointeur sur une lumière.
- **Material * getMaterial (const QString &name) const**
Récupère un pointeur sur un material.
- **GLuint getShader (const QString &name) const**
Récupère l'id d'un shader.
- **QString getShaderNameByID (const GLuint id) const**
Recherche le nom d'un shader à partir de son ID OpenGL.
- **void removeLight (const QString &name)**
Supprime la Light voulu
- **void removePiece (const QString &name)**
Supprime la Piece demandée
- **QStringList getPiecesName () const**
Retourne la liste des noms des pièces dans la scène.
- **QStringList getLightsNames () const**
Retourne la liste des noms des lumières dans la scène.
- **QStringList getMaterialsNames () const**
Retourne la liste des noms des matériaux dans la scène.
- **QStringList getShadersNames () const**
Retourne la liste des noms des shaders dans la scène.
- **QString getMaterialName (Material *m) const**
Retourne le nom du Material passé en paramètre.
- **void addPiece (const QString &name, Piece *o)**
Ajoute une pièce à la scène.
- **void addLight (const QString &name, Light *)**
Ajoute une lumière à la scène.
- **void addMaterial (const QString &name, Material *m)**
Ajoute un material à la scène.
- **void addShader (const QString &name, QOpenGLShaderProgram *s)**
Ajoute un shader à la scène.
- **bool saveAsXML (const QString &fileName)**
Sauvegarde de la scène au format xml.

Attributs publics

- **Camera * _camera**

Fonctions membres privées

- **void loadLights (const QDomElement &dom)**
Charge les lumières.
- **void loadMaterials (const QDomElement &dom)**
Charge les materials.
- **void loadPieces (const QDomElement &dom)**
Charge les pièces.
- **void loadShaders (const QDomElement &dom)**
Charge les shaders.
- **void orderLights ()**
Ordonne les Light en fonction de la distance à la caméra.
- **void saveMaterials (QDomElement &root, QDomDocument &doc) const**
- **void saveLights (QDomElement &root, QDomDocument &doc) const**
- **void saveShaders (QDomElement &root, QDomDocument &doc) const**
- **void savePieces (QDomElement &root, QDomDocument &doc) const**

Fonctions membres privées statiques

- **static vec3 readPosition (const QDomElement &e)**
Retourne un vec3 correspondant au vecteur position lu.
- **static vec3 readScale (const QDomElement &e)**
Retourne un vec3 correspondant au scale lu.
- **static vec3 readRotation (const QDomElement &e)**
Retourne un vec3 correspondant à la rotation.

Attributs privés

```
— QMap< QString, Piece * > _pieces
— QMap< QString, Light * > _lights
— QMap< QString, Material * > _materials
— QMap< QString,
    QOpenGLShaderProgram * > _shaders
— QMap< QString, QOpenGLShader * > _loadedShaders
— mat4 _projectionMatrix
— QMap< float, Light * > _orderedLights
— MyOpenGLWidget * _parent
```

5.17.1 Description détaillée

Classe principale.

Se charge de lire le xml et de générer la scène. Nécessite un context OpenGL 3.2 au minimum

5.17.2 Documentation des constructeurs et destructeur

5.17.2.1 Scene ::Scene (MyOpenGLWidget * *parent*)

Constructeur vide.

Construit une scène vide

5.17.2.2 Scene ::Scene (MyOpenGLWidget * *parent*, const QString & *fileName*)

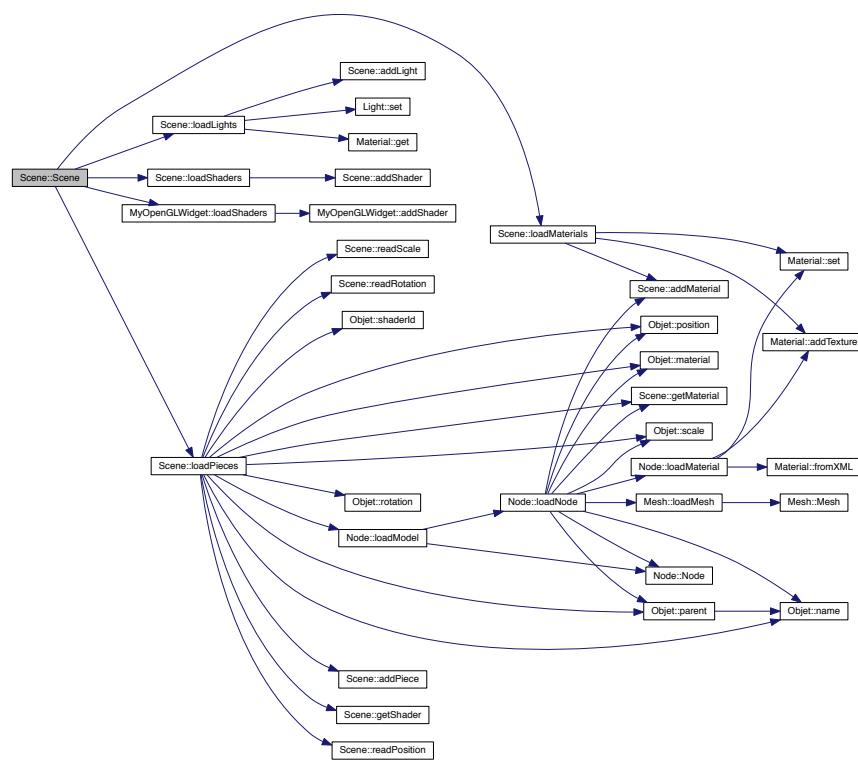
Charge une scène à partir d'un fichier xml la décrivant.

La scène est chargée à partir d'un fichier xml la décrivant, si le fichier n'est pas trouvé l'exécution s'arrête

Paramètres

<i>fileName</i>	Chemin vers le fichier
-----------------	------------------------

Voici le graphe d'appel pour cette fonction :



5.17.2.3 `Scene::~Scene()`

Destructeur de la scène.

Destructeur de la scène

Voici le graphe d'appel pour cette fonction :



5.17.3 Documentation des fonctions membres

5.17.3.1 `void Scene::addLight(const QString & name, Light * l)`

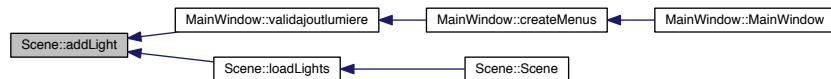
Ajoute une lumière à la scène.

Ajoute une lumière à la scène, le nom doit être unique, sinon elle ne sera pas ajoutée

Paramètres

<i>name</i>	nom de la lumière à ajouter
<i>l</i>	pointeur sur la lumière

Voici le graphe des appelants de cette fonction :



5.17.3.2 void Scene ::addMaterial (const QString & name, Material * m)

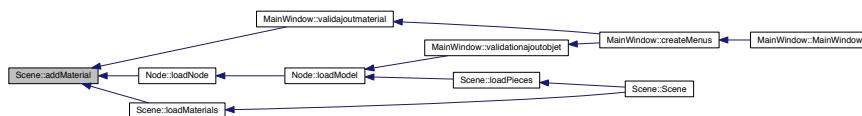
Ajoute un material à la scène.

Ajoute un material à la scène, le nom doit être unique, autrement il ne sera pas ajouté

Paramètres

<i>name</i>	nom du material à ajouter
<i>m</i>	pointeur sur le material

Voici le graphe des appelants de cette fonction :



5.17.3.3 void Scene ::addPiece (const QString & name, Piece * o)

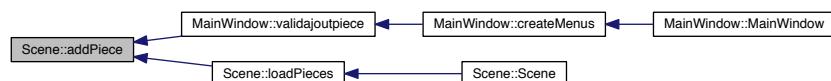
Ajoute une pièce à la scène.

Ajoute une pièce à la scène, le nom ne doit pas être déjà présent dans la scène, si l'est la pièce n'est pas ajoutée

Paramètres

<i>name</i>	nom de l'objet à ajouter
<i>o</i>	pointeur sur l'objet à ajouter

Voici le graphe des appelants de cette fonction :



5.17.3.4 void Scene ::addShader (const QString & name, QOpenGLShaderProgram * s)

Ajoute un shader à la scène.

Ajoute un shader à la scène, le nom doit être unique, autrement il ne sera pas ajouté

Paramètres

<i>name</i>	nom du shader à ajouter
<i>s</i>	id du shader

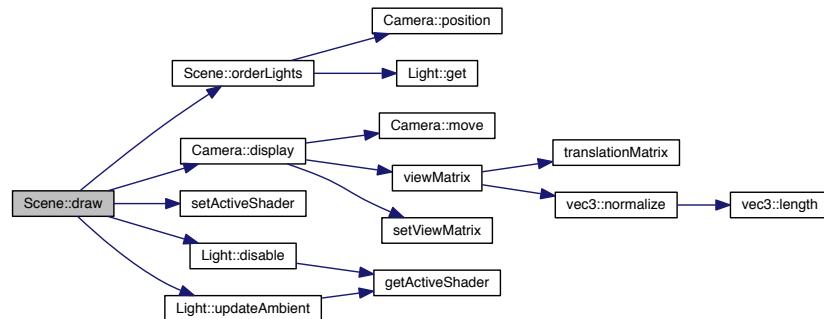
Voici le graphe des appelants de cette fonction :



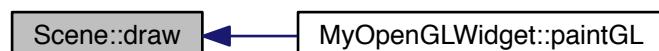
5.17.3.5 void Scene ::draw ()

Affiche la scène.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



5.17.3.6 `Light * Scene::getLight(const QString & name) const`

Récupère un pointeur sur une lumière.

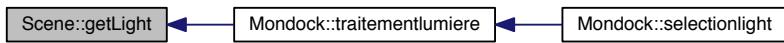
Paramètres

<i>name</i>	nom de la lumière
-------------	-------------------

Renvoie

Retourne un pointeur valide si le nom est présent, NULL autrement

Voici le graphe des appelants de cette fonction :

**5.17.3.7 QStringList Scene ::getLightsNames () const**

Retourne la liste des noms des lumières dans la scène.

Renvoie

QStringList contenant les noms des lumières de la scène

Voici le graphe des appelants de cette fonction :

**5.17.3.8 Material * Scene ::getMaterial (const QString & name) const**

Récupère un pointeur sur un material.

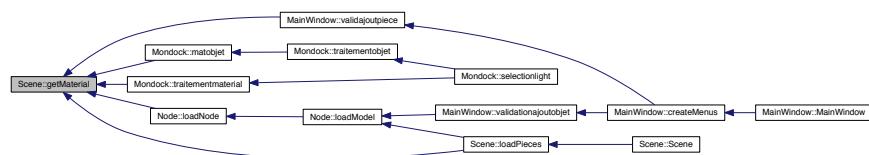
Paramètres

<i>name</i>	nom du material
-------------	-----------------

Renvoie

Retourne un pointeur valide si le nom est présent, NULL autrement

Voici le graphe des appelants de cette fonction :



5.17.3.9 **QString Scene::getMaterialName (Material * m) const**

Retourne le nom du [Material](#) passé en paramètre.

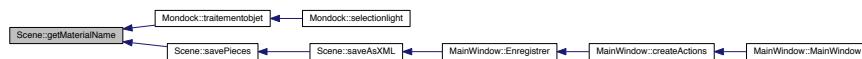
Paramètres

<i>m</i>	Material dont on veut le nom
----------	------------------------------

Renvoie

nom du Material

Voici le graphe des appelants de cette fonction :

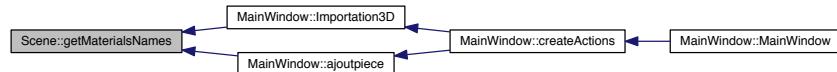
**5.17.3.10 QStringList Scene ::getMaterialsNames () const**

Retourne la liste des noms des matériaux dans la scène.

Renvoie

QStringList contenant les noms des matériaux de la scène

Voici le graphe des appelants de cette fonction :

**5.17.3.11 Piece * Scene ::getPiece (const QString & name) const**

Récupère un pointeur sur une pièce.

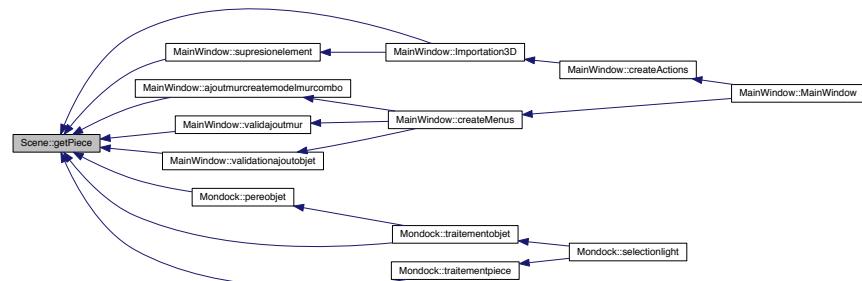
Paramètres

<i>name</i>	nom de la pièce
-------------	-----------------

Renvoie

Retourne le pointeur si le nom est trouvé, NULL autrement

Voici le graphe des appelants de cette fonction :



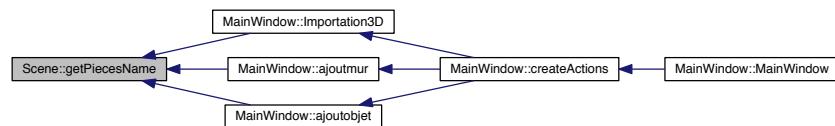
5.17.3.12 QStringList Scene ::getPiecesName () const

Retourne la liste des noms des pièces dans la scène.

Renvoie

QStringList contenant les noms des pièces de la scène

Voici le graphe des appelants de cette fonction :



5.17.3.13 GLuint Scene ::getShader (const QString & name) const

Récupère l'id d'un shader.

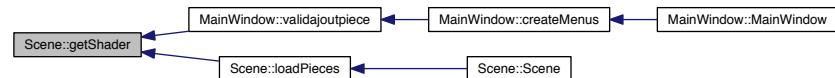
Paramètres

<i>name</i>	nom du shader
-------------	---------------

Renvoie

Retourne l'id d'un shader valide si le nom est présent, 0 autrement

Voici le graphe des appelants de cette fonction :



5.17.3.14 QString Scene ::getShaderNameByID (const GLuint id) const

Recherche le nom d'un shader à partir de son ID OpenGL.

Paramètres

<i>id</i>	identifiant OpenGL du shader
-----------	------------------------------

Renvoie

nom du shader s'il existe

Voici le graphe des appelants de cette fonction :



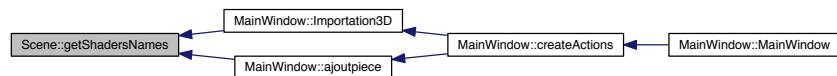
5.17.3.15 QStringList Scene ::getShadersNames () const

Retourne la liste des noms des shaders dans la scène.

Renvoie

QStringList contenant les noms des shaders de la scène

Voici le graphe des appelants de cette fonction :



5.17.3.16 void Scene ::loadLights (const QDomElement & dom) [private]

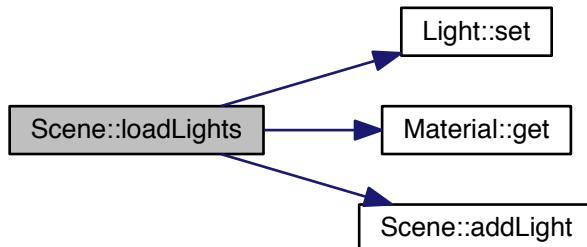
Charge les lumières.

Charge les lumières à partir du XML

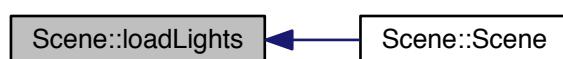
Paramètres

<i>dom</i>	noeud XML correspondant à <lumieres>
------------	--------------------------------------

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



5.17.3.17 void Scene ::loadMaterials (const QDomElement & dom) [private]

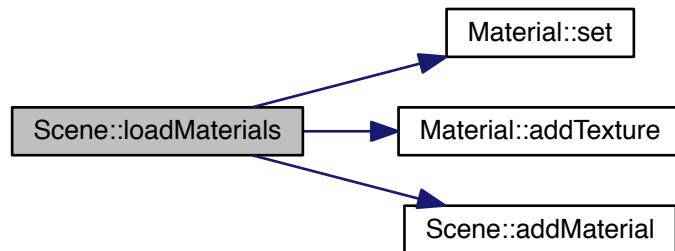
Charge les materials.

Charge les materials à partir du XML

Paramètres

<i>dom</i>	noeud XML correspondant à <materiaux>
------------	---------------------------------------

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appels de cette fonction :

**5.17.3.18 void Scene ::loadPieces (const QDomElement & dom) [private]**

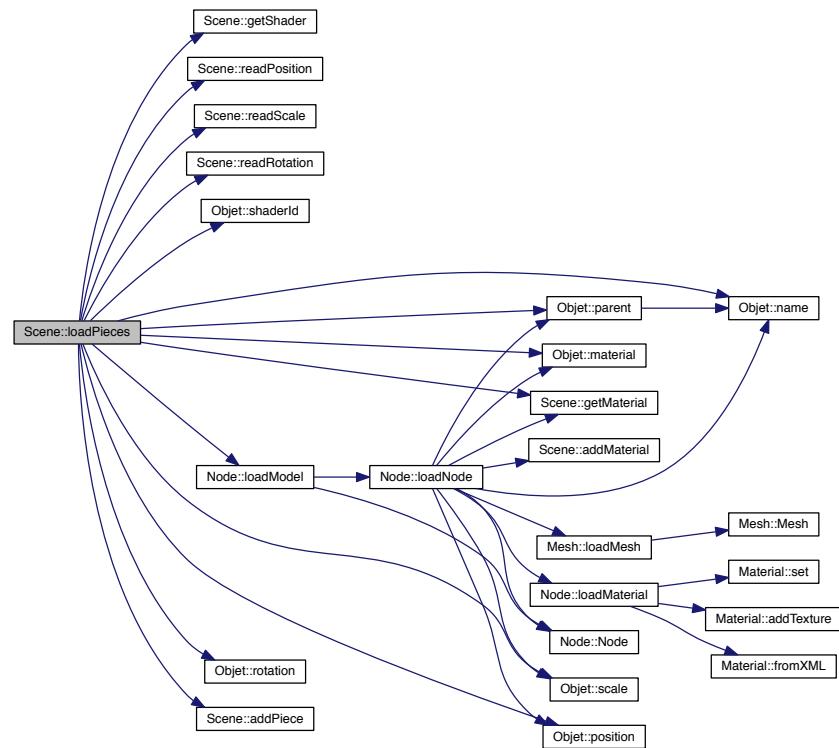
Charge les pièces.

Charge les pièces à partir du XML

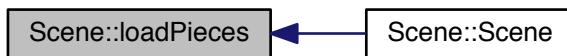
Paramètres

<i>dom</i>	noeud XML correspondant à <pieces>
------------	------------------------------------

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appels de cette fonction :



5.17.3.19 void Scene ::loadShaders (const QDomElement & dom) [private]

Charge les shaders.

Charge les shaders à partir du XML

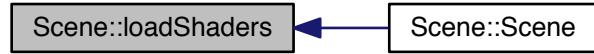
Paramètres

<code>dom</code>	noeud XML correspondant à <shaders>
------------------	-------------------------------------

Voici le graphe d'appel pour cette fonction :



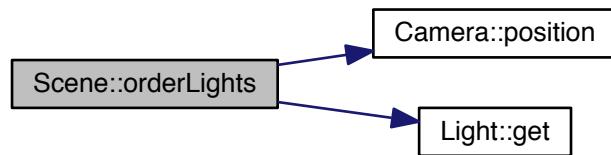
Voici le graphe des appelants de cette fonction :



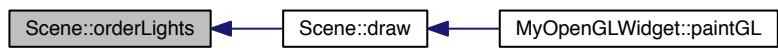
5.17.3.20 void Scene ::orderLights() [private]

Ordonne les [Light](#) en fonction de la distance à la caméra.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



5.17.3.21 `vec3 Scene::readPosition (const QDomElement & e) [static], [private]`

Retourne un `vec3` correspondant au vecteur position lu.

Paramètres

e	élément à lire
---	----------------

Renvoie

position

Voici le graphe des appelants de cette fonction :

**5.17.3.22 vec3 Scene::readRotation (const QDomElement & e) [static], [private]**

Retourne un **vec3** correspondant à la rotation.

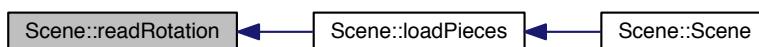
Paramètres

e	élément à lire
---	----------------

Renvoie

vecteur de rotation

Voici le graphe des appelants de cette fonction :

**5.17.3.23 vec3 Scene::readScale (const QDomElement & e) [static], [private]**

Retourne un **vec3** correspondant au scale lu.

Paramètres

e	élément à lire
---	----------------

Renvoie

facteur de scale

Voici le graphe des appelants de cette fonction :



5.17.3.24 void Scene ::removeLight (const QString & name)

supprime la [Light](#) voulu

Paramètres

<i>name</i>	nom de la Light à supprimer
-------------	---

Voici le graphe des appelants de cette fonction :



5.17.3.25 void Scene ::removePiece (const QString & name)

supprime la [Piece](#) demandée

Paramètres

<i>name</i>	nom de la Piece à supprimer
-------------	---

Voici le graphe des appelants de cette fonction :



5.17.3.26 bool Scene ::saveAsXML (const QString & fileName)

Sauvegarde de la scène au format xml.

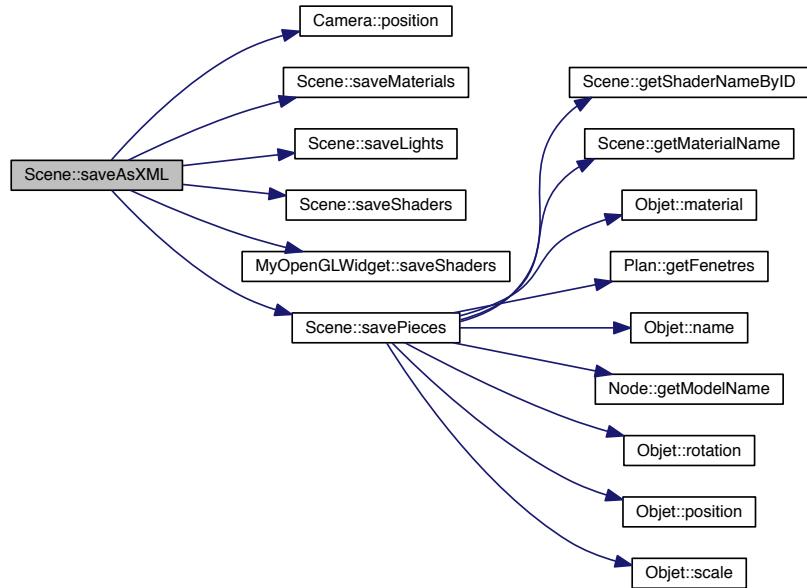
Paramètres

<i>fileName</i>	nom du fichier dans lequel sauvegarde
-----------------	---------------------------------------

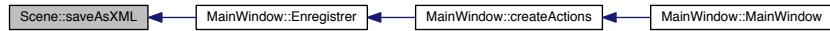
Renvoie

indique si le ficheir a été sauvegardé ou pas

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



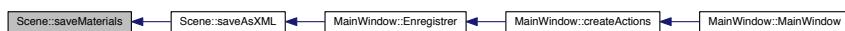
5.17.3.27 void Scene ::saveLights (QDomElement & root, QDomDocument & doc) const [private]

Voici le graphe des appelants de cette fonction :



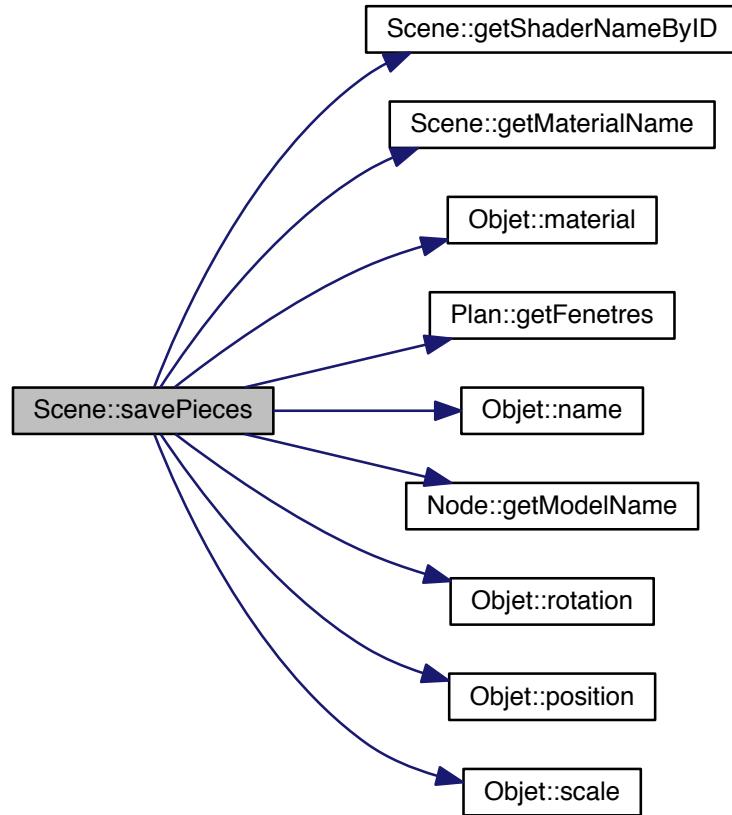
5.17.3.28 void Scene ::saveMaterials (QDomElement & root, QDomDocument & doc) const [private]

Voici le graphe des appelants de cette fonction :



5.17.3.29 void Scene ::savePieces (QDomElement & root, QDomDocument & doc) const [private]

Voici le graphe d'appel pour cette fonction :

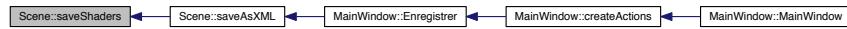


Voici le graphe des appels de cette fonction :



5.17.3.30 void Scene ::saveShaders (QDomElement & root, QDomDocument & doc) const [private]

Voici le graphe des appels de cette fonction :



5.17.4 Documentation des données membres

5.17.4.1 Camera* Scene::_camera

Caméra de la scène

5.17.4.2 QMap<QString, Light *> Scene::_lights [private]

Map des lumières constituant la scène, identifiées par leur nom, doit être unique

5.17.4.3 QMap<QString, QOpenGLShader *> Scene::_loadedShaders [private]

Fichiers de shaders déjà chargés, utilisé pour l'export

5.17.4.4 QMap<QString, Material *> Scene::_materials [private]

Map des matériaux constituant la scène, identifiés par leur nom, doit être unique

5.17.4.5 QMap<float, Light *> Scene::_orderedLights [private]

Light ordonnées par leurs distance à la caméra

5.17.4.6 MyOpenGLWidget* Scene::_parent [private]

5.17.4.7 QMap<QString, Piece *> Scene::_pieces [private]

Map des pièces constituant la scène, identifiées par leur nom, doit être unique

5.17.4.8 mat4 Scene::_projectionMatrix [private]

Matrice de projection, recalculée à chaque redimensionnement du widget

5.17.4.9 QMap<QString, QOpenGLShaderProgram *> Scene::_shaders [private]

Map des shader constituant la scène, identifiés par leur nom, doit être unique

La documentation de cette classe a été générée à partir des fichiers suivants :

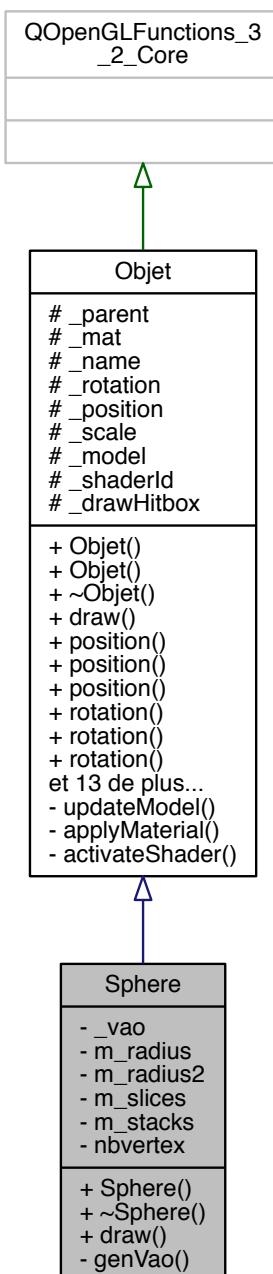
- [scene/scene.hpp](#)
- [scene/scene.cpp](#)

5.18 Référence de la classe Sphere

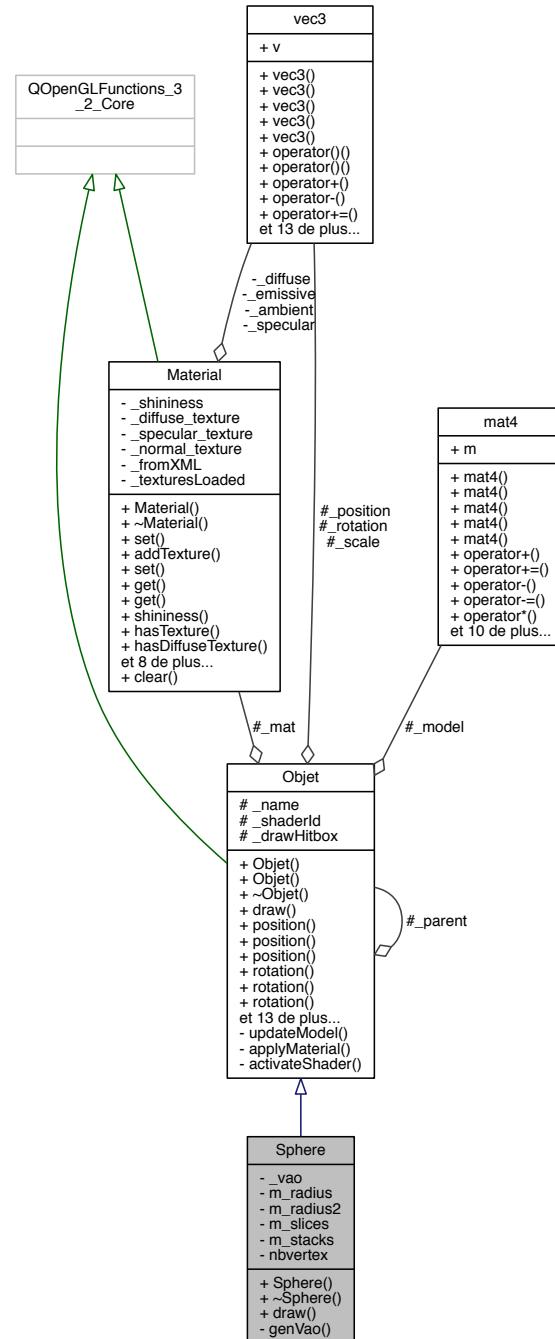
Primitive tore.

```
#include <sphere.hpp>
```

Graphe d'héritage de Sphere :



Graphe de collaboration de Sphere :



Fonctions membres publiques

- **Sphere** (GLdouble `m_radius`=1, GLdouble `m_radius2`=1, GLint `m_slices`=1, GLint `m_stacks`=1, Material *`mat`=NULL, vec3 `rotation`=vec3(), vec3 `position`=vec3())
- **~Sphere ()**
- void **draw ()**

Affichage de l'*Objet*.

Fonctions membres privées

— void [genVao \(\)](#)

Attributs privés

- GLuint [_vao](#)
- GLdouble [m_radius](#)
- GLdouble [m_radius2](#)
- GLint [m_slices](#)
- GLint [m_stacks](#)
- GLsizei [nbvertex](#)

Membres hérités additionnels

5.18.1 Description détaillée

Primitive tore.

Avertissement

Peut être plus utilisable

Voir également

[Cube](#)

5.18.2 Documentation des constructeurs et destructeur

5.18.2.1 `Sphere : :Sphere (GLdouble m_radius = 1, GLdouble m_radius2 = 1, GLint m_slices = 1, GLint m_stacks = 1, Material * mat = NULL, vec3 rotation = vec3 (), vec3 position = vec3 ())`

Voici le graphe d'appel pour cette fonction :



5.18.2.2 `Sphere : :~Sphere ()`

5.18.3 Documentation des fonctions membres

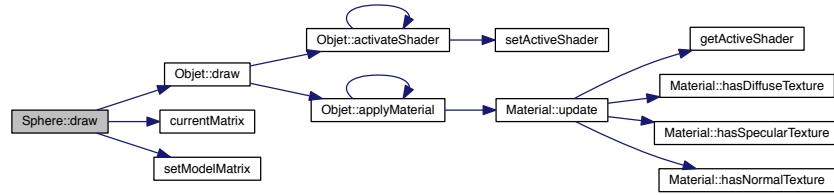
5.18.3.1 void `Sphere : :draw () [virtual]`

Affichage de l'[Objet](#).

Active le shader de l'[Objet](#) et applique le material

Réimplémentée à partir de [Objet](#).

Voici le graphe d'appel pour cette fonction :



5.18.3.2 void Sphere ::genVao () [private]

Voici le graphe des appels de cette fonction :



5.18.4 Documentation des données membres

5.18.4.1 GLuint Sphere ::_vao [private]

5.18.4.2 GLdouble Sphere ::m_radius [private]

5.18.4.3 GLdouble Sphere ::m_radius2 [private]

5.18.4.4 GLint Sphere ::m_slices [private]

5.18.4.5 GLint Sphere ::m_stacks [private]

5.18.4.6 GLsizei Sphere ::nbvertex [private]

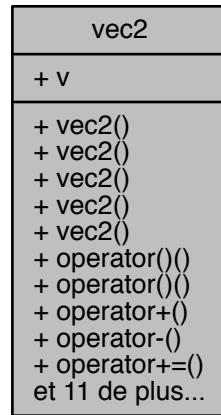
La documentation de cette classe a été générée à partir des fichiers suivants :

- objets/sphere.hpp
- objets/sphere.cpp

5.19 Référence de la structure vec2

```
#include <helper.hpp>
```

Graphe de collaboration de vec2 :



Fonctions membres publiques

- `vec2` (float x, float y)
- `vec2` (const `vec2` &vv)
- `vec2` (const `vec3` &vv)
- `vec2` (const `vec4` &vv)
- `vec2` ()
- `vec2` & `operator()` (const `vec2` &)
- `vec2` & `operator()` (float x, float y)
- `vec2` `operator+` (const `vec2` &) const
- `vec2` `operator-` (const `vec2` &) const
- `vec2` & `operator+=` (const `vec2` &)
- `vec2` & `operator-=` (const `vec2` &)
- `vec2` `operator-` (void) const
- `vec2` `operator*` (const float) const
- `vec2` `operator/` (const float) const
- `vec2` & `operator**=` (const float)
- `vec2` & `operator/=` (const float)
- `vec2` & `operator*=` (const `vec2` &)
- float & `operator[]` (int)
- float `operator[]` (int) const
- float `length` () const
- void `normalize` ()

Attributs publics

- float `v` [2]

5.19.1 Documentation des constructeurs et destructeur

5.19.1.1 `vec2::vec2(float x, float y)`

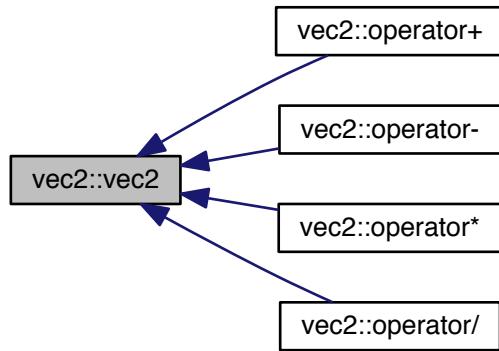
5.19.1.2 `vec2::vec2(const vec2 & vv)`

5.19.1.3 `vec2::vec2(const vec3 & vv)`

5.19.1.4 `vec2::vec2(const vec4 & vv)`

5.19.1.5 `vec2 ::vec2()`

Voici le graphe des appelants de cette fonction :



5.19.2 Documentation des fonctions membres

5.19.2.1 `float vec2 ::length() const`

Voici le graphe des appelants de cette fonction :

5.19.2.2 `void vec2 ::normalize()`

Voici le graphe d'appel pour cette fonction :



5.19.2.3 `vec2 & vec2 ::operator() (const vec2 & vv)`

5.19.2.4 `vec2 & vec2 ::operator() (float x, float y)`

5.19.2.5 `vec2 vec2 ::operator* (const float t) const`

Voici le graphe d'appel pour cette fonction :



5.19.2.6 `vec2 & vec2 ::operator*=(const float t)`

5.19.2.7 `vec2 vec2 ::operator+ (const vec2 & vv) const`

Voici le graphe d'appel pour cette fonction :



5.19.2.8 `vec2 & vec2 ::operator+=(const vec2 & vv)`

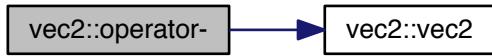
5.19.2.9 `vec2 vec2 ::operator- (const vec2 & vv) const`

Voici le graphe d'appel pour cette fonction :



5.19.2.10 `vec2 vec2 ::operator- (void) const`

Voici le graphe d'appel pour cette fonction :



5.19.2.11 `vec2 & vec2 ::operator-= (const vec2 & vv)`

5.19.2.12 `vec2 vec2 ::operator/ (const float t) const`

Voici le graphe d'appel pour cette fonction :



5.19.2.13 `vec2 & vec2 ::operator/= (const float t)`

5.19.2.14 `vec2 & vec2 ::operator= (const vec2 & vv)`

5.19.2.15 `float & vec2 ::operator[] (int i)`

5.19.2.16 `float vec2 ::operator[] (int i) const`

5.19.3 Documentation des données membres

5.19.3.1 `float vec2 ::v[2]`

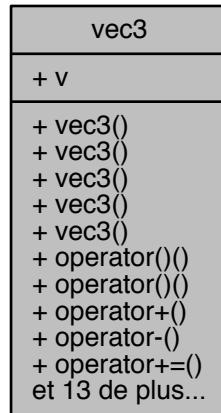
La documentation de cette structure a été générée à partir des fichiers suivants :

- [helper.hpp](#)
- [helper.cpp](#)

5.20 Référence de la structure vec3

```
#include <helper.hpp>
```

Graphe de collaboration de vec3 :



Fonctions membres publiques

- `vec3` (float x, float y, float z)
- `vec3` (const `vec2` &vv, float z)
- `vec3` (const `vec3` &vv)
- `vec3` (const `vec4` &vv)
- `vec3` ()
- `vec3` & `operator()` (const `vec3` &)
- `vec3` & `operator()` (float x, float y, float z)
- `vec3` `operator+` (const `vec3` &) const
- `vec3` `operator-` (const `vec3` &) const
- `vec3` & `operator+=` (const `vec3` &)
- `vec3` & `operator-=` (const `vec3` &)
- `vec3` `operator-` (void) const
- `vec3` `operator*` (const float) const
- `vec3` `operator/` (const float) const
- `vec3` & `operator*=` (const float)
- `vec3` & `operator/=` (const float)
- `vec3` `operator^` (const `vec3` &) const
- float `operator*` (const `vec3` &) const
- `vec3` & `operator=` (const `vec3` &)
- float & `operator[]` (int)
- float `operator[]` (int) const
- float `length` () const
- void `normalize` ()

Attributs publics

- float `v` [3]

5.20.1 Documentation des constructeurs et destructeur

5.20.1.1 `vec3` ::`vec3` (float x, float y, float z)

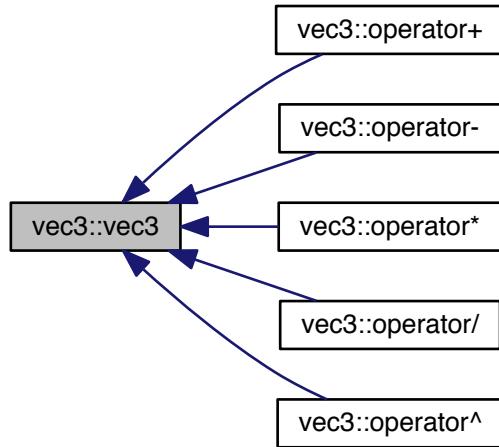
5.20.1.2 `vec3` ::`vec3` (const `vec2` & vv, float z)

5.20.1.3 `vec3` ::`vec3` (const `vec3` & vv)

5.20.1.4 `vec3::vec3(const vec4 & vv)`

5.20.1.5 `vec3::vec3()`

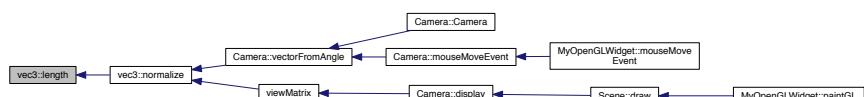
Voici le graphe des appelants de cette fonction :



5.20.2 Documentation des fonctions membres

5.20.2.1 `float vec3::length() const`

Voici le graphe des appelants de cette fonction :

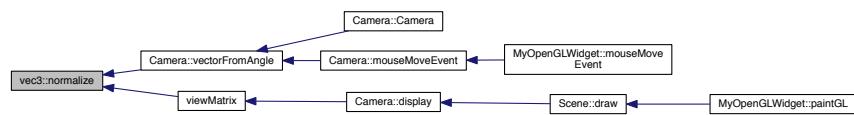


5.20.2.2 `void vec3::normalize()`

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appels de cette fonction :

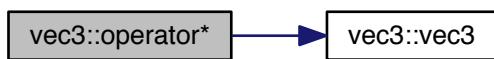


5.20.2.3 `vec3 & vec3 ::operator() (const vec3 & vv)`

5.20.2.4 `vec3 & vec3 ::operator() (float x, float y, float z)`

5.20.2.5 `vec3 vec3 ::operator* (const float t) const`

Voici le graphe d'appel pour cette fonction :



5.20.2.6 `float vec3 ::operator* (const vec3 & vv) const`

5.20.2.7 `vec3 & vec3 ::operator*=(const float t)`

5.20.2.8 `vec3 vec3 ::operator+ (const vec3 & vv) const`

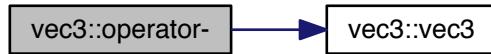
Voici le graphe d'appel pour cette fonction :



5.20.2.9 `vec3 & vec3 ::operator+=(const vec3 & vv)`

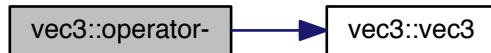
5.20.2.10 `vec3 vec3 ::operator- (const vec3 & vv) const`

Voici le graphe d'appel pour cette fonction :



5.20.2.11 `vec3 vec3 ::operator- (void) const`

Voici le graphe d'appel pour cette fonction :



5.20.2.12 `vec3 & vec3 ::operator-= (const vec3 & vv)`

5.20.2.13 `vec3 vec3 ::operator/ (const float t) const`

Voici le graphe d'appel pour cette fonction :



5.20.2.14 `vec3 & vec3 ::operator/= (const float t)`

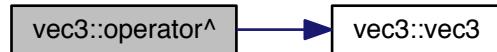
5.20.2.15 `vec3 & vec3 ::operator= (const vec3 & vv)`

5.20.2.16 `float & vec3 ::operator[] (int i)`

5.20.2.17 `float vec3 ::operator[] (int i) const`

5.20.2.18 vec3 vec3 ::operator^ (const vec3 & vv) const

Voici le graphe d'appel pour cette fonction :



5.20.3 Documentation des données membres

5.20.3.1 float vec3 ::v[3]

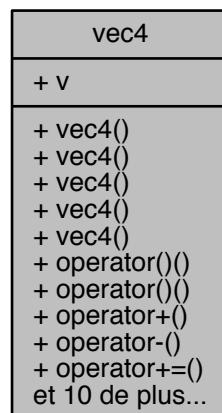
La documentation de cette structure a été générée à partir des fichiers suivants :

- [helper.hpp](#)
- [helper.cpp](#)

5.21 Référence de la structure vec4

```
#include <helper.hpp>
```

Graphe de collaboration de vec4 :



Fonctions membres publiques

- [vec4 \(float x, float y, float z, float w\)](#)
- [vec4 \(const vec2 &vv, float z, float w\)](#)
- [vec4 \(const vec3 &vv, float w\)](#)
- [vec4 \(const vec4 &vv\)](#)
- [vec4 \(\)](#)

- `vec4 & operator()` (`const vec4 &`)
- `vec4 & operator()` (`float x, float y, float z, float w`)
- `vec4 operator+ (const vec4 &)` const
- `vec4 operator- (const vec4 &)` const
- `vec4 & operator+= (const vec4 &)`
- `vec4 & operator-= (const vec4 &)`
- `vec4 operator- (void) const`
- `vec4 operator* (const float) const`
- `vec4 operator/ (const float) const`
- `vec4 & operator*= (const float)`
- `vec4 & operator/= (const float)`
- `float & operator[] (int)`
- `float operator[] (int) const`
- `float length () const`
- `void normalize ()`

Attributs publics

- `float v [4]`

5.21.1 Documentation des constructeurs et destructeur

5.21.1.1 `vec4 ::vec4 (float x, float y, float z, float w)`

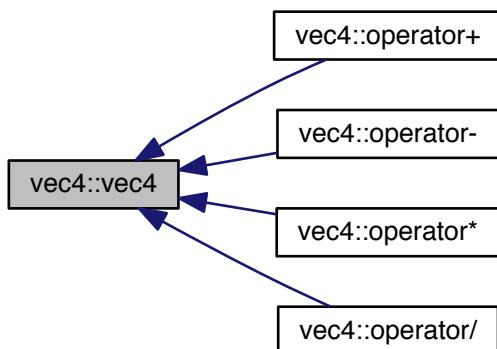
5.21.1.2 `vec4 ::vec4 (const vec2 & vv, float z, float w)`

5.21.1.3 `vec4 ::vec4 (const vec3 & vv, float w)`

5.21.1.4 `vec4 ::vec4 (const vec4 & vv)`

5.21.1.5 `vec4 ::vec4 ()`

Voici le graphe des appels de cette fonction :



5.21.2 Documentation des fonctions membres

5.21.2.1 float vec4 ::length () const

Voici le graphe des appels de cette fonction :

**5.21.2.2 void vec4 ::normalize ()**

Voici le graphe d'appel pour cette fonction :

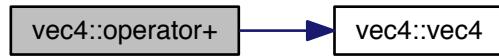
**5.21.2.3 vec4 & vec4 ::operator() (const vec4 & vv)****5.21.2.4 vec4 & vec4 ::operator() (float x, float y, float z, float w)****5.21.2.5 vec4 vec4 ::operator* (const float t) const**

Voici le graphe d'appel pour cette fonction :

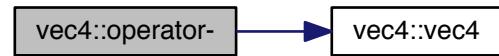
**5.21.2.6 vec4 & vec4 ::operator*=(const float t)**

5.21.2.7 vec4 vec4 ::operator+ (const vec4 & vv) const

Voici le graphe d'appel pour cette fonction :

**5.21.2.8 vec4 & vec4 ::operator+=(const vec4 & vv)****5.21.2.9 vec4 vec4 ::operator- (const vec4 & vv) const**

Voici le graphe d'appel pour cette fonction :

**5.21.2.10 vec4 vec4 ::operator- (void) const**

Voici le graphe d'appel pour cette fonction :

**5.21.2.11 vec4 & vec4 ::operator-= (const vec4 & vv)**

5.21.2.12 `vec4 vec4 ::operator/ (const float t) const`

Voici le graphe d'appel pour cette fonction :



5.21.2.13 `vec4 & vec4 ::operator/= (const float t)`

5.21.2.14 `float & vec4 ::operator[](int i)`

5.21.2.15 `float vec4 ::operator[](int i) const`

5.21.3 Documentation des données membres

5.21.3.1 `float vec4 ::v[4]`

La documentation de cette structure a été générée à partir des fichiers suivants :

- [helper.hpp](#)
- [helper.cpp](#)

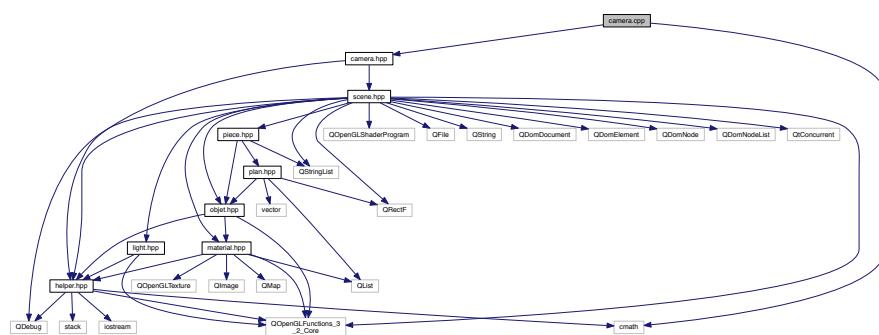
Chapitre 6

Documentation des fichiers

6.1 Référence du fichier camera.cpp

```
#include "camera.hpp"
#include <cmath>
```

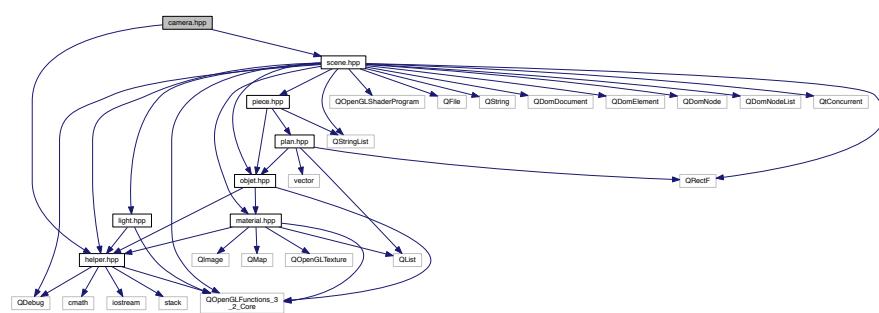
Graphe des dépendances par inclusion de camera.cpp :



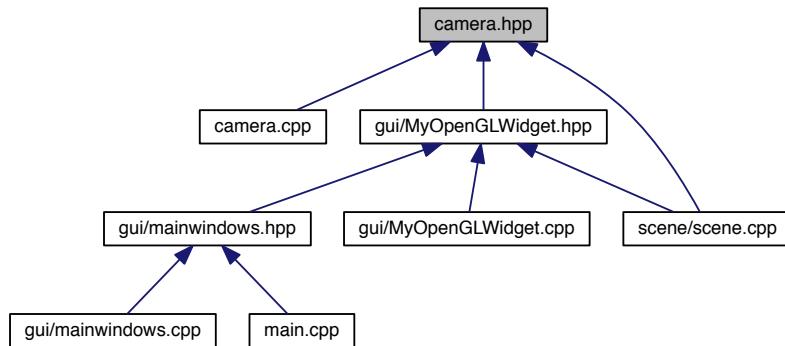
6.2 Référence du fichier camera.hpp

```
#include "helper.hpp"
#include "scene.hpp"
```

Graphe des dépendances par inclusion de camera.hpp :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



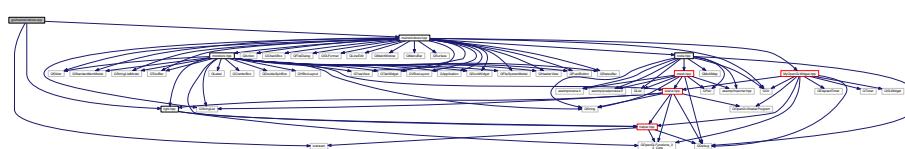
Classes

- class [Camera](#)
Camera pour Scene.

6.3 Référence du fichier `gui/mainwindows.cpp`

```
#include "mainwindows.hpp"
#include <iostream>
#include "light.hpp"
```

Graphe des dépendances par inclusion de `mainwindows.cpp` :

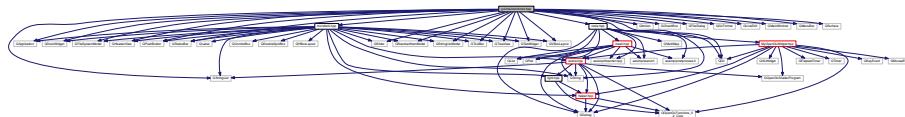


6.4 Référence du fichier `gui/mainwindows.hpp`

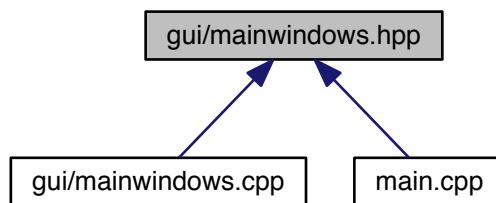
```
#include "mondock.hpp"
```

```
#include "node.hpp"
#include "MyOpenGLWidget.hpp"
#include <QAction>
#include <QApplication>
#include <QCheckBox>
#include <QDir>
#include <QDockWidget>
#include <QFileDialog>
#include <QFileSystemModel>
#include <QGLFormat>
#include <QHeaderView>
#include <QLineEdit>
#include <QMainWindow>
#include <QMenuBar>
#include <QPushButton>
#include <QSurface>
#include <QSlider>
#include <QStandardItemModel>
#include <QStatusBar>
#include <QStringList>
#include <QStringListModel>
#include <QToolBar>
#include <QTreeView>
#include <QTabWidget>
#include <QVBoxLayout>
```

Graphe des dépendances par inclusion de mainwindows.hpp :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



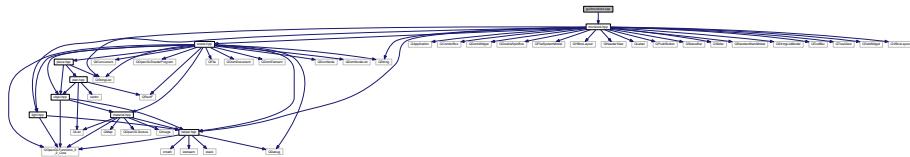
Classes

— class [MainWindow](#)

6.5 Référence du fichier gui/mondock.cpp

```
#include "mondock.hpp"
```

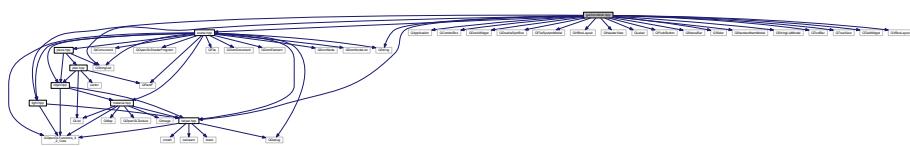
Graphe des dépendances par inclusion de mondock.cpp :



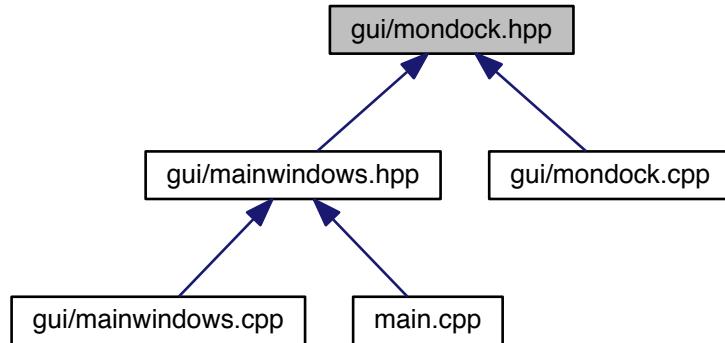
6.6 Référence du fichier gui/mondock.hpp

```
#include "scene.hpp"
#include "helper.hpp"
#include "light.hpp"
#include <QApplication>
#include <QComboBox>
#include <QDockWidget>
#include <QDoubleSpinBox>
#include <QFileSystemModel>
#include <QHBoxLayout>
#include <QHeaderView>
#include <QLabel>
#include <QPushButton>
#include <QStatusBar>
#include <QSlider>
#include <QString>
#include <QStandardItemModel>
#include <QStringListModel>
#include <QStringList>
#include <QToolBar>
#include <QTreeView>
#include <QTabWidget>
#include <QVBoxLayout>
```

Graphe des dépendances par inclusion de mondock.hpp :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



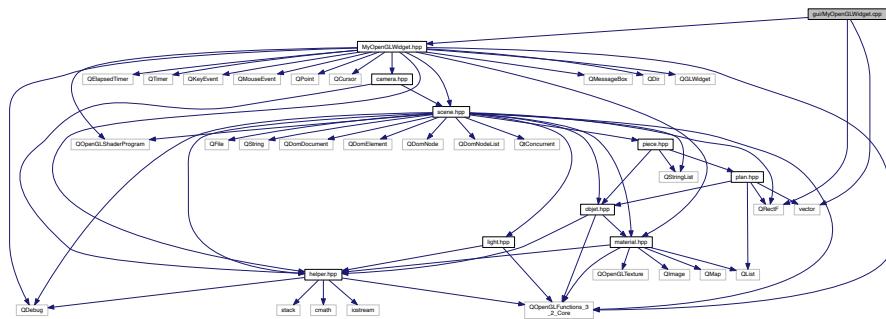
Classes

- class [Mondock](#)

6.7 Référence du fichier gui/MyOpenGLWidget.cpp

```
#include "MyOpenGLWidget.hpp"
#include <QRectF>
#include <vector>
```

Graphe des dépendances par inclusion de `MyOpenGLWidget.cpp` :

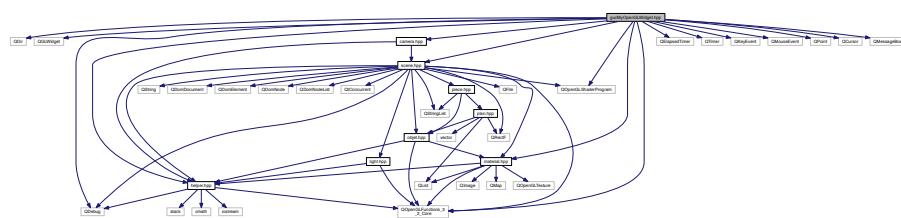


6.8 Référence du fichier gui/MyOpenGLWidget.hpp

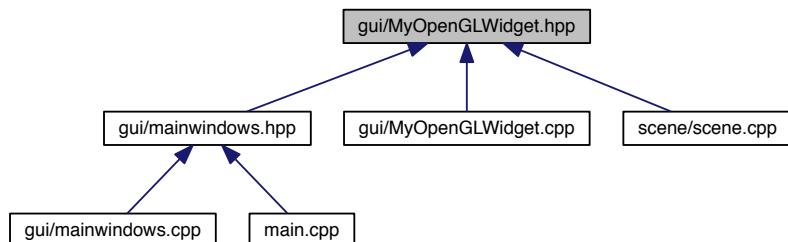
```
#include <QDir>
```

```
#include <QGLWidget>
#include <QOpenGLFunctions_3_2_Core>
#include <QOpenGLShaderProgram>
#include <QEapsedTimer>
#include <QTimer>
#include <QKeyEvent>
#include <QMouseEvent>
#include <QPoint>
#include <QCursor>
#include <QDebug>
#include "helper.hpp"
#include "camera.hpp"
#include "material.hpp"
#include "scene.hpp"
#include " QMessageBox"
```

Graphe des dépendances par inclusion de MyOpenGLWidget.hpp :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



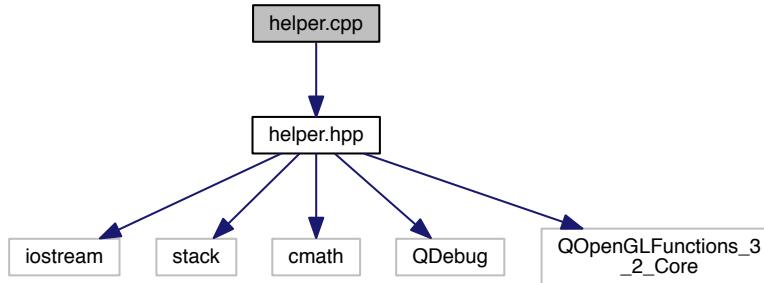
Classes

- class [MyOpenGLWidget](#)

6.9 Référence du fichier helper.cpp

```
#include "helper.hpp"
```

Graphe des dépendances par inclusion de helper.cpp :



Fonctions

```

— void popMatrix ()
— void pushMatrix (mat4 m)
— mat4 currentMatrix ()
— mat4 translationMatrix (float x, float y, float z)
— mat4 translationMatrix (const vec3 &vv)
— mat4 translate (const mat4 &mm, const vec3 &vv)
— mat4 XrotationMatrix (float alpha)
— mat4 YrotationMatrix (float alpha)
— mat4 ZrotationMatrix (float alpha)
— mat4 Xrotate (const mat4 &mm, float alpha)
— mat4 Yrotate (const mat4 &mm, float alpha)
— mat4 Zrotate (const mat4 &mm, float alpha)
— mat4 scaleMatrix (float x, float y, float z)
— mat4 scaleMatrix (const vec3 &vv)
— mat4 scales (const mat4 &mm, const vec3 &vv)
— mat4 viewMatrix (const vec3 &eye, const vec3 &center, const vec3 &up)
— mat4 projectionMatrix (float fovy, float aspect, float near, float far)
— template<typename charT , typename traits >
  std : :basic_ostream< charT,
  traits > & operator<< (std : :basic_ostream< charT, traits > &os, const vec2 &vv)
— qDebug operator<< (QDebug dbg, const vec2 &v)
— void print (const vec2 &vv)
— template<typename charT , typename traits >
  std : :basic_ostream< charT,
  traits > & operator<< (std : :basic_ostream< charT, traits > &os, const vec3 &vv)
— qDebug operator<< (QDebug dbg, const vec3 &v)
— void print (const vec3 &vv)
— template<typename charT , typename traits >
  std : :basic_ostream< charT,
  traits > & operator<< (std : :basic_ostream< charT, traits > &os, const vec4 &vv)
— qDebug operator<< (QDebug dbg, const vec4 &v)
— void print (const vec4 &vv)
— template<typename charT , typename traits >
  std : :basic_ostream< charT,
  traits > & operator<< (std : :basic_ostream< charT, traits > &os, const mat3 &mm)
— qDebug operator<< (QDebug dbg, const mat3 &v)
— void print (const mat3 &mm)
— template<typename charT , typename traits >
  std : :basic_ostream< charT,
  traits > & operator<< (std : :basic_ostream< charT, traits > &os, const mat4 &mm)
— qDebug operator<< (QDebug dbg, const mat4 &v)
— void print (const mat4 &mm)
— void __openGL_check_error (const char *file, int line)
— void setProjectionMatrix (mat4 projection)
— void setModelMatrix (mat4 model)
— void setViewMatrix (mat4 view)
— void setActiveShader (GLuint shader)
  
```

— GLuint **getActiveShader ()**

Variables

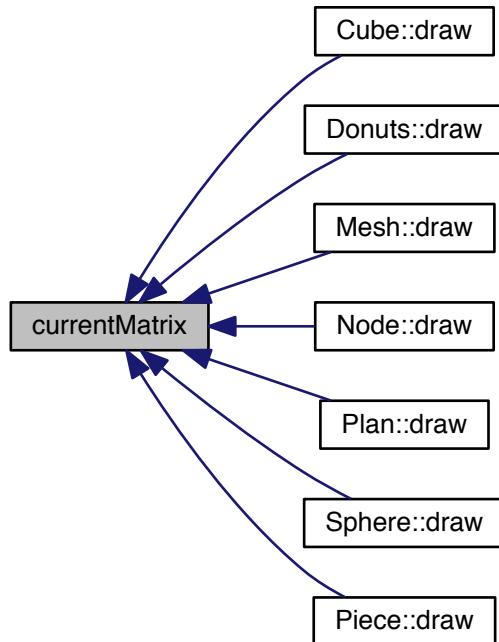
— static std ::stack< mat4 > matrix_stack
— static mat4 currentProjectionMatrix
— static mat4 currentViewMatrix
— static mat4 currentModelMatrix
— static GLuint activeShader = 0

6.9.1 Documentation des fonctions

6.9.1.1 void __openGL_check_error (const char * *file*, int *line*)

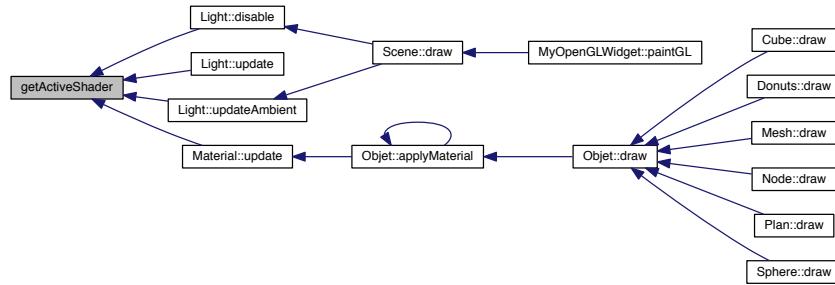
6.9.1.2 mat4 **currentMatrix ()**

Voici le graphe des appelleurs de cette fonction :



6.9.1.3 GLuint getActiveShader ()

Voici le graphe des appelants de cette fonction :



6.9.1.4 template<typename charT , typename traits > std : :basic_ostream<charT,traits>& operator<< (std : :basic_ostream< charT, traits > & os, const vec2 & vv)

6.9.1.5 qDebug operator<< (QDebug dbg, const vec2 & v)

6.9.1.6 template<typename charT , typename traits > std : :basic_ostream<charT,traits>& operator<< (std : :basic_ostream< charT, traits > & os, const vec3 & vv)

6.9.1.7 qDebug operator<< (QDebug dbg, const vec3 & v)

6.9.1.8 template<typename charT , typename traits > std : :basic_ostream<charT,traits>& operator<< (std : :basic_ostream< charT, traits > & os, const vec4 & vv)

6.9.1.9 qDebug operator<< (QDebug dbg, const vec4 & v)

6.9.1.10 template<typename charT , typename traits > std : :basic_ostream<charT,traits>& operator<< (std : :basic_ostream< charT, traits > & os, const mat3 & mm)

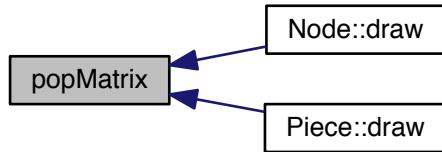
6.9.1.11 qDebug operator<< (QDebug dbg, const mat3 & v)

6.9.1.12 template<typename charT , typename traits > std : :basic_ostream<charT,traits>& operator<< (std : :basic_ostream< charT, traits > & os, const mat4 & mm)

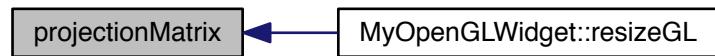
6.9.1.13 qDebug operator<< (QDebug dbg, const mat4 & v)

6.9.1.14 void popMatrix()

Voici le graphe des appels de cette fonction :

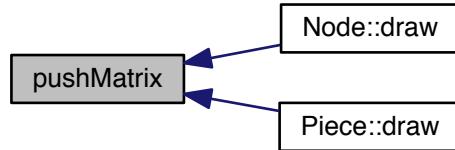
**6.9.1.15 void print(const vec2 & vv)****6.9.1.16 void print(const vec3 & vv)****6.9.1.17 void print(const vec4 & vv)****6.9.1.18 void print(const mat3 & mm)****6.9.1.19 void print(const mat4 & mm)****6.9.1.20 mat4 projectionMatrix(float fovy, float aspect, float near, float far)**

Voici le graphe des appels de cette fonction :



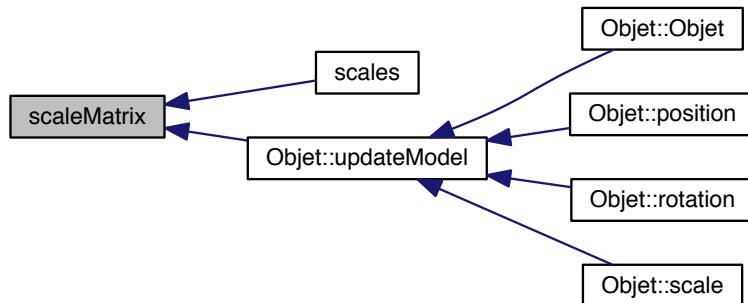
6.9.1.21 void pushMatrix (mat4 m)

Voici le graphe des appelants de cette fonction :



6.9.1.22 mat4 scaleMatrix (float x, float y, float z)

Voici le graphe des appelants de cette fonction :



6.9.1.23 mat4 scaleMatrix (const vec3 & vv)

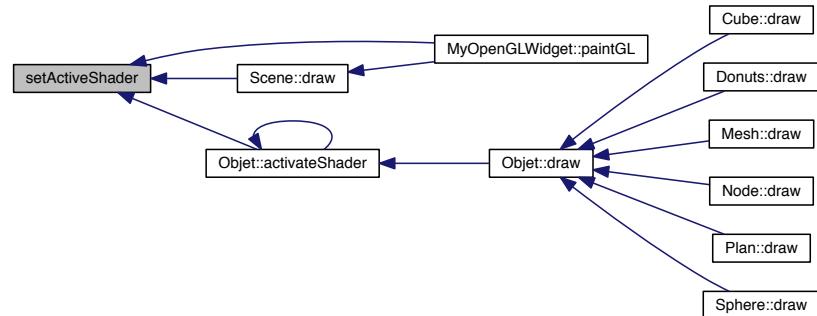
6.9.1.24 mat4 scales (const mat4 & mm, const vec3 & vv)

Voici le graphe d'appel pour cette fonction :



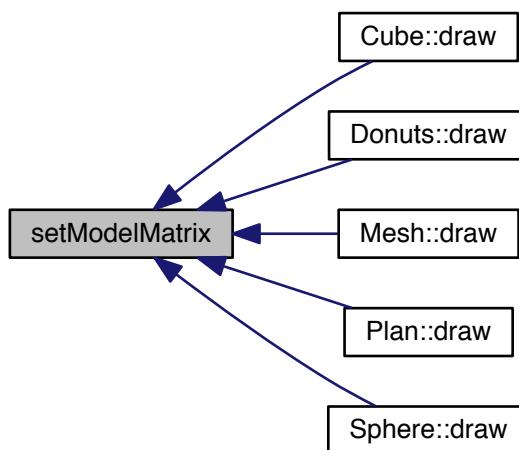
6.9.1.25 void setActiveShader (GLuint shader)

Voici le graphe des appelants de cette fonction :



6.9.1.26 void setModelMatrix (mat4 model)

Voici le graphe des appelants de cette fonction :

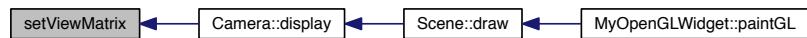


6.9.1.27 void setProjectionMatrix (mat4 *projection*)

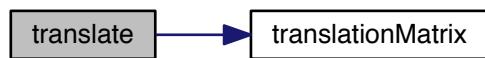
Voici le graphe des appelants de cette fonction :

6.9.1.28 void setViewMatrix (mat4 *view*)

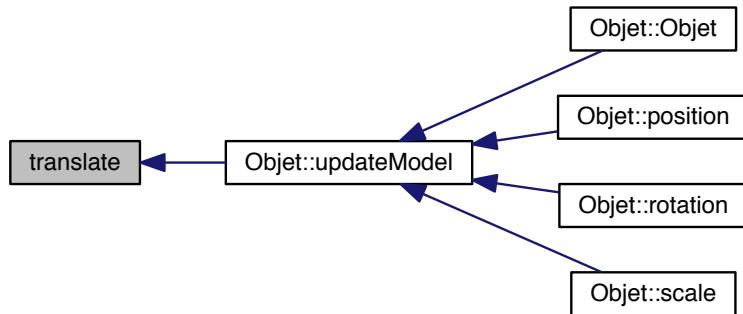
Voici le graphe des appelants de cette fonction :

6.9.1.29 mat4 translate (const mat4 & *mm*, const vec3 & *vv*)

Voici le graphe d'appel pour cette fonction :

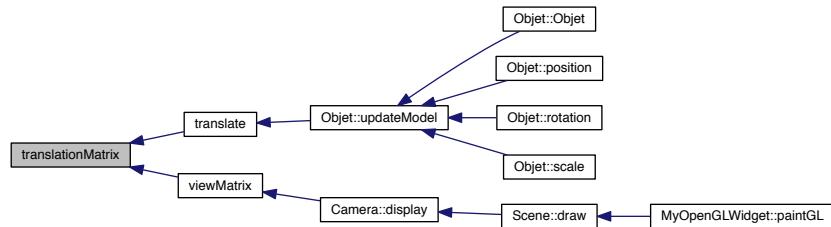


Voici le graphe des appelants de cette fonction :



6.9.1.30 mat4 translationMatrix (float x, float y, float z)

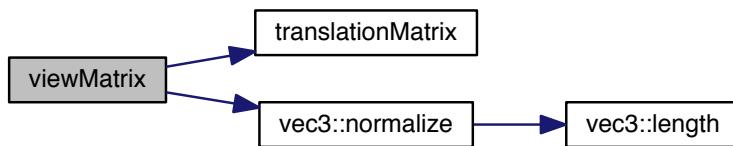
Voici le graphe des appelants de cette fonction :



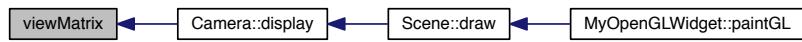
6.9.1.31 mat4 translationMatrix (const vec3 & vv)

6.9.1.32 mat4 viewMatrix (const vec3 & eye, const vec3 & center, const vec3 & up)

Voici le graphe d'appel pour cette fonction :

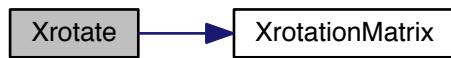


Voici le graphe des appelants de cette fonction :

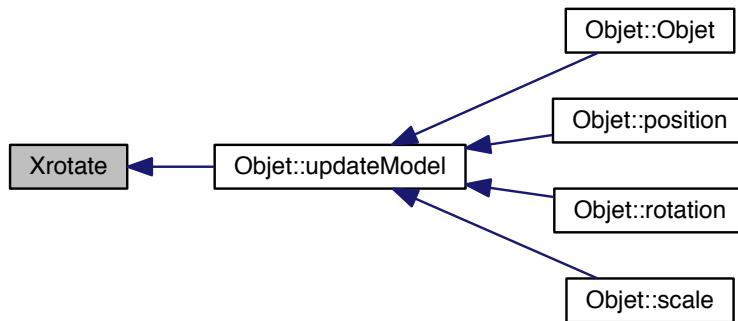


6.9.1.33 mat4 Xrotate (const mat4 & mm, float alpha)

Voici le graphe d'appel pour cette fonction :

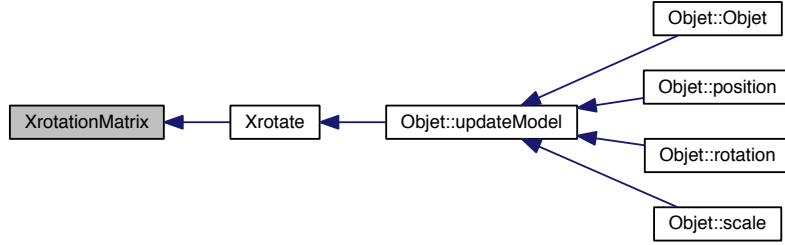


Voici le graphe des appelants de cette fonction :



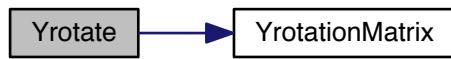
6.9.1.34 mat4 XrotationMatrix (float alpha)

Voici le graphe des appelants de cette fonction :

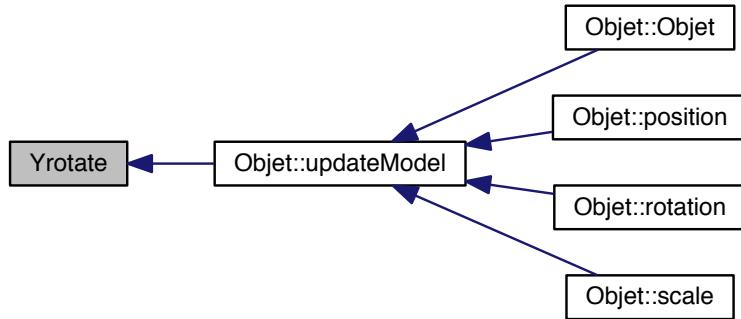


6.9.1.35 mat4 Yrotate (const mat4 & mm, float alpha)

Voici le graphe d'appel pour cette fonction :

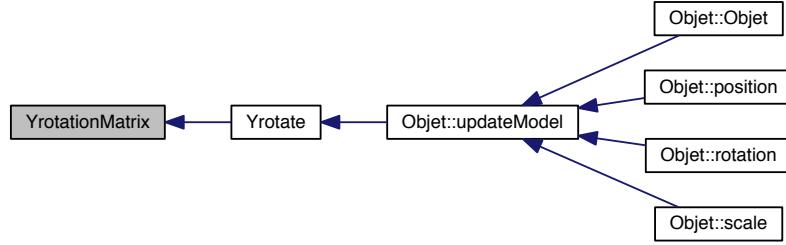


Voici le graphe des appelants de cette fonction :



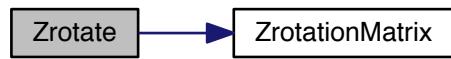
6.9.1.36 mat4 YrotationMatrix (float alpha)

Voici le graphe des appelants de cette fonction :

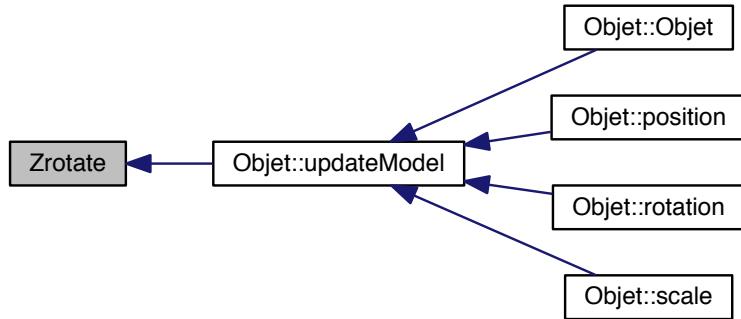


6.9.1.37 mat4 Zrotate (const mat4 & mm, float alpha)

Voici le graphe d'appel pour cette fonction :

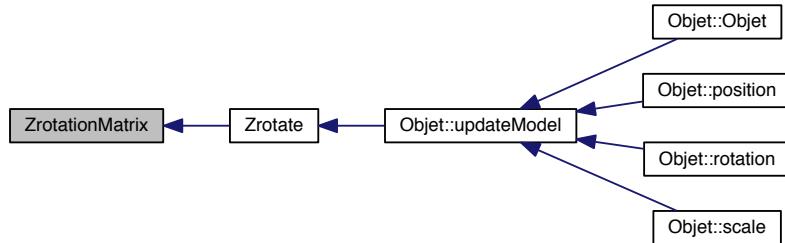


Voici le graphe des appelants de cette fonction :



6.9.1.38 mat4 ZrotationMatrix (float alpha)

Voici le graphe des appelants de cette fonction :



6.9.2 Documentation des variables

6.9.2.1 GLuint activeShader = 0 [static]

6.9.2.2 mat4 currentModelMatrix [static]

6.9.2.3 mat4 currentProjectionMatrix [static]

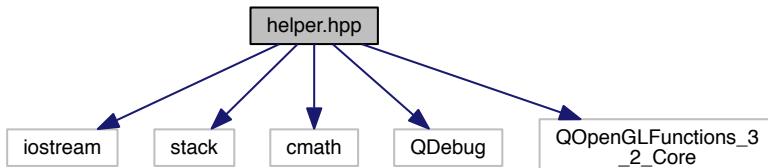
6.9.2.4 mat4 currentViewMatrix [static]

6.9.2.5 std::stack<mat4> matrix_stack [static]

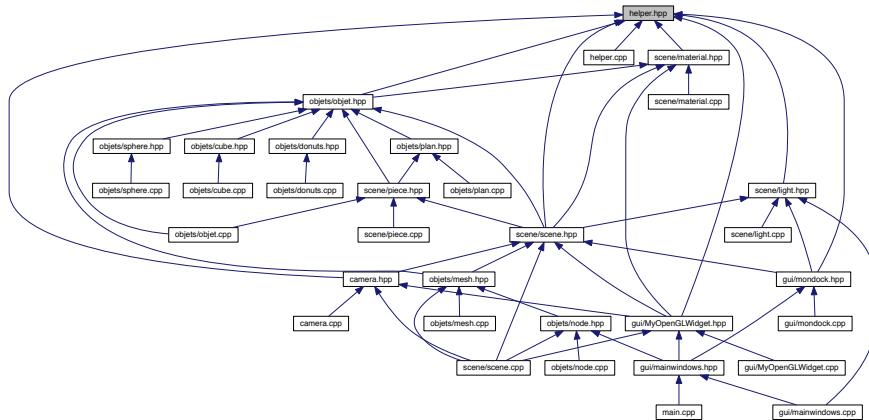
6.10 Référence du fichier helper.hpp

```
#include <iostream>
#include <stack>
#include <cmath>
#include <QDebug>
#include <QOpenGLFunctions_3_2_Core>
```

Graphe des dépendances par inclusion de helper.hpp :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



Classes

- struct `vec2`
- struct `vec3`
- struct `vec4`
- struct `mat3`
- struct `mat4`

Macros

- #define `M_PI` 3.141592
- #define `to_rad(alpha)` (`alpha * (M_PI / 180.0f)`)
- #define `openGL_check_error()` `_openGL_check_error(_FILE_, _LINE_)`

Définitions de type

- typedef struct `vec2` `vec2`
- typedef struct `vec3` `vec3`
- typedef struct `vec4` `vec4`
- typedef struct `mat3` `mat3`
- typedef struct `mat4` `mat4`

Fonctions

- void `_openGL_check_error` (const char *file, int line)
- void `popMatrix` ()
- void `pushMatrix` (`mat4`)
- `mat4 currentMatrix` ()
- `mat4 translationMatrix` (float x, float y, float z)
- `mat4 translationMatrix` (const `vec3` &)
- `mat4 translate` (const `mat4` &, const `vec3` &)
- `mat4 XrotationMatrix` (float alpha)
- `mat4 YrotationMatrix` (float alpha)
- `mat4 ZrotationMatrix` (float alpha)
- `mat4 Xrotate` (const `mat4` &, float alpha)
- `mat4 Yrotate` (const `mat4` &, float alpha)
- `mat4 Zrotate` (const `mat4` &, float alpha)
- `mat4 scaleMatrix` (float x, float y, float z)
- `mat4 scaleMatrix` (const `vec3` &)
- `mat4 scales` (const `mat4` &, const `vec3` &)
- `mat4 viewMatrix` (const `vec3` &eye, const `vec3` ¢er, const `vec3` &up)
- `mat4 projectionMatrix` (float fov, float aspect, float zNear, float zFar)
- template<typename charT , typename traits >
std : :basic_ostream< charT , traits > &os, const `vec2` &)
traits > & `operator<<` (std : :basic_ostream< charT , traits > &os, const `vec2` &)

```

— QDebug operator<< (QDebug dbg, const vec2 &v)
— void print (const vec2 &)
— template<typename charT , typename traits >
    std : :basic_ostream< charT,
    traits > & operator<< (std : :basic_ostream< charT, traits > &os, const vec3 &)
— QDebug operator<< (QDebug dbg, const vec3 &v)
— void print (const vec3 &)
— template<typename charT , typename traits >
    std : :basic_ostream< charT,
    traits > & operator<< (std : :basic_ostream< charT, traits > &os, const vec4 &)
— QDebug operator<< (QDebug dbg, const vec4 &v)
— void print (const vec4 &)
— template<typename charT , typename traits >
    std : :basic_ostream< charT,
    traits > & operator<< (std : :basic_ostream< charT, traits > &os, const mat3 &)
— QDebug operator<< (QDebug dbg, const mat3 &m)
— void print (const mat3 &)
— template<typename charT , typename traits >
    std : :basic_ostream< charT,
    traits > & operator<< (std : :basic_ostream< charT, traits > &os, const mat4 &)
— QDebug operator<< (QDebug dbg, const mat4 &m)
— void print (const mat4 &)
— void setProjectionMatrix (mat4 projection)
— void setModelMatrix (mat4 model)
— void setViewMatrix (mat4 view)
— void setActiveShader (GLuint shader)
— GLuint getActiveShader ()

```

6.10.1 Documentation des macros

6.10.1.1 #define M_PI 3.141592

6.10.1.2 #define openGL_check_error() __openGL_check_error(__FILE__, __LINE__)

6.10.1.3 #define to_rad(alpha) (alpha * (M_PI / 180.0f))

6.10.2 Documentation des définitions de type

6.10.2.1 typedef struct mat3 mat3

6.10.2.2 typedef struct mat4 mat4

6.10.2.3 typedef struct vec2 vec2

6.10.2.4 typedef struct vec3 vec3

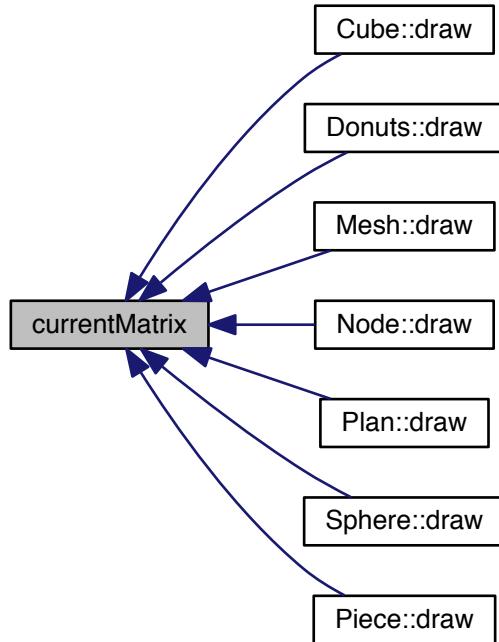
6.10.2.5 typedef struct vec4 vec4

6.10.3 Documentation des fonctions

6.10.3.1 void __openGL_check_error (const char * file, int line)

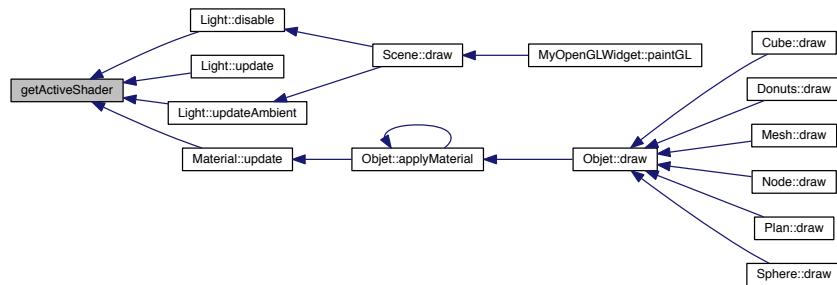
6.10.3.2 mat4 currentMatrix()

Voici le graphe des appelants de cette fonction :



6.10.3.3 GLuint getActiveShader()

Voici le graphe des appelants de cette fonction :



6.10.3.4 template<typename charT , typename traits > std :: basic_ostream<charT,traits>& operator<< (std :: basic_ostream< charT, traits > & os, const vec2 &)

6.10.3.5 qDebug operator<< (QDebug dbg, const vec2 & v)

6.10.3.6 template<typename charT , typename traits > std : :basic_ostream<charT,traits>& operator<< (std : :basic_ostream< charT, traits > & os, const vec3 &)

6.10.3.7 QDebug operator<< (QDebug dbg, const vec3 & v)

6.10.3.8 template<typename charT , typename traits > std : :basic_ostream<charT,traits>& operator<< (std : :basic_ostream< charT, traits > & os, const vec4 &)

6.10.3.9 QDebug operator<< (QDebug dbg, const vec4 & v)

6.10.3.10 template<typename charT , typename traits > std : :basic_ostream<charT,traits>& operator<< (std : :basic_ostream< charT, traits > & os, const mat3 &)

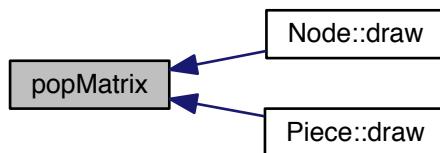
6.10.3.11 QDebug operator<< (QDebug dbg, const mat3 & m)

6.10.3.12 template<typename charT , typename traits > std : :basic_ostream<charT,traits>& operator<< (std : :basic_ostream< charT, traits > & os, const mat4 &)

6.10.3.13 QDebug operator<< (QDebug dbg, const mat4 & m)

6.10.3.14 void popMatrix ()

Voici le graphe des appels de cette fonction :



6.10.3.15 void print (const vec2 &)

6.10.3.16 void print (const vec3 &)

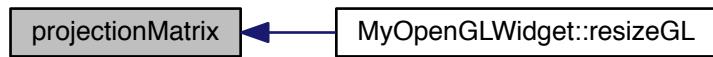
6.10.3.17 void print (const vec4 &)

6.10.3.18 void print (const mat3 &)

6.10.3.19 void print (const mat4 &)

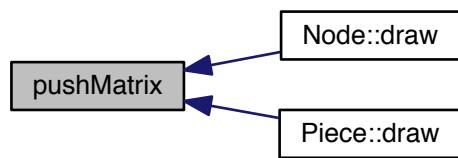
6.10.3.20 mat4 projectionMatrix (float *fov*, float *aspect*, float *zNear*, float *zFar*)

Voici le graphe des appelants de cette fonction :



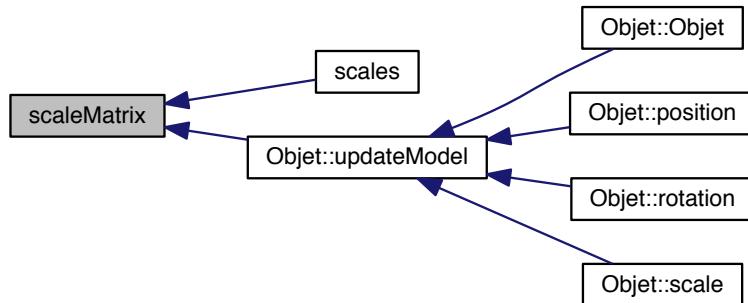
6.10.3.21 void pushMatrix (mat4)

Voici le graphe des appelants de cette fonction :



6.10.3.22 mat4 scaleMatrix (float *x*, float *y*, float *z*)

Voici le graphe des appelants de cette fonction :



6.10.3.23 `mat4 scaleMatrix (const vec3 &)`

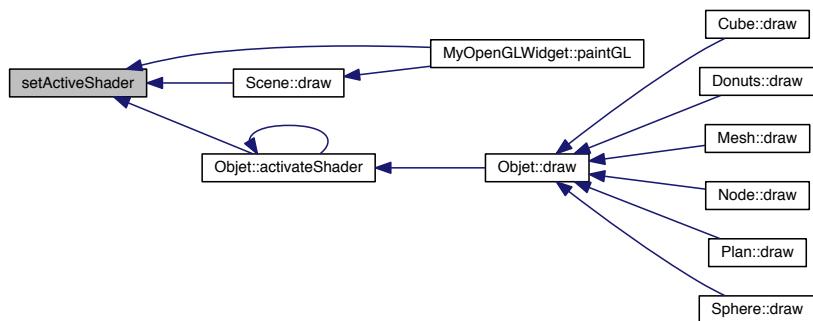
6.10.3.24 `mat4 scales (const mat4 & , const vec3 &)`

Voici le graphe d'appel pour cette fonction :



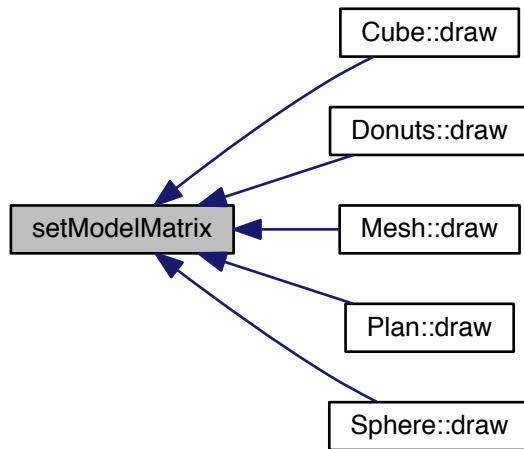
6.10.3.25 `void setActiveShader (GLuint shader)`

Voici le graphe des appels de cette fonction :



6.10.3.26 void setModelMatrix (mat4 *model*)

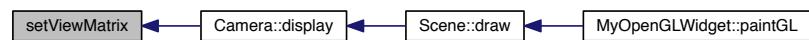
Voici le graphe des appelants de cette fonction :

6.10.3.27 void setProjectionMatrix (mat4 *projection*)

Voici le graphe des appelants de cette fonction :

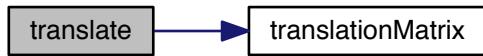
6.10.3.28 void setViewMatrix (mat4 *view*)

Voici le graphe des appelants de cette fonction :

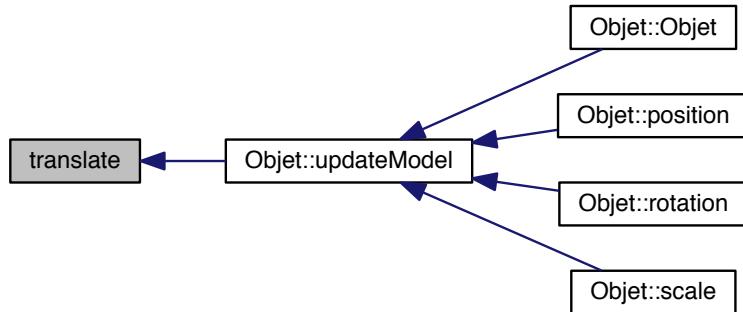


6.10.3.29 mat4 translate (const mat4 & , const vec3 &)

Voici le graphe d'appel pour cette fonction :

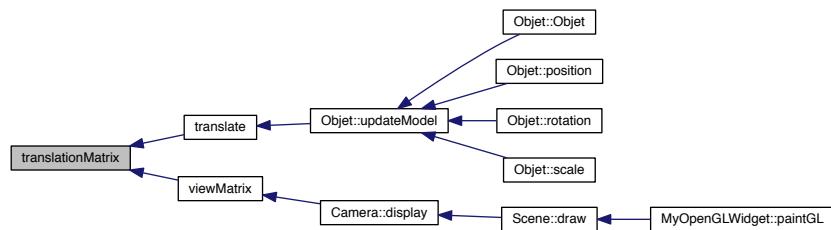


Voici le graphe des appelants de cette fonction :



6.10.3.30 mat4 translationMatrix (float x, float y, float z)

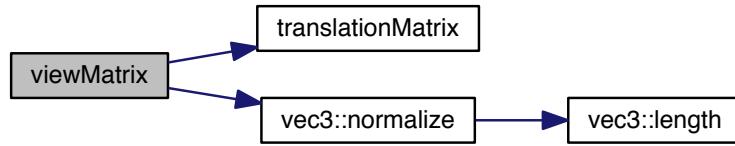
Voici le graphe des appelants de cette fonction :



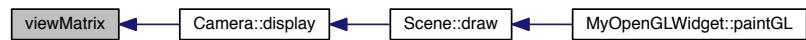
6.10.3.31 mat4 translationMatrix (const vec3 &)

6.10.3.32 mat4 viewMatrix (const vec3 & eye, const vec3 & center, const vec3 & up)

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

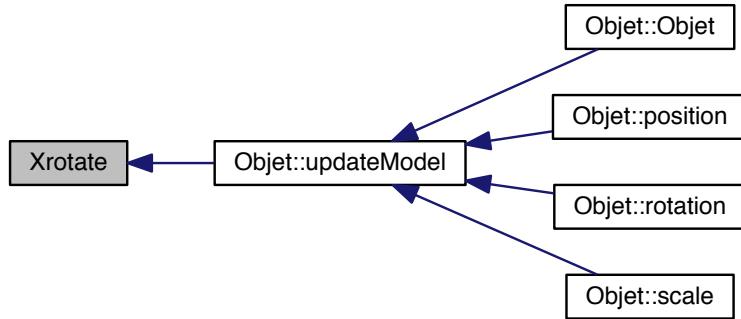


6.10.3.33 mat4 Xrotate (const mat4 & , float alpha)

Voici le graphe d'appel pour cette fonction :

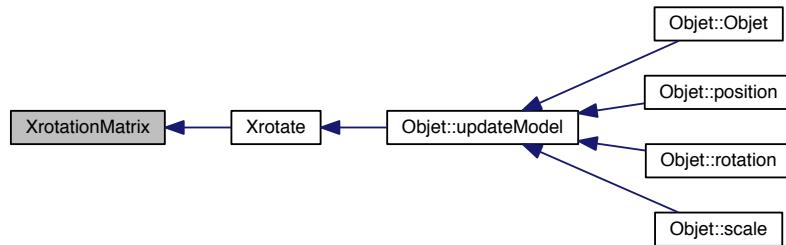


Voici le graphe des appelants de cette fonction :



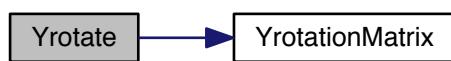
6.10.3.34 mat4 XrotationMatrix (float alpha)

Voici le graphe des appelants de cette fonction :

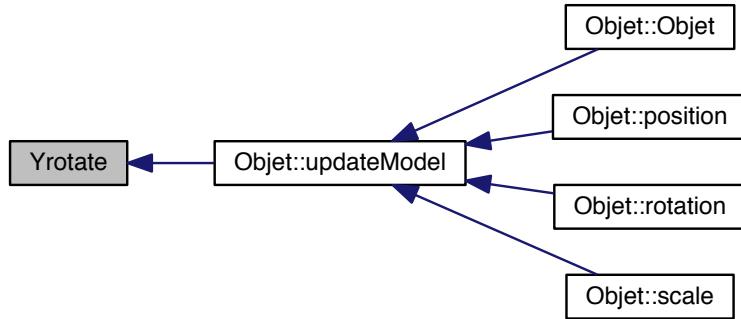


6.10.3.35 mat4 Yrotate (const mat4 & , float alpha)

Voici le graphe d'appel pour cette fonction :

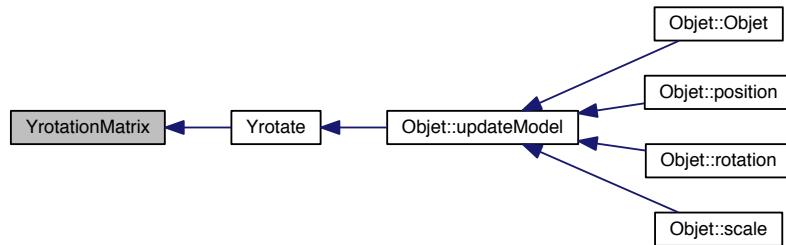


Voici le graphe des appelants de cette fonction :



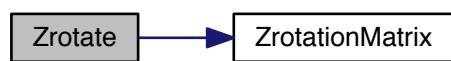
6.10.3.36 mat4 YrotationMatrix (float alpha)

Voici le graphe des appelants de cette fonction :

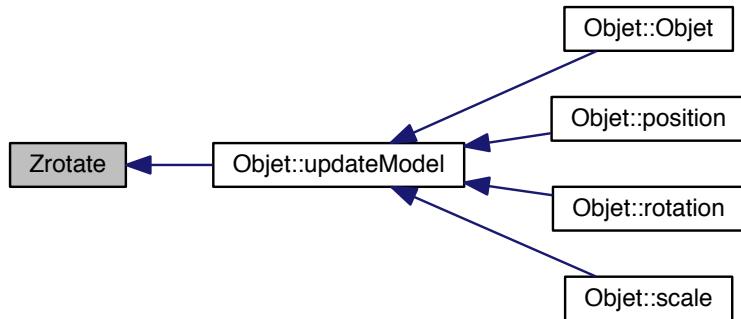


6.10.3.37 mat4 Zrotate (const mat4 & , float alpha)

Voici le graphe d'appel pour cette fonction :

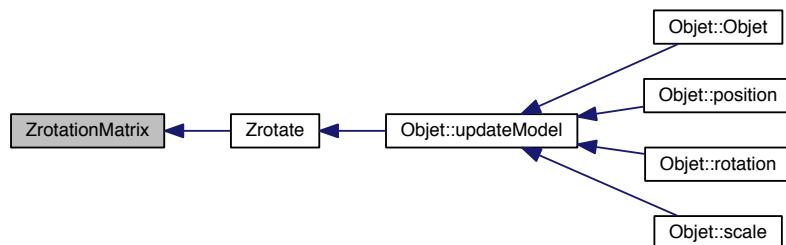


Voici le graphe des appels de cette fonction :



6.10.3.38 mat4 ZrotationMatrix (float alpha)

Voici le graphe des appels de cette fonction :

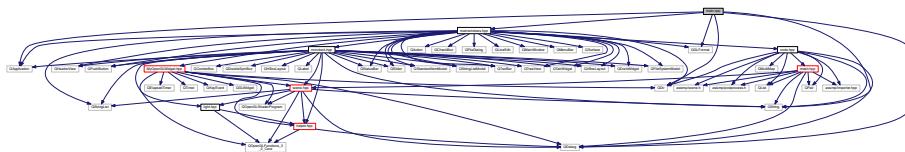


6.11 Référence du fichier main.cpp

```

#include <QApplication>
#include <QDir>
#include <QGLFormat>
#include "mainwindows.hpp"
#include <QString>
#include <QDebug>
    
```

Graphe des dépendances par inclusion de main.cpp :



Fonctions

- int `main` (int argc, char **argv)

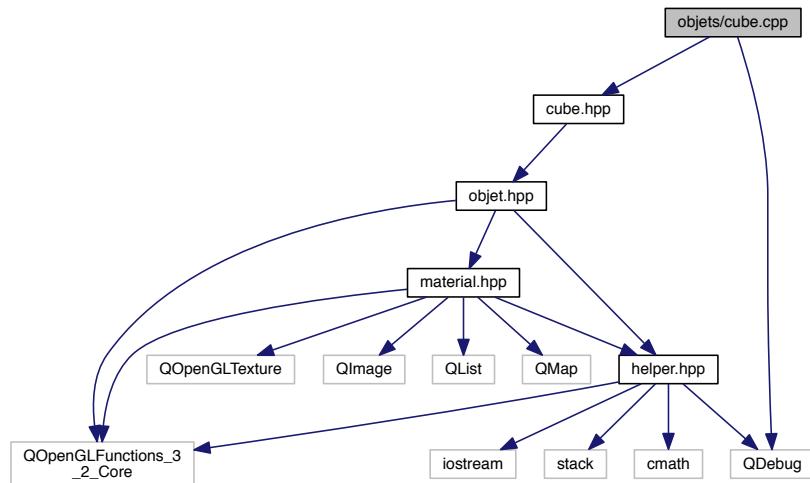
6.11.1 Documentation des fonctions

6.11.1.1 int main (int argc, char ** argv)

6.12 Référence du fichier objets/cube.cpp

```
#include "cube.hpp"
#include <QDebug>
```

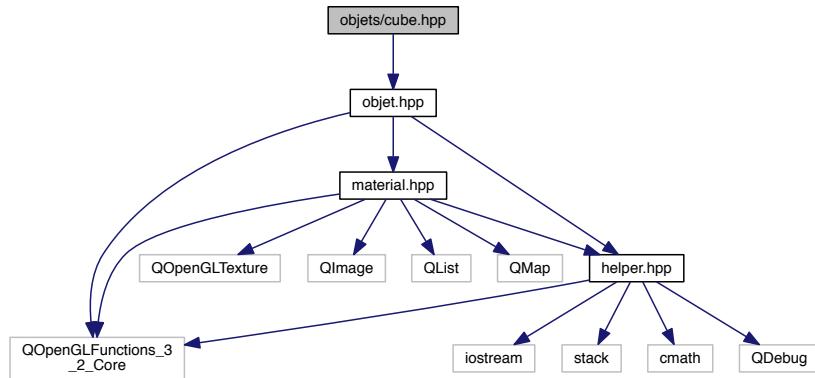
Graphe des dépendances par inclusion de cube.cpp :



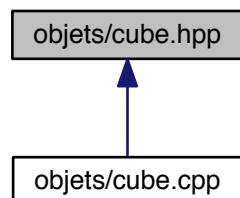
6.13 Référence du fichier objets/cube.hpp

```
#include "objet.hpp"
```

Graphique des dépendances par inclusion de cube.hpp :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



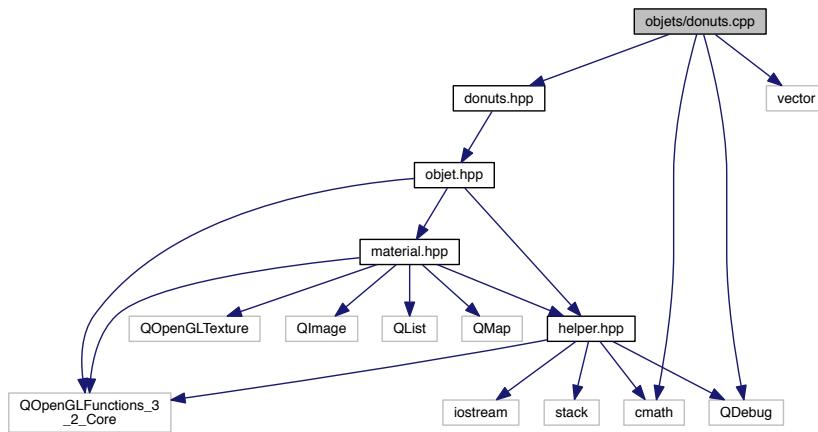
Classes

- class [Cube](#)
Primitive cube.

6.14 Référence du fichier objets/donuts.cpp

```
#include "donuts.hpp"
#include <QDebug>
#include <cmath>
#include <vector>
```

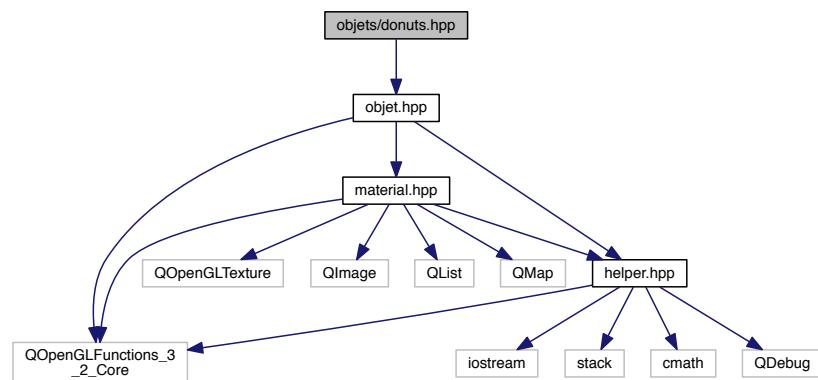
Graphe des dépendances par inclusion de donuts.cpp :



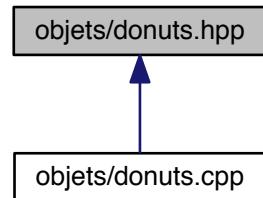
6.15 Référence du fichier objets/donuts.hpp

```
#include "objet.hpp"
```

Graphe des dépendances par inclusion de donuts.hpp :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



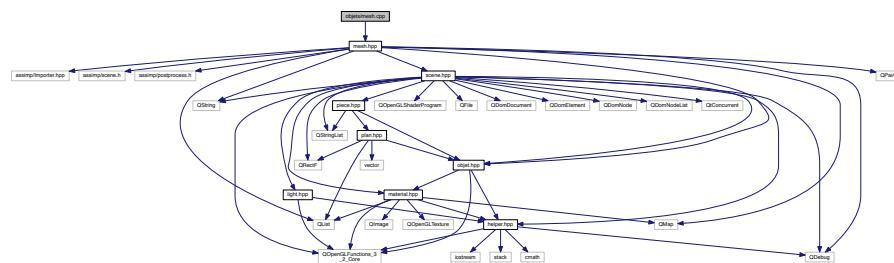
Classes

- class Donuts
Primitive tore.

6.16 Référence du fichier objets/mesh.cpp

```
#include "mesh.hpp"
```

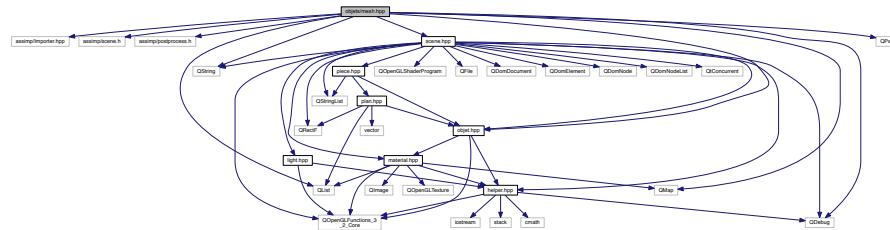
Graphe des dépendances par inclusion de mesh.cpp :



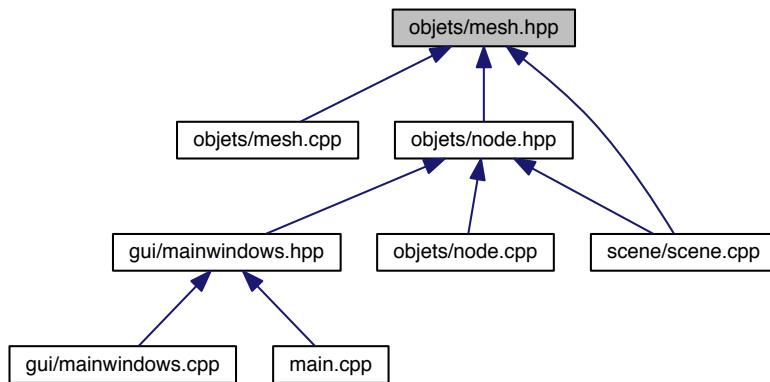
6.17 Référence du fichier objets/mesh.hpp

```
#include <assimp/Importer.hpp>
#include <assimp/scene.h>
#include <assimp/postprocess.h>
#include <QString>
#include <QDebug>
#include <QList>
#include <QMap>
#include <QPair>
#include "objet.hpp"
#include "scene.hpp"
```

Graphe des dépendances par inclusion de mesh.hpp :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



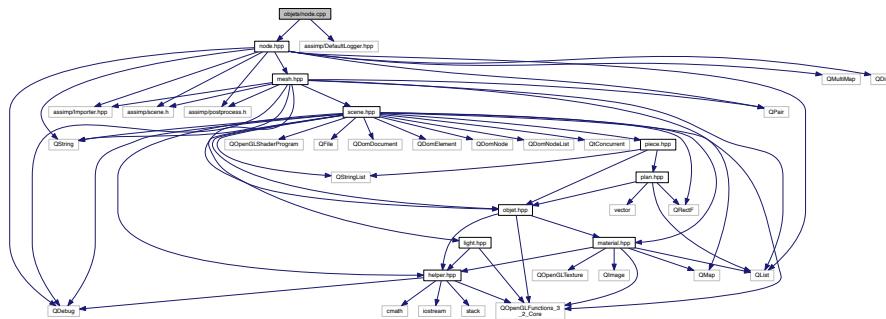
Classes

- class Mesh
 - Charge un modèle 3D.
 - struct Mesh : :MeshInfo

6.18 Référence du fichier objets/node.cpp

```
#include "node.hpp"
#include <assimp/DefaultLogger.hpp>
```

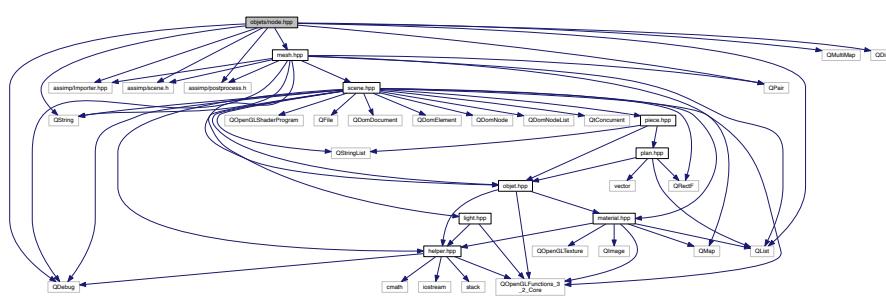
Graphe des dépendances par inclusion de node.cpp :



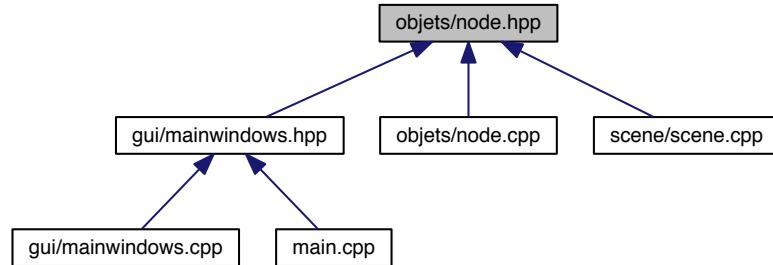
6.19 Référence du fichier objets/node.hpp

```
#include "mesh.hpp"
#include <assimp/Importer.hpp>
#include <assimp/scene.h>
#include <assimp/postprocess.h>
#include <QString>
#include <QDebug>
#include <QList>
#include <QMutiMap>
#include <QPair>
#include <ODir>
```

Graphe des dépendances par inclusion de node.hpp :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



Classes

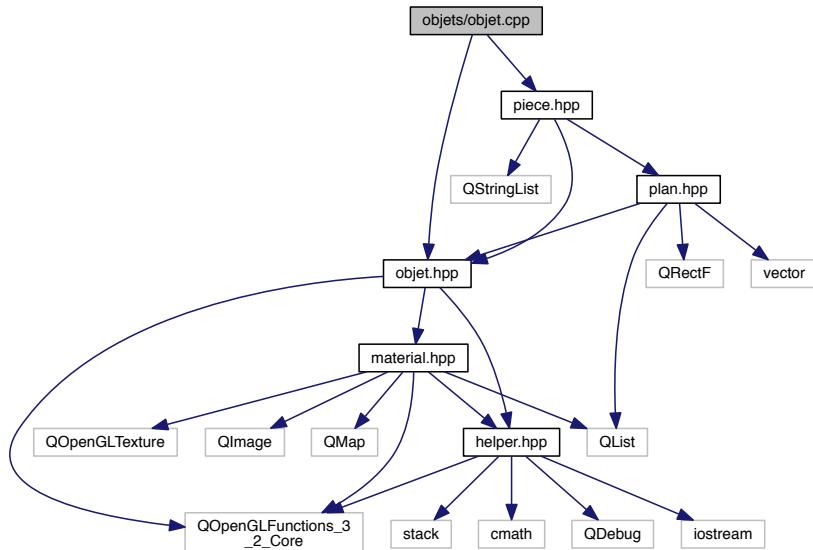
- class **Node**

Contient un node de modèle 3D.

6.20 Référence du fichier objets/objet.cpp

```
#include "objet.hpp"
#include "piece.hpp"
```

Graphe des dépendances par inclusion de objet.cpp :

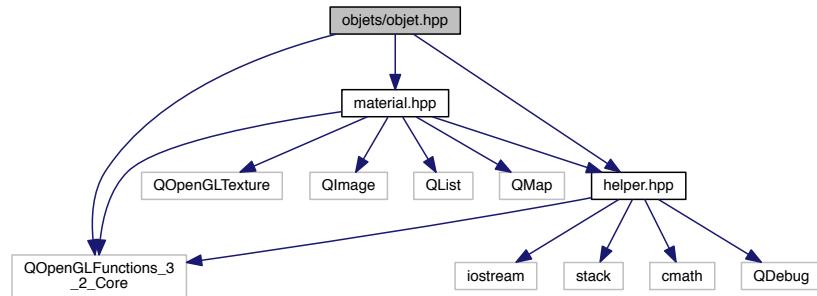


6.21 Référence du fichier objets/objet.hpp

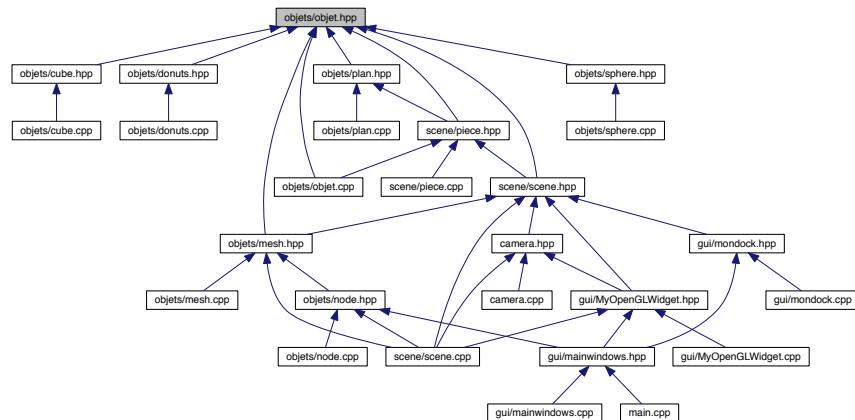
```
#include <QOpenGLFunctions_3_2_Core>
```

```
#include "material.hpp"
#include "helper.hpp"
```

Graphe des dépendances par inclusion de objet.hpp :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



Classes

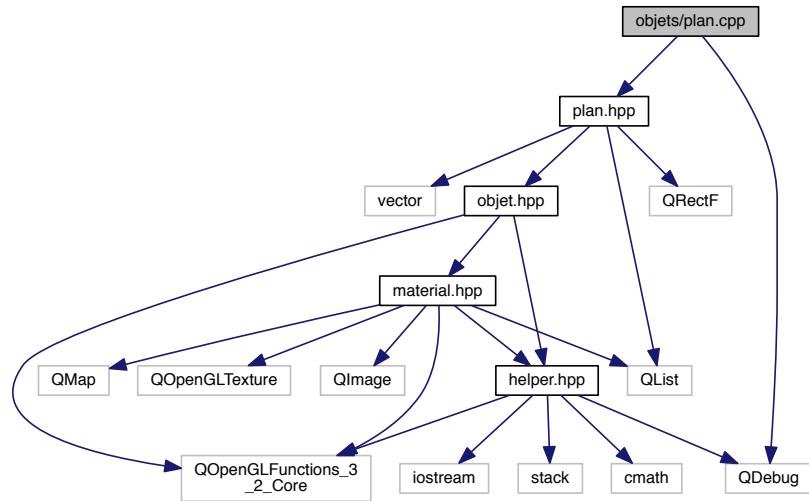
— class Objet

Classe de base affichable.

6.22 Référence du fichier objets/plan.cpp

```
#include "plan.hpp"
#include <QDebug>
```

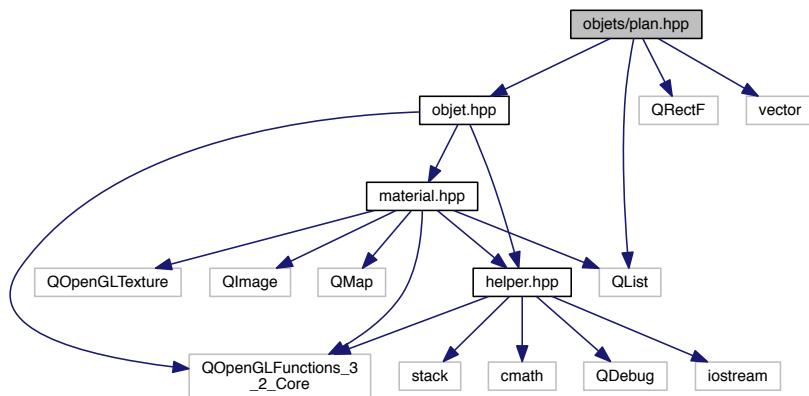
Graphe des dépendances par inclusion de plan.cpp :



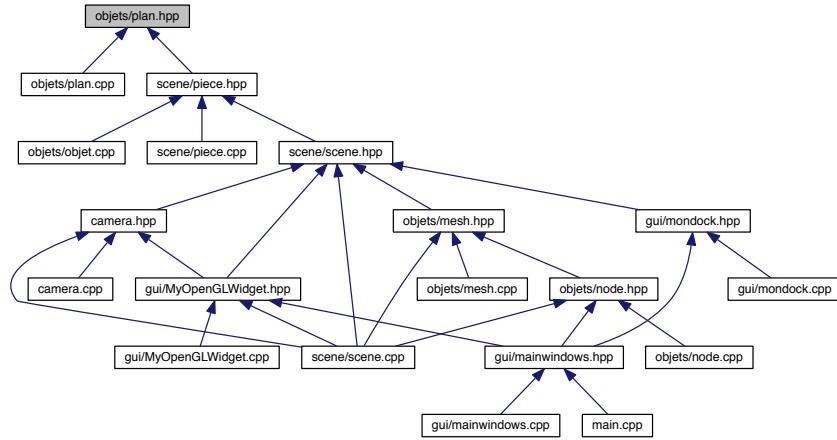
6.23 Référence du fichier objets/plan.hpp

```
#include "objet.hpp"
#include <QRectF>
#include <QList>
#include <vector>
```

Graphe des dépendances par inclusion de plan.hpp :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



Classes

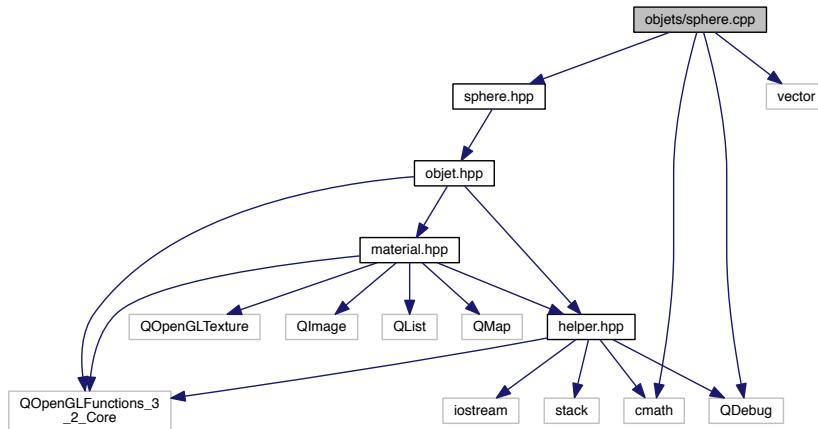
- class [Plan](#)

Plan utilisé par la [Piece](#).

6.24 Référence du fichier objets/sphere.cpp

```
#include "sphere.hpp"
#include <QDebug>
#include <cmath>
#include <vector>
```

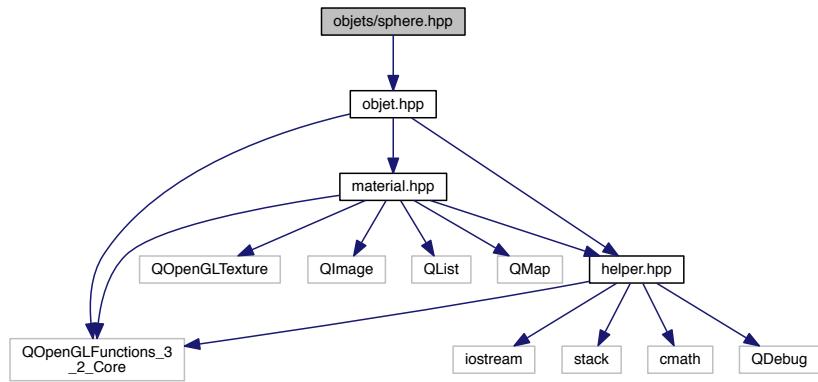
Graphe des dépendances par inclusion de sphere.cpp :



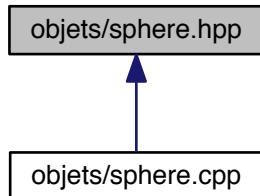
6.25 Référence du fichier objets/sphere.hpp

```
#include "objet.hpp"
```

Graphe des dépendances par inclusion de sphere.hpp :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



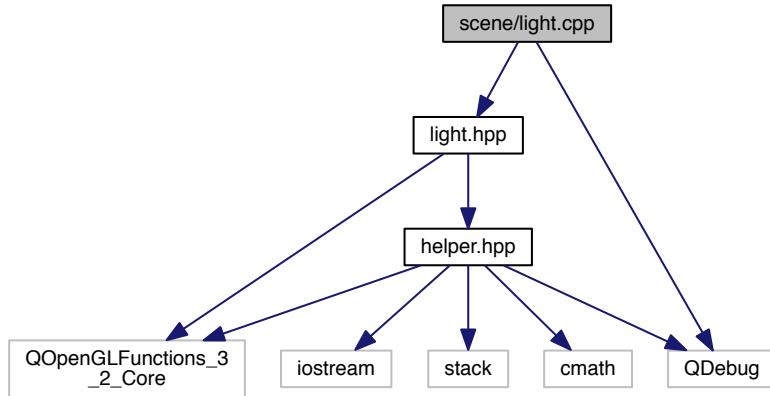
Classes

- class **Sphere**
Primitive tore.

6.26 Référence du fichier scene/light.cpp

```
#include "light.hpp"
#include <QDebug>
```

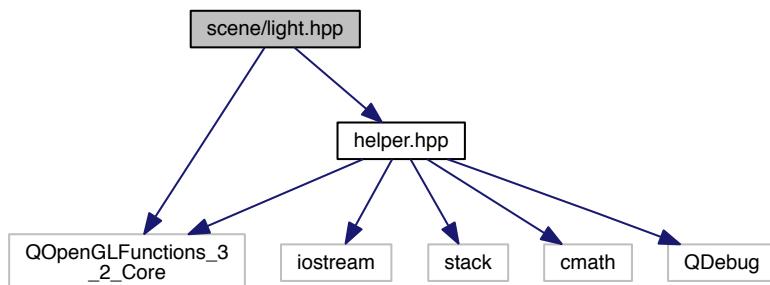
Graphe des dépendances par inclusion de light.cpp :



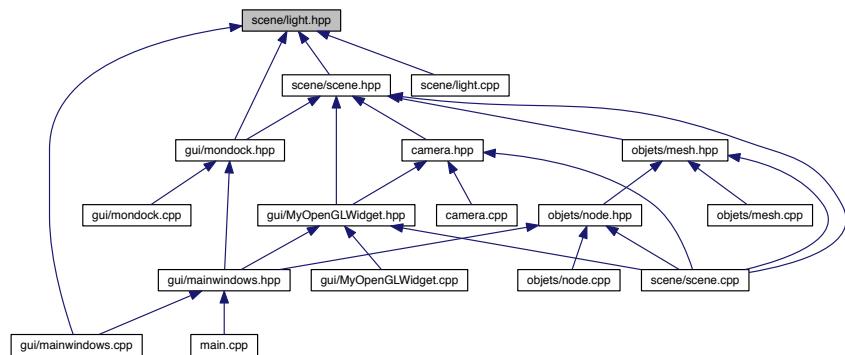
6.27 Référence du fichier scene/light.hpp

```
#include <QOpenGLFunctions_3_2_Core>
#include "helper.hpp"
```

Graphe des dépendances par inclusion de light.hpp :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



Classes

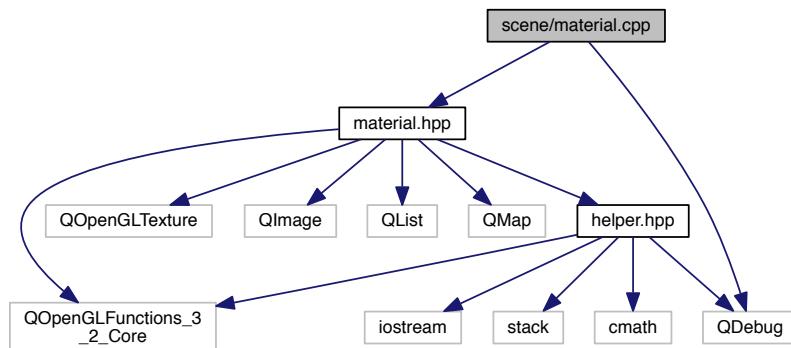
- class [Light](#)

Lumière pour la Scene.

6.28 Référence du fichier scene/material.cpp

```
#include "material.hpp"
#include <QDebug>
```

Graphe des dépendances par inclusion de material.cpp :



Fonctions

- [QDebug operator<< \(QDebug dbg, const Material &m\)](#)

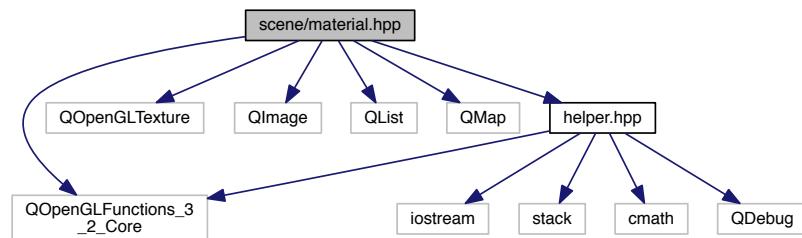
6.28.1 Documentation des fonctions

6.28.1.1 [QDebug operator<< \(qDebug dbg, const Material & m \)](#)

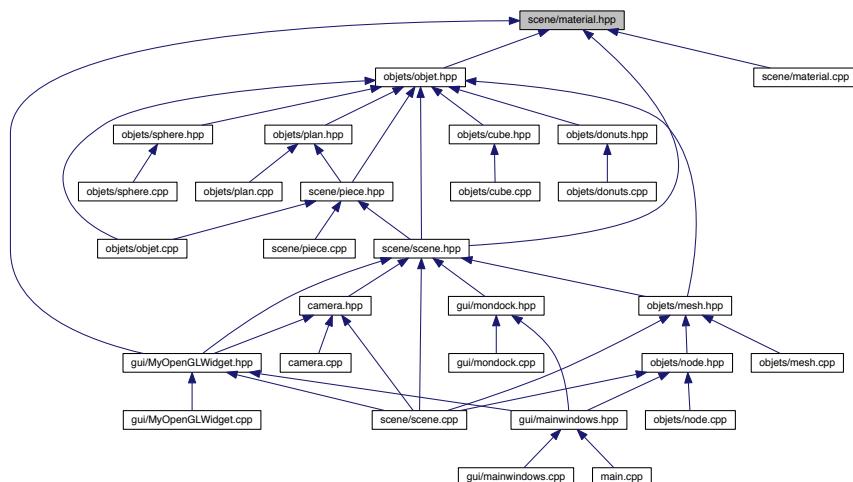
6.29 Référence du fichier scene/material.hpp

```
#include <QOpenGLFunctions_3_2_Core>
#include <QOpenGLTexture>
#include <QImage>
#include <QList>
#include <QMap>
#include "helper.hpp"
#include "material.hpp"
```

Graphe des dépendances par inclusion de material.hpp :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



Classes

- class [Material](#)
Matériaux pour la Scene.

Fonctions

- `QDebug operator<< (QDebug dbg, const Material &m)`

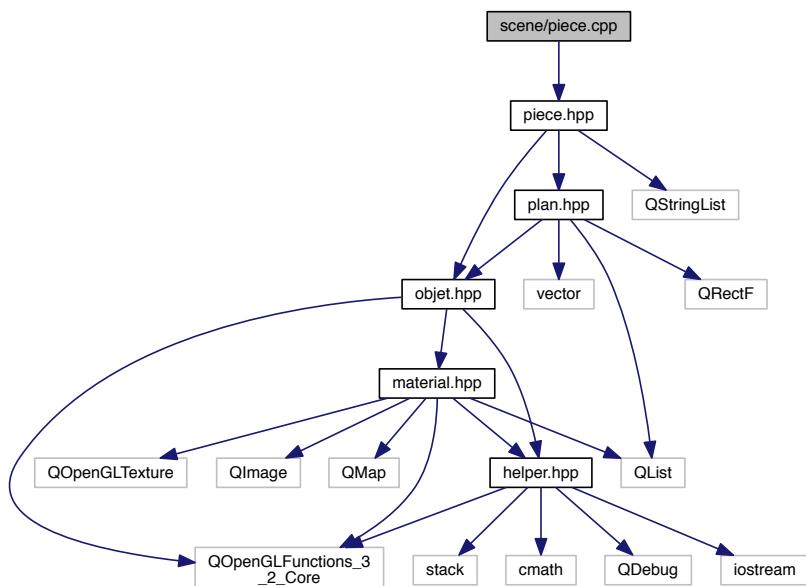
6.29.1 Documentation des fonctions

6.29.1.1 QDebug operator<< (QDebug *dbg*, const Material & *m*)

6.30 Référence du fichier scene/piece.cpp

```
#include "piece.hpp"
```

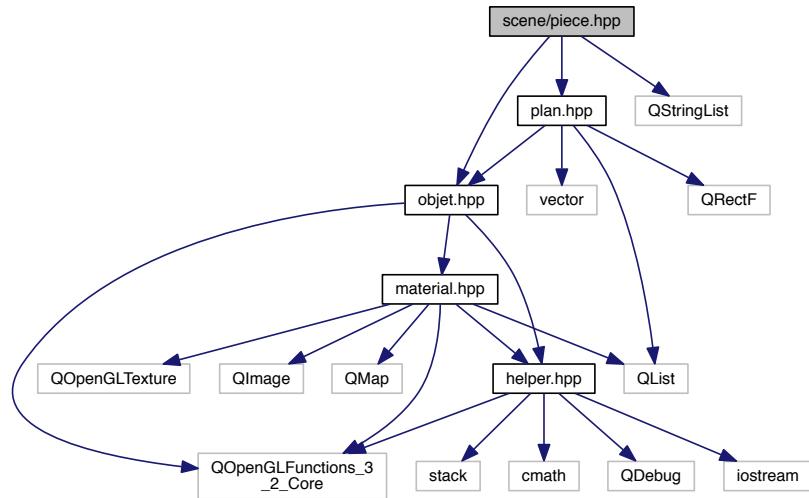
Graphe des dépendances par inclusion de piece.cpp :



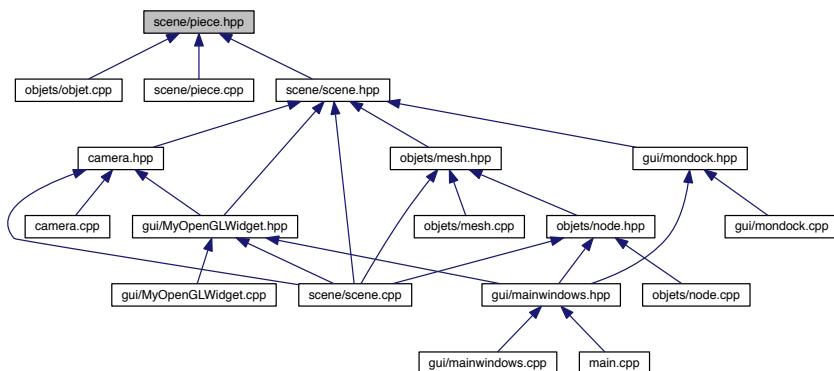
6.31 Référence du fichier scene/piece.hpp

```
#include "objet.hpp"
#include "plan.hpp"
#include <QStringList>
```

Graphe des dépendances par inclusion de piece.hpp :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



Classes

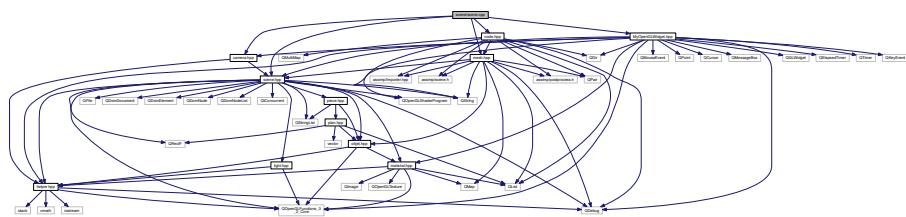
- class [Piece](#)

Pièce, contient un ensemble d'objet qu'elle contient.

6.32 Référence du fichier `scene/scene.cpp`

```
#include "scene.hpp"
#include "mesh.hpp"
#include "node.hpp"
#include "camera.hpp"
#include "MyOpenGLWidget.hpp"
```

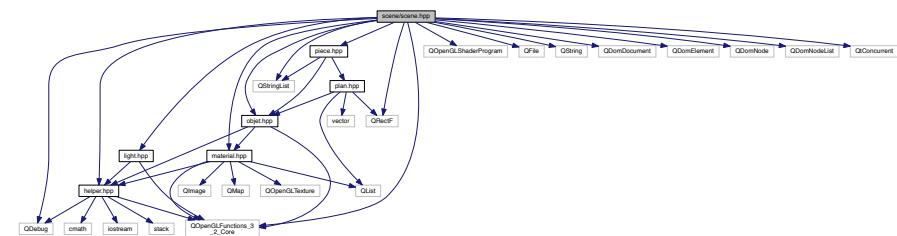
Graphe des dépendances par inclusion de scene.cpp :



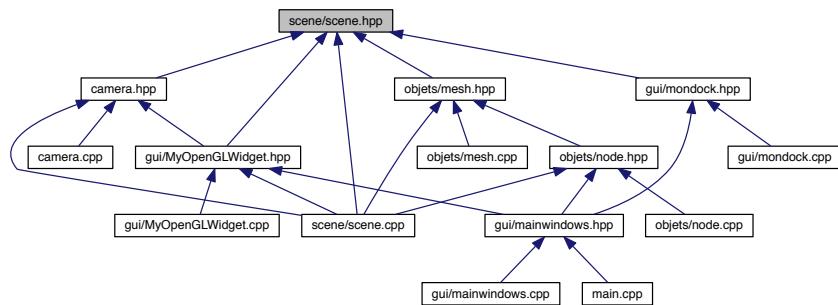
6.33 Référence du fichier scene/scene.hpp

```
#include "helper.hpp"
#include "light.hpp"
#include "material.hpp"
#include "objet.hpp"
#include "piece.hpp"
#include <QOpenGLFunctions_3_2_Core>
#include <QOpenGLShaderProgram>
#include <QDebug>
#include <QFile>
#include <QString>
#include <QStringList>
#include <QDomDocument>
#include <QDomElement>
#include <QDOMNode>
#include <QDOMNodeList>
#include <QRectF>
#include <QtConcurrent>
```

Graphe des dépendances par inclusion de scene.hpp :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



Classes

- class [Scene](#)
Classe principale.

Macros

- #define [MAX_LIGHT](#) 8

6.33.1 Documentation des macros

6.33.1.1 #define MAX_LIGHT 8

Index

ambient
 Light, 27, 28
amblightfuncx
 Mondock, 84
amblightfuncy
 Mondock, 85
amblightfuncz
 Mondock, 85
ambmaterialfuncx
 Mondock, 86
ambmaterialfuncy
 Mondock, 86
ambmaterialfuncz
 Mondock, 87

boxobjerotationx
 Mondock, 114
boxobjerotationy
 Mondock, 114
boxobjerotationz
 Mondock, 114
boxobjetscalex
 Mondock, 114
boxobjetscaley
 Mondock, 114
boxobjetscalez
 Mondock, 114
boxobjettransx
 Mondock, 114
boxobjettransy
 Mondock, 114
boxobjettransz
 Mondock, 114

Camera, 9
 Camera, 11
 display, 12
 go, 12
 move, 13
 position, 13
clear
 Material, 67
combomaterial
 Mondock, 114
comboparent
 Mondock, 115
Cube, 15
 Cube, 18
 draw, 18

det
 mat4, 60
diflightfuncx
 Mondock, 87
diflightfuncy
 Mondock, 88
diflightfuncz
 Mondock, 88
difmaterialfuncx
 Mondock, 89
difmaterialfuncy
 Mondock, 89
difmaterialfuncz
 Mondock, 90
dimensions
 Piece, 155, 156
dimentionpiecex
 Mondock, 115
dimentionpiecey
 Mondock, 115
dimentionpiecez
 Mondock, 115
disable
 Light, 28
display
 Camera, 12
dockmodele
 Mondock, 115
dockscene
 Mondock, 115
dockvue
 Mondock, 115
Donuts, 19
 Donuts, 22
 draw, 22
 nbvertex, 23
draw
 Cube, 18
 Donuts, 22
 Mesh, 77
 Node, 130
 Objet, 142
 Piece, 156
 Plan, 160
 Scene, 168
 Sphere, 186

editionambiente
 Mondock, 115

get
 Light, 28, 29
 Material, 68

go
 Camera, 12

indexgparent
 Mondock, 115

inverse
 mat4, 61

labelamlayout
 Mondock, 115

labelambx
 Mondock, 115

labelamby
 Mondock, 115

labelambz
 Mondock, 115

labeldiflayout
 Mondock, 115

labeldifx
 Mondock, 115

labeldify
 Mondock, 115

labeldifz
 Mondock, 115

labelobjetproprmaterial
 Mondock, 115

labelobjetproprparent
 Mondock, 115

labelsliderobjetscale
 Mondock, 115

labelspelement
 Mondock, 115

labelspelex
 Mondock, 116

labelspey
 Mondock, 116

labelspez
 Mondock, 116

layouteditionlighamb
 Mondock, 116

layoutglobalscaleobjet
 Mondock, 116

layoutobjetscaleall
 Mondock, 116

layouttabobjetpropr
 Mondock, 116

layouttabobjetproprcombo
 Mondock, 116

layouttabobjetproprlabel
 Mondock, 116

layouttabobjetrotation
 Mondock, 116

layouttabobjetscale
 Mondock, 116

layouttabobjetrans
 Mondock, 116

length
 vec2, 189
 vec3, 193
 vec4, 197

Light, 23
 ambient, 27, 28
 disable, 28
 get, 28, 29
 Light, 26
 set, 29
 update, 30

lightamlayout
 Mondock, 116

lightdiflayout
 Mondock, 116

lightdoublespinboxx
 Mondock, 116

lightdoublespinboxy
 Mondock, 116

lightdoublespinboxz
 Mondock, 116

lightselectioner
 Mondock, 116

lightspelayout
 Mondock, 116

lumiereambiantejustx
 Mondock, 90

lumiereambiantejusty
 Mondock, 91

lumiereambiantejustz
 Mondock, 91

m
 mat3, 59
 mat4, 63

mat3, 56
 m, 59
 mat3, 57
 operator*, 57, 58
 operator*=
 operator+
 operator+=
 operator-
 operator-=
 operator=
 mat4, 59
 det, 60
 inverse, 61
 m, 63
 mat4, 60
 operator*, 61
 operator*=
 operator+
 operator+=
 operator-
 operator-=
 operator=
 transpose, 62

Material, 63

clear, 67
get, 68
Material, 66
operator<<, 72
set, 70, 71
shininess, 71
update, 71
material
 Objet, 143, 144
materialselectioner
 Mondock, 116
materialsinboxambx
 Mondock, 116
materialsinboxamby
 Mondock, 116
materialsinboxambz
 Mondock, 116
materialsinboxdifax
 Mondock, 116
materialsinboxdify
 Mondock, 116
materialsinboxdifz
 Mondock, 116
materialsinboxspea
 Mondock, 116
materialsinboxspex
 Mondock, 116
materialsinboxspey
 Mondock, 117
materialsinboxspez
 Mondock, 117
matobjet
 Mondock, 92
Mesh, 73
 draw, 77
 Mesh, 76
 Node, 78
modelmaterial
 Mondock, 117
modelpiece
 Mondock, 117
Mondock, 79
 amblightfuncx, 84
 amblightfuncy, 85
 amblightfuncz, 85
 ambmaterialfuncx, 86
 ambmaterialfuncy, 86
 ambmaterialfuncz, 87
 boxobjetrotationx, 114
 boxobjetrotationy, 114
 boxobjetrotationz, 114
 boxobjetscalex, 114
 boxobjetscaley, 114
 boxobjetscalez, 114
 boxobjettransx, 114
 boxobjettransy, 114
 boxobjettransz, 114
 combomaterial, 114
 comboparent, 115
 diflightfuncx, 87
 diflightfuncy, 88
 diflightfuncz, 88
 difmaterialfuncx, 89
 difmaterialfuncy, 89
 difmaterialfuncz, 90
 dimentionpiecex, 115
 dimentionpiecey, 115
 dimentionpiecez, 115
 dockmodele, 115
 dockscene, 115
 dockvue, 115
 editionambiante, 115
 indexgparent, 115
 labelamblayout, 115
 labelambx, 115
 labelamby, 115
 labelambz, 115
 labeldiflayout, 115
 labeldifx, 115
 labeldify, 115
 labeldifz, 115
 labelobjetproprmaterial, 115
 labelobjetproprparent, 115
 labelsliderobjetscale, 115
 labelspeLayout, 115
 labelspex, 116
 labelspey, 116
 labelspez, 116
 layouteditionlighamb, 116
 layoutglobalscaleobjet, 116
 layoutobjetscaleall, 116
 layouttabobjetpropr, 116
 layouttabobjetproprcombo, 116
 layouttabobjetproprlabel, 116
 layouttabobjettrotation, 116
 layouttabobjetscale, 116
 layouttabobjettrans, 116
 lightamblayout, 116
 lightdiflayout, 116
 lightdoublespinboxx, 116
 lightdoublespinboxy, 116
 lightdoublespinboxz, 116
 lightselectioner, 116
 lightspeLayout, 116
 lumiereambiantejustx, 90
 lumiereambiantejusty, 91
 lumiereambiantejustz, 91
 materialselectioner, 116
 materialsinboxambx, 116
 materialsinboxamby, 116
 materialsinboxambz, 116
 materialsinboxdifax, 116
 materialsinboxdify, 116
 materialsinboxdifz, 116
 materialsinboxspea, 116
 materialsinboxspex, 116

materialspinboxspey, 117
 materialspinboxspez, 117
 matobjet, 92
 modelmaterial, 117
 modelpiece, 117
 Mondock, 84
 objetselectioner, 117
 pereobjet, 92
 piecedimlayout, 117
 pieceposilayout, 117
 piecerotatelayout, 117
 piecescalelayout, 117
 pieceselectioner, 117
 positionpiecex, 117
 positionpiecey, 117
 positionpiecez, 117
 poslightfuncx, 93
 poslightfuncy, 93
 poslightfuncz, 94
 rotobjectx, 94
 rotobjecty, 95
 rotobjectz, 95
 rotpiecex, 96
 rotpiecey, 96
 rotpiecez, 97
 rotspinboxx, 117
 rotspinboxy, 117
 rotspinboxz, 117
 scalelabel, 117
 scaleobjectx, 97
 scaleobjecty, 98
 scaleobjectz, 98
 scalepiecex, 99
 scalepiecey, 99
 scalepiecez, 100
 scaleslider, 117
 scalesliderlayout, 117
 scalesliderobject, 100
 scalesliderpiece, 101
 scalespinboxlayout, 117
 scalespinboxx, 117
 scalespinboxy, 117
 scalespinboxz, 117
 selection, 117
 selectionlight, 101
 sliderambx, 117
 slideramby, 117
 sliderambz, 117
 sliderdify, 117
 sliderdifz, 118
 sliderobjetscale, 118
 sliderspex, 118
 sliderspey, 118
 sliderspez, 118
 slotpositionpiecex, 102
 slotpositionpiecey, 103
 slotpositionpiecez, 103
 spelightfuncx, 104
 spelightfuncy, 104
 spelightfuncz, 105
 spematerialfuncx, 105
 spematerialfuncy, 106
 spematerialfuncz, 106
 spematerialfunc, 107
 spinboxlumiereambiantex, 118
 spinboxlumiereambiantey, 118
 spinboxlumiereambiantez, 118
 tabamlayout, 118
 tabdiflayout, 118
 tablight, 118
 tablightamb, 118
 tablightdif, 118
 tablightpos, 118
 tablightposlayout, 118
 tablightspe, 118
 tabmaterial, 118
 tabmaterialamb, 118
 tabmaterialamblayout, 118
 tabmaterialdif, 118
 tabmaterialdiflayout, 118
 tabmaterialspe, 118
 tabmaterialspe layout, 118
 tabobjet, 118
 tabobjetprop, 118
 tabobjetrotation, 118
 tabobjetscale, 118
 tabobjetrans, 119
 tabpiece, 119
 tabsplayout, 119
 traitementambiante, 107
 traitementlumiere, 108
 traitementmaterial, 109
 traitementobjet, 110
 traitementpiece, 111
 transobjectx, 112
 transobjecty, 113
 transobjectz, 113
 widgetpiecedim, 119
 widgetpieceposi, 119
 widgetpiecerotate, 119
 widgetpiecescale, 119
 move
 Camera, 13
 name
 Objet, 144
 nbvertex
 Donuts, 23
 Sphere, 187
 Node, 126
 draw, 130
 Mesh, 78
 Node, 129, 130
 normalize
 vec2, 189
 vec3, 193

vec4, 198
Objet, 136
draw, 142
material, 143, 144
name, 144
Objet, 140
parent, 145
position, 145, 146
rotation, 146, 147
scale, 147, 148
objetselectioner
 Mondock, 117
operator<<
 Material, 72
operator*
 mat3, 57, 58
 mat4, 61
 vec2, 190
 vec3, 194
 vec4, 198
operator*=
 mat3, 58
 mat4, 61
 vec2, 190
 vec3, 194
 vec4, 198
operator^
 vec3, 195
operator()
 vec2, 189, 190
 vec3, 194
 vec4, 198
operator+
 mat3, 58
 mat4, 61
 vec2, 190
 vec3, 194
 vec4, 198
operator+=
 mat3, 58
 mat4, 62
 vec2, 190
 vec3, 194
 vec4, 199
operator-
 mat3, 58
 mat4, 62
 vec2, 190
 vec3, 194, 195
 vec4, 199
operator-=
 mat3, 58
 mat4, 62
 vec2, 191
 vec3, 195
 vec4, 199
operator/
 vec2, 191
 vec3, 195
 vec4, 199
operator/=
 vec2, 191
 vec3, 195
 vec4, 200
operator=
 mat3, 58
 mat4, 62
 vec2, 191
 vec3, 195
parent
 Objet, 145
pereobjet
 Mondock, 92
Piece, 151
 dimensions, 155, 156
 draw, 156
 Piece, 154
piecedimlayout
 Mondock, 117
pieceposilayout
 Mondock, 117
pielerotatelayout
 Mondock, 117
piecescalelayout
 Mondock, 117
pieceselectioner
 Mondock, 117
Plan, 157
 draw, 160
 Plan, 160
position
 Camera, 13
 Objet, 145, 146
positionpiecex
 Mondock, 117
positionpiecey
 Mondock, 117
positionpiecez
 Mondock, 117
poslightfuncx
 Mondock, 93
poslightfuncy
 Mondock, 93
poslightfuncz
 Mondock, 94
rotation
 Objet, 146, 147
rotobjectx
 Mondock, 94
rotobjecty
 Mondock, 95
rotobjectz
 Mondock, 95
rotpiecex
 Mondock, 96

rotpiecey
 Mondock, 96
rotpiecez
 Mondock, 97
rotspinboxx
 Mondock, 117
rotspinboxy
 Mondock, 117
rotspinboxz
 Mondock, 117

scale
 Objet, 147, 148
scalelabel
 Mondock, 117
scaleobjectx
 Mondock, 97
scaleobjecty
 Mondock, 98
scaleobjectz
 Mondock, 98
scalepiecex
 Mondock, 99
scalepiecey
 Mondock, 99
scalepiecez
 Mondock, 100
scaleslider
 Mondock, 117
scalesliderlayout
 Mondock, 117
scalesliderobject
 Mondock, 100
scalesliderpiece
 Mondock, 101
scalespinboxlayout
 Mondock, 117
scalespinboxx
 Mondock, 117
scalespinboxy
 Mondock, 117
scalespinboxz
 Mondock, 117
Scene, 161
 draw, 168
 Scene, 165
selection
 Mondock, 117
selectionlight
 Mondock, 101
set
 Light, 29
 Material, 70, 71
shininess
 Material, 71
sliderambx
 Mondock, 117
sliderambz
 Mondock, 117
sliderdfix
 Mondock, 117
sliderdfiy
 Mondock, 118
sliderdfiz
 Mondock, 118
sliderobjetscale
 Mondock, 118
sliderspex
 Mondock, 118
sliderspey
 Mondock, 118
sliderspez
 Mondock, 118
slotpositionpiecex
 Mondock, 102
slotpositionpiecey
 Mondock, 103
slotpositionpiecez
 Mondock, 103
spelightfuncx
 Mondock, 104
spelightfuncy
 Mondock, 104
spelightfuncz
 Mondock, 105
spematerialfuncx
 Mondock, 105
spematerialfuncy
 Mondock, 106
spematerialfuncz
 Mondock, 106
spematerialfunca
 Mondock, 107
Sphere, 183
 draw, 186
 nbvertex, 187
 Sphere, 186
spinboxlumiereambientex
 Mondock, 118
spinboxlumiereambiantey
 Mondock, 118
spinboxlumiereambiantez
 Mondock, 118

tabamblayout
 Mondock, 118
tabdiflayout
 Mondock, 118
tablight
 Mondock, 118
tablightamb
 Mondock, 118
tablightdif
 Mondock, 118
tablightpos
 Mondock, 118

tablightposlayout
 Mondock, 118

tablightspe
 Mondock, 118

tabmaterial
 Mondock, 118

tabmaterialamb
 Mondock, 118

tabmaterialalambLayout
 Mondock, 118

tabmaterialdif
 Mondock, 118

tabmaterialdifLayout
 Mondock, 118

tabmaterialspe
 Mondock, 118

tabmaterialspsLayout
 Mondock, 118

tabobjet
 Mondock, 118

tabobjetprop
 Mondock, 118

tabobjetrotation
 Mondock, 118

tabobjetscale
 Mondock, 118

tabobjettrans
 Mondock, 119

tabpiece
 Mondock, 119

tabspLayout
 Mondock, 119

traitementambiante
 Mondock, 107

traitementlumiere
 Mondock, 108

traitementmaterial
 Mondock, 109

traitementobjet
 Mondock, 110

traitementpiece
 Mondock, 111

transobjectx
 Mondock, 112

transobjecty
 Mondock, 113

transobjectz
 Mondock, 113

transpose
 mat4, 62

update
 Light, 30
 Material, 71

v
 vec2, 191
 vec3, 196
 vec4, 200

vec2, 187
length, 189
normalize, 189
operator*, 190
operator*=, 190
operator(), 189, 190
operator+, 190
operator+=, 190
operator-, 190
operator-=, 191
operator/, 191
operator/=, 191
operator=, 191
v, 191
vec2, 188

vec3, 191
length, 193
normalize, 193
operator*, 194
operator*=, 194
operator[^], 195
operator(), 194
operator+, 194
operator+=, 194
operator-, 194, 195
operator-=, 195
operator/, 195
operator/=, 195
operator=, 195
v, 196
vec3, 192, 193

vec4, 196
length, 197
normalize, 198
operator*, 198
operator*=, 198
operator(), 198
operator+, 198
operator+=, 199
operator-, 199
operator-=, 199
operator/, 199
operator/=, 200
v, 200
vec4, 197

widgetpiecedim
 Mondock, 119

widgetpieceposi
 Mondock, 119

widgetpiecerotate
 Mondock, 119

widgetpiecescale
 Mondock, 119