

PIDGIRL Engine
1.0

Generated by Doxygen 1.8.4

Wed Aug 14 2013 13:21:29

Contents

1	Module Index	1
1.1	Modules	1
2	Namespace Index	3
2.1	Namespace List	3
3	Hierarchical Index	5
3.1	Class Hierarchy	5
4	Class Index	9
4.1	Class List	9
5	File Index	15
5.1	File List	15
6	Module Documentation	23
6.1	AStar	23
6.1.1	Detailed Description	23
6.2	Commands	24
6.2.1	Detailed Description	27
6.3	CommandProcessor	28
6.3.1	Detailed Description	28
6.4	Common	29
6.4.1	Detailed Description	29
6.5	GUI	30
6.5.1	Detailed Description	30
6.6	Model	31
6.6.1	Detailed Description	31
6.7	Game	32
6.7.1	Detailed Description	32
6.8	State	33
6.8.1	Detailed Description	33
6.9	Network	34

6.9.1	Detailed Description	34
6.10	Object	35
6.10.1	Detailed Description	35
6.11	RSA	36
6.11.1	Detailed Description	36
7	Namespace Documentation	37
7.1	ObjectFromID Namespace Reference	37
7.1.1	Function Documentation	37
7.1.1.1	objectFrom	37
7.1.1.2	objectFrom	37
7.1.1.3	objectFrom	37
7.1.1.4	objectFrom	37
7.2	PIDGIRL Namespace Reference	37
7.2.1	Detailed Description	40
7.2.2	Enumeration Type Documentation	40
7.2.2.1	GameType	40
7.2.2.2	MainToolButton	40
7.2.3	Function Documentation	40
7.2.3.1	newSeed	40
7.2.3.2	nextID	40
7.2.3.3	showUsage	40
7.2.3.4	writeLog	40
7.2.3.5	writeLogDone	41
7.2.3.6	writeLogFail	41
7.2.4	Variable Documentation	41
7.2.4.1	done	41
7.2.4.2	EDITOR_VIEW_HEIGHT	41
7.2.4.3	EDITOR_VIEW_WIDTH	41
7.2.4.4	EMAIL_FOLDER_NAME	41
7.2.4.5	EMAIL_SUFFIX	41
7.2.4.6	GAME_INFO_SUFFIX	41
7.2.4.7	IP_ADDRESS_REG_EXP_STRING	41
7.2.4.8	MAP_SUFFIX	42
7.2.4.9	MAX_EMAIL_PLAYER_COUNT	42
7.2.4.10	MAX_FILE_ACCESS_RETRIES	42
7.2.4.11	MAX_GRID_LINE_SIZE	42
7.2.4.12	MAX_HEIGHT	42
7.2.4.13	MAX_HORIZONTAL_SQUARE_COUNT	42
7.2.4.14	MAX_PLAYER_NAME_LENGTH	42

7.2.4.15	MAX_SQUARE_SIZE	42
7.2.4.16	MAX_UNDO_REDO_SIZE	42
7.2.4.17	MAX_VERTICAL_SQUARE_COUNT	42
7.2.4.18	MAX_ZOOM_FACTOR	42
7.2.4.19	MIN_GRID_LINE_SIZE	42
7.2.4.20	MIN_HEIGHT	42
7.2.4.21	MIN_HORIZONTAL_SQUARE_COUNT	43
7.2.4.22	MIN_PLAYER_NAME_LENGTH	43
7.2.4.23	MIN_SQUARE_SIZE	43
7.2.4.24	MIN_VERTICAL_SQUARE_COUNT	43
7.2.4.25	MIN_ZOOM_FACTOR	43
7.2.4.26	NAME_REG_EXP_STRING	43
7.2.4.27	PIDGIRL_VERSION	43
7.2.4.28	RSA_FOLDER_NAME	43
7.2.4.29	RSA_PLAINTEXT_FILE_NAME	43
7.2.4.30	RSA_PRIVATE_KEY_FILE_NAME	43
7.2.4.31	RSA_PUBLIC_KEY_FILE_NAME	43
7.2.4.32	SAVE_FOLDER_NAME	43
7.2.4.33	SAVE_SUFFIX	44
7.2.4.34	SECONDS_UNTIL_RETRYING_FILE_ACCESS	44
7.2.4.35	SEND_DATA_RETRIES	44
7.2.4.36	SERVER_PLAYER_ID	44
7.2.4.37	STANDARD_GRID_LINE_SIZE	44
7.2.4.38	STANDARD_HEIGHT	44
7.2.4.39	STANDARD_HORIZONTAL_SQUARE_COUNT	44
7.2.4.40	STANDARD_SQUARE_SIZE	44
7.2.4.41	STANDARD_VERTICAL_SQUARE_COUNT	44
7.2.4.42	STANDARD_ZOOM_FACTOR	44
7.2.4.43	TOOL_BUTTON_COUNT_IN_ROW	44
7.2.4.44	WORLD_FOLDER_NAME	44
7.2.4.45	WORLD_SUFFIX	45
8	Class Documentation	47
8.1	AddToChosenList Class Reference	47
8.1.1	Detailed Description	48
8.1.2	Constructor & Destructor Documentation	48
8.1.2.1	AddToChosenList	48
8.1.3	Member Function Documentation	48
8.1.3.1	execute	48
8.1.4	Member Data Documentation	48

8.1.4.1	availableFileSystemModel	48
8.1.4.2	chosenListModel	48
8.1.4.3	currentIndex	48
8.2	ApplyWorldPropertiesChanges Class Reference	49
8.2.1	Detailed Description	49
8.3	ApplyMapGeometryChanges Class Reference	49
8.3.1	Detailed Description	50
8.3.2	Constructor & Destructor Documentation	50
8.3.2.1	ApplyMapGeometryChanges	50
8.3.3	Member Function Documentation	50
8.3.3.1	applyGeometryChanges	50
8.3.3.2	execute	50
8.3.4	Member Data Documentation	51
8.3.4.1	horizontalDirection	51
8.3.4.2	horizontalNewValue	51
8.3.4.3	maxHeightDirection	51
8.3.4.4	maxHeightNewValue	51
8.3.4.5	verticalDirection	51
8.3.4.6	verticalNewValue	51
8.4	ApplyWorldPropertiesChanges Class Reference	51
8.4.1	Constructor & Destructor Documentation	52
8.4.1.1	ApplyWorldPropertiesChanges	52
8.4.2	Member Function Documentation	52
8.4.2.1	applyChosenStarterMaps	52
8.4.2.2	execute	52
8.4.3	Member Data Documentation	52
8.4.3.1	chosenListModel	52
8.5	AskForSave Class Reference	52
8.5.1	Detailed Description	53
8.5.2	Constructor & Destructor Documentation	53
8.5.2.1	AskForSave	53
8.5.3	Member Function Documentation	53
8.5.3.1	execute	53
8.6	AskForSaveMap Class Reference	54
8.6.1	Constructor & Destructor Documentation	54
8.6.1.1	AskForSaveMap	54
8.6.2	Member Function Documentation	54
8.6.2.1	execute	54
8.6.3	Member Data Documentation	55
8.6.3.1	yncDialog	55

8.7	AskForSaveWorld Class Reference	55
8.7.1	Detailed Description	55
8.7.2	Constructor & Destructor Documentation	56
8.7.2.1	AskForSaveWorld	56
8.7.3	Member Function Documentation	56
8.7.3.1	execute	56
8.7.4	Member Data Documentation	56
8.7.4.1	yncDialog	56
8.8	AStarMap Class Reference	56
8.8.1	Detailed Description	57
8.8.2	Constructor & Destructor Documentation	57
8.8.2.1	AStarMap	57
8.8.2.2	~AStarMap	57
8.8.3	Member Function Documentation	57
8.8.3.1	calculatePath	57
8.8.3.2	fromMapModel	57
8.8.3.3	getMapHeight	58
8.8.3.4	getMapWidth	58
8.8.3.5	getPath	58
8.8.3.6	getPossibleNeighborsOf	58
8.8.3.7	refEntry	58
8.8.3.8	refEntry	58
8.8.3.9	retracePath	58
8.8.4	Member Data Documentation	58
8.8.4.1	map	58
8.8.4.2	path	58
8.9	AStarMapEntry Class Reference	58
8.9.1	Constructor & Destructor Documentation	60
8.9.1.1	AStarMapEntry	60
8.9.1.2	AStarMapEntry	60
8.9.1.3	AStarMapEntry	60
8.9.1.4	AStarMapEntry	60
8.9.2	Member Function Documentation	60
8.9.2.1	calculateF	60
8.9.2.2	copy	60
8.9.2.3	fromStackModel	60
8.9.2.4	getF	60
8.9.2.5	getG	60
8.9.2.6	getPosition	60
8.9.2.7	getWeight	60

8.9.2.8	hasInfinityWeight	60
8.9.2.9	heuristicDistanceTo	60
8.9.2.10	init	61
8.9.2.11	operator!=	62
8.9.2.12	operator=	62
8.9.2.13	operator==	62
8.9.2.14	refCameFrom	62
8.9.2.15	setCameFrom	62
8.9.2.16	setF	62
8.9.2.17	setG	62
8.9.2.18	setWeight	62
8.9.2.19	setWeightToInfinity	62
8.9.3	Member Data Documentation	62
8.9.3.1	cameFrom	62
8.9.3.2	f	62
8.9.3.3	g	62
8.9.3.4	position	62
8.9.3.5	weight	62
8.10	BaseCommand Class Reference	63
8.10.1	Detailed Description	64
8.10.2	Constructor & Destructor Documentation	64
8.10.2.1	BaseCommand	64
8.10.3	Member Function Documentation	64
8.10.3.1	execute	64
8.10.3.2	getCommandIcon	65
8.10.3.3	isClearingUndoRedo	65
8.10.3.4	isErrorEnabled	65
8.10.3.5	isExecutable	65
8.10.3.6	isUndoable	65
8.10.3.7	redo	65
8.10.3.8	setErrorEnabled	65
8.10.3.9	setModels	65
8.10.3.10	undo	66
8.10.4	Member Data Documentation	66
8.10.4.1	errorEnabled	66
8.11	BrickWall Class Reference	66
8.11.1	Detailed Description	67
8.11.2	Constructor & Destructor Documentation	67
8.11.2.1	BrickWall	67
8.11.2.2	BrickWall	67

8.11.3	Member Function Documentation	67
8.11.3.1	copy	67
8.11.3.2	createInformationWidget	67
8.11.3.3	deserialize	67
8.11.3.4	deserialize	67
8.11.3.5	getCommandList	67
8.11.3.6	getObjectID	68
8.11.3.7	serialize	68
8.11.3.8	serialize	68
8.11.3.9	update	68
8.12	BroadcastCommand Class Reference	68
8.12.1	Detailed Description	69
8.12.2	Constructor & Destructor Documentation	69
8.12.2.1	BroadcastCommand	69
8.12.3	Member Function Documentation	69
8.12.3.1	execute	69
8.12.4	Member Data Documentation	69
8.12.4.1	broadcastPlayerIDs	69
8.13	BroadcastSpawnCharacter Class Reference	70
8.13.1	Detailed Description	70
8.13.2	Constructor & Destructor Documentation	71
8.13.2.1	BroadcastSpawnCharacter	71
8.13.3	Member Function Documentation	71
8.13.3.1	execute	71
8.13.4	Member Data Documentation	71
8.13.4.1	newCharacter	71
8.14	BroadcastUpdateCharacter Class Reference	71
8.14.1	Detailed Description	72
8.14.2	Constructor & Destructor Documentation	73
8.14.2.1	BroadcastUpdateCharacter	73
8.14.3	Member Function Documentation	73
8.14.3.1	execute	73
8.14.4	Member Data Documentation	73
8.14.4.1	newCharacter	73
8.14.4.2	oldCharacter	73
8.15	ChangePlayerIDOnServer Class Reference	73
8.15.1	Detailed Description	74
8.15.2	Constructor & Destructor Documentation	74
8.15.2.1	ChangePlayerIDOnServer	74
8.15.3	Member Function Documentation	74

8.15.3.1	execute	74
8.15.4	Member Data Documentation	75
8.15.4.1	newID	75
8.16	ChangeZoomFactor Class Reference	75
8.16.1	Detailed Description	75
8.16.2	Constructor & Destructor Documentation	76
8.16.2.1	ChangeZoomFactor	76
8.16.3	Member Function Documentation	76
8.16.3.1	execute	76
8.16.4	Member Data Documentation	76
8.16.4.1	newFactor	76
8.17	ChooseMainTool Class Reference	76
8.17.1	Detailed Description	77
8.17.2	Constructor & Destructor Documentation	77
8.17.2.1	ChooseMainTool	77
8.17.3	Member Function Documentation	77
8.17.3.1	execute	77
8.17.4	Member Data Documentation	77
8.17.4.1	button	77
8.18	ClearPath Class Reference	78
8.18.1	Detailed Description	78
8.18.2	Constructor & Destructor Documentation	78
8.18.2.1	ClearPath	78
8.18.3	Member Function Documentation	78
8.18.3.1	execute	78
8.19	ClientCommand Class Reference	79
8.19.1	Detailed Description	80
8.19.2	Constructor & Destructor Documentation	80
8.19.2.1	ClientCommand	80
8.19.3	Member Function Documentation	80
8.19.3.1	execute	80
8.19.3.2	isExecutable	80
8.20	ClientDisconnected Class Reference	80
8.20.1	Detailed Description	81
8.20.2	Constructor & Destructor Documentation	81
8.20.2.1	ClientDisconnected	81
8.20.3	Member Function Documentation	81
8.20.3.1	execute	81
8.21	ClientGameModel Class Reference	82
8.21.1	Detailed Description	83

8.21.2	Constructor & Destructor Documentation	83
8.21.2.1	ClientGameModel	83
8.21.3	Member Function Documentation	83
8.21.3.1	addNewPlayer	83
8.21.3.2	addtoPreloadedMaps	84
8.21.3.3	beginTurn	84
8.21.3.4	changePlayerID	84
8.21.3.5	convertFromDifferentGameModel	84
8.21.3.6	deavtivatePlayer	84
8.21.3.7	deserialize	85
8.21.3.8	endTurn	85
8.21.3.9	firstTurn	85
8.21.3.10	getGameType	85
8.21.3.11	getPlayerModels	85
8.21.3.12	getPreloadedMaps	85
8.21.3.13	isYourTurn	85
8.21.3.14	loadMap	85
8.21.3.15	loadMap	85
8.21.3.16	nextTurn	86
8.21.3.17	refCurrentMap	86
8.21.3.18	refPlayer	86
8.21.3.19	refPlayerModel	86
8.21.3.20	refPreloadedMap	86
8.21.3.21	serialize	86
8.21.3.22	setCurrentMap	86
8.21.3.23	setYourTurn	86
8.21.3.24	spawnCharacter	86
8.21.3.25	updateCharacter	87
8.21.4	Member Data Documentation	87
8.21.4.1	currentMap	87
8.21.4.2	processor	87
8.21.4.3	yourTurn	87
8.22	ClientModel Class Reference	87
8.22.1	Detailed Description	89
8.22.2	Constructor & Destructor Documentation	89
8.22.2.1	ClientModel	89
8.22.2.2	~ClientModel	89
8.22.3	Member Function Documentation	89
8.22.3.1	changeClientID	89
8.22.3.2	connectTo	89

8.22.3.3	disconnectedFromServer	89
8.22.3.4	getConnectedPlayerIDs	89
8.22.3.5	getConnectionList	90
8.22.3.6	isClient	90
8.22.3.7	isServer	90
8.22.3.8	receiveData	90
8.22.3.9	receiveData	90
8.22.3.10	receiveData	90
8.22.3.11	receiveData	90
8.22.3.12	receiveData	90
8.22.3.13	receivePendingData	90
8.22.3.14	sendData	90
8.22.3.15	sendData	91
8.22.3.16	sendData	91
8.22.3.17	sendData	91
8.22.3.18	sendData	91
8.22.3.19	socketError	91
8.22.4	Member Data Documentation	91
8.22.4.1	connection	91
8.22.4.2	ipAddress	91
8.22.4.3	processor	91
8.23	ClientSocket Class Reference	91
8.23.1	Constructor & Destructor Documentation	92
8.23.1.1	ClientSocket	92
8.23.1.2	~ClientSocket	92
8.23.2	Member Function Documentation	92
8.23.2.1	disconnected	92
8.23.2.2	getPlayerID	92
8.23.2.3	gotDisconnected	92
8.23.2.4	gotReadyRead	92
8.23.2.5	readyRead	92
8.23.2.6	refSocket	92
8.23.2.7	setPlayerID	92
8.23.3	Member Data Documentation	92
8.23.3.1	playerID	92
8.23.3.2	socket	92
8.24	Command Class Reference	93
8.24.1	Detailed Description	94
8.24.2	Constructor & Destructor Documentation	94
8.24.2.1	Command	94

8.24.3	Member Function Documentation	94
8.24.3.1	execute	94
8.24.3.2	isClearingUndoRedo	95
8.24.3.3	isExecutable	95
8.24.3.4	isUndoable	95
8.24.3.5	redo	95
8.24.3.6	setModels	96
8.24.3.7	undo	96
8.24.4	Member Data Documentation	96
8.24.4.1	editorModel	96
8.24.4.2	gameModel	96
8.24.4.3	mapModel	96
8.24.4.4	networkModel	96
8.24.4.5	processor	96
8.24.4.6	undoRedo	96
8.24.4.7	worldModel	96
8.25	CommandAction Class Reference	96
8.25.1	Detailed Description	97
8.25.2	Constructor & Destructor Documentation	97
8.25.2.1	CommandAction	97
8.25.3	Member Function Documentation	97
8.25.3.1	commandActionTriggered	97
8.25.3.2	triggerWithName	97
8.26	CommonCommand Class Reference	97
8.26.1	Detailed Description	98
8.26.2	Constructor & Destructor Documentation	98
8.26.2.1	CommonCommand	98
8.26.3	Member Function Documentation	98
8.26.3.1	execute	98
8.26.3.2	isExecutable	99
8.27	ConnectTo Class Reference	99
8.27.1	Detailed Description	100
8.27.2	Constructor & Destructor Documentation	100
8.27.2.1	ConnectTo	100
8.27.3	Member Function Documentation	100
8.27.3.1	connectToServer	100
8.27.3.2	execute	100
8.27.3.3	initToConnectDialog	100
8.27.4	Member Data Documentation	101
8.27.4.1	clientModel	101

8.27.4.2	connectToDialog	101
8.27.4.3	connectToLayout	101
8.27.4.4	IPAddressEdit	101
8.27.4.5	IPAddressLabel	101
8.27.4.6	IPAddressValidator	101
8.27.4.7	portEdit	101
8.27.4.8	portLabel	101
8.27.4.9	portValidator	101
8.27.4.10	startNetworkButton	101
8.28	ContextEditorMenuState Class Reference	101
8.28.1	Detailed Description	101
8.28.2	Constructor & Destructor Documentation	102
8.28.2.1	ContextEditorMenuState	102
8.28.3	Member Function Documentation	102
8.28.3.1	nextState	102
8.28.3.2	showContextMenu	102
8.29	ContextGameMenuState Class Reference	102
8.29.1	Detailed Description	103
8.29.2	Constructor & Destructor Documentation	103
8.29.2.1	ContextGameMenuState	103
8.29.3	Member Function Documentation	103
8.29.3.1	nextState	103
8.29.3.2	showContextMenu	103
8.30	CopyFolder Class Reference	104
8.30.1	Detailed Description	104
8.30.2	Constructor & Destructor Documentation	105
8.30.2.1	CopyFolder	105
8.30.3	Member Function Documentation	105
8.30.3.1	copy	105
8.30.3.2	execute	105
8.30.4	Member Data Documentation	105
8.30.4.1	from	105
8.30.4.2	to	105
8.31	CopyWall Class Reference	105
8.31.1	Detailed Description	106
8.31.2	Constructor & Destructor Documentation	106
8.31.2.1	CopyWall	106
8.31.3	Member Function Documentation	106
8.31.3.1	execute	106
8.31.4	Member Data Documentation	107

8.31.4.1	wall	107
8.32	CreateCharacter Class Reference	107
8.32.1	Detailed Description	108
8.32.2	Constructor & Destructor Documentation	108
8.32.2.1	CreateCharacter	108
8.32.2.2	~CreateCharacter	108
8.32.3	Member Function Documentation	108
8.32.3.1	characterDone	108
8.32.3.2	connectionsChanged	109
8.32.3.3	deleteConnectionListItems	109
8.32.3.4	execute	109
8.32.3.5	init	109
8.32.4	Member Data Documentation	109
8.32.4.1	characterCreationDialog	109
8.32.4.2	characterCreationLayout	109
8.32.4.3	chooseListItems	109
8.32.4.4	chooseListModel	109
8.32.4.5	chooseListView	109
8.32.4.6	chooseStartMap	109
8.32.4.7	chooseStartMapLayout	109
8.32.4.8	connectionLabel	109
8.32.4.9	connectionListItems	109
8.32.4.10	connectionListModel	109
8.32.4.11	connectionListView	109
8.32.4.12	informationWidget	109
8.32.4.13	informationWidgetLayout	109
8.32.4.14	playerNameEdit	109
8.32.4.15	playerNameValidator	110
8.32.4.16	serverOverview	110
8.32.4.17	serverOverviewLayout	110
8.33	CutWall Class Reference	110
8.33.1	Detailed Description	110
8.33.2	Constructor & Destructor Documentation	111
8.33.2.1	CutWall	111
8.33.3	Member Function Documentation	111
8.33.3.1	execute	111
8.33.4	Member Data Documentation	111
8.33.4.1	stackModel	111
8.34	DuplicateWorldEntry Class Reference	111
8.34.1	Detailed Description	112

8.34.2	Constructor & Destructor Documentation	112
8.34.2.1	DuplicateWorldEntry	112
8.34.3	Member Function Documentation	112
8.34.3.1	execute	112
8.34.4	Member Data Documentation	112
8.34.4.1	index	112
8.35	EditorMenu Class Reference	113
8.35.1	Detailed Description	114
8.35.2	Constructor & Destructor Documentation	114
8.35.2.1	EditorMenu	114
8.35.3	Member Function Documentation	114
8.35.3.1	about	114
8.35.3.2	createNewWorld	114
8.35.3.3	loadWorld	115
8.35.3.4	quit	115
8.35.3.5	saveMap	115
8.35.3.6	saveMapAs	115
8.35.3.7	saveWorld	115
8.35.3.8	saveWorldAs	115
8.35.3.9	showMapProperties	115
8.35.3.10	showWorldProperties	115
8.35.3.11	updateSize	115
8.35.3.12	useRedo	115
8.35.3.13	useUndo	115
8.35.3.14	zoom	115
8.35.4	Member Data Documentation	116
8.35.4.1	aboutAction	116
8.35.4.2	editMenu	116
8.35.4.3	editorModel	116
8.35.4.4	fileMenu	116
8.35.4.5	helpMenu	116
8.35.4.6	loadWorldAction	116
8.35.4.7	mapPropertiesAction	116
8.35.4.8	mapPropertiesDialog	116
8.35.4.9	newWorldAction	116
8.35.4.10	processor	116
8.35.4.11	quitAction	116
8.35.4.12	redoAction	116
8.35.4.13	saveMapAction	116
8.35.4.14	saveMapAsAction	116

8.35.4.15	saveWorldAction	116
8.35.4.16	saveWorldAsAction	116
8.35.4.17	undoAction	116
8.35.4.18	viewMenu	116
8.35.4.19	worldModel	116
8.35.4.20	worldPropertiesAction	116
8.35.4.21	worldPropertiesDialog	116
8.35.4.22	zoomActionGroup	116
8.35.4.23	zoomMenu	116
8.36	EditorModel Class Reference	117
8.36.1	Detailed Description	117
8.36.2	Constructor & Destructor Documentation	117
8.36.2.1	EditorModel	117
8.36.3	Member Function Documentation	117
8.36.3.1	getIsEditor	117
8.36.3.2	refMainObject	117
8.36.3.3	setMainToolObject	117
8.36.4	Member Data Documentation	118
8.36.4.1	isEditor	118
8.36.4.2	mainToolObject	118
8.37	EditorTools Class Reference	118
8.37.1	Detailed Description	119
8.37.2	Constructor & Destructor Documentation	119
8.37.2.1	EditorTools	119
8.37.3	Member Function Documentation	119
8.37.3.1	addButton	119
8.37.3.2	initializeToolButtons	119
8.37.3.3	setMainTool	119
8.37.4	Member Data Documentation	119
8.37.4.1	buttonVector	119
8.37.4.2	editorModel	120
8.37.4.3	gridLayout	120
8.37.4.4	processor	120
8.37.4.5	toolButtons	120
8.38	EMailGameModel Class Reference	120
8.38.1	Detailed Description	121
8.38.2	Constructor & Destructor Documentation	121
8.38.2.1	EMailGameModel	121
8.38.3	Member Function Documentation	121
8.38.3.1	beginTurn	121

8.38.3.2	convertFromDifferentGameModel	121
8.38.3.3	deserialize	121
8.38.3.4	deserialize	122
8.38.3.5	endTurn	122
8.38.3.6	firstTurn	122
8.38.3.7	getGameType	122
8.38.3.8	getTurnCount	122
8.38.3.9	nextPlayerID	122
8.38.3.10	nextTurn	122
8.38.3.11	refMailGameInfo	122
8.38.3.12	serialize	122
8.38.3.13	serialize	122
8.38.3.14	setTurnCount	122
8.38.4	Member Data Documentation	122
8.38.4.1	mailGameInfos	122
8.38.4.2	turnCount	122
8.39	EEmailSetupWidget Class Reference	123
8.39.1	Detailed Description	123
8.39.2	Constructor & Destructor Documentation	123
8.39.2.1	EEmailSetupWidget	123
8.39.3	Member Data Documentation	123
8.39.3.1	emailWidgetLayout	123
8.39.3.2	gameStartButton	123
8.39.3.3	playerCountLabel	123
8.39.3.4	playerCountSpinBox	123
8.40	EndTurn Class Reference	124
8.40.1	Detailed Description	124
8.40.2	Constructor & Destructor Documentation	124
8.40.2.1	EndTurn	124
8.40.3	Member Function Documentation	125
8.40.3.1	execute	125
8.41	FileManager Class Reference	125
8.41.1	Detailed Description	126
8.41.2	Constructor & Destructor Documentation	126
8.41.2.1	FileManager	126
8.41.3	Member Function Documentation	126
8.41.3.1	deserialize	126
8.41.3.2	fileClose	126
8.41.3.3	fileExists	126
8.41.3.4	fileOpen	126

8.41.3.5	fileRemove	127
8.41.3.6	refFile	127
8.41.3.7	serialize	127
8.41.3.8	setFilePath	127
8.41.4	Member Data Documentation	127
8.41.4.1	file	127
8.42	FindPath Class Reference	127
8.42.1	Detailed Description	128
8.42.2	Constructor & Destructor Documentation	128
8.42.2.1	FindPath	128
8.42.3	Member Function Documentation	128
8.42.3.1	execute	129
8.42.4	Member Data Documentation	129
8.42.4.1	aStarMap	129
8.42.4.2	from	129
8.42.4.3	to	129
8.43	FirstTurn Class Reference	129
8.43.1	Detailed Description	130
8.43.2	Constructor & Destructor Documentation	130
8.43.2.1	FirstTurn	130
8.43.3	Member Function Documentation	130
8.43.3.1	done	130
8.43.3.2	execute	130
8.43.3.3	indexChanged	131
8.43.3.4	init	131
8.43.4	Member Data Documentation	131
8.43.4.1	choosableCharacterWidgets	131
8.43.4.2	chooseCharacterDialog	131
8.43.4.3	chooseCharacterLayout	131
8.43.4.4	inactivePlayerChosen	131
8.43.4.5	inactivePlayers	131
8.43.4.6	newCharacterLayout	131
8.43.4.7	newCharacterWidget	131
8.43.4.8	players	131
8.44	GameCommand Class Reference	131
8.44.1	Detailed Description	132
8.44.2	Constructor & Destructor Documentation	133
8.44.2.1	GameCommand	133
8.44.3	Member Function Documentation	133
8.44.3.1	execute	133

8.44.3.2	isExecutable	133
8.44.3.3	quit	133
8.45	GameModel Class Reference	133
8.45.1	Detailed Description	135
8.45.2	Constructor & Destructor Documentation	136
8.45.2.1	GameModel	136
8.45.3	Member Function Documentation	136
8.45.3.1	addNewPlayer	136
8.45.3.2	addtoPreloadedMaps	137
8.45.3.3	beginTurn	137
8.45.3.4	changePlayerID	137
8.45.3.5	convertFromDifferentGameModel	137
8.45.3.6	deavtivatePlayer	137
8.45.3.7	deletePlayerModels	138
8.45.3.8	deserialize	138
8.45.3.9	endTurn	138
8.45.3.10	firstTurn	138
8.45.3.11	getCurrentPlayerID	138
8.45.3.12	getGameType	138
8.45.3.13	getPlayerChoosesStartLocation	138
8.45.3.14	getPlayerID	138
8.45.3.15	getPlayerModels	138
8.45.3.16	getPreloadedMaps	138
8.45.3.17	isYourTurn	138
8.45.3.18	loadMap	139
8.45.3.19	loadMap	140
8.45.3.20	nextTurn	140
8.45.3.21	refCurrentMap	140
8.45.3.22	refPlayer	140
8.45.3.23	refPlayerModel	140
8.45.3.24	refPreloadedMap	140
8.45.3.25	serialize	140
8.45.3.26	setCurrentMap	140
8.45.3.27	setNewMapSignal	141
8.45.3.28	setPlayerChoosesStartLocation	141
8.45.3.29	setPlayerID	141
8.45.3.30	setPlayerModels	141
8.45.3.31	setRequestedData	141
8.45.3.32	setRequestedObject	141
8.45.3.33	spawnCharacter	141

8.45.3.34	updateCharacter	141
8.45.4	Member Data Documentation	141
8.45.4.1	playerChoosesStartLocation	141
8.45.4.2	playerID	142
8.45.4.3	playerModels	142
8.45.4.4	requestedData	142
8.45.4.5	requestedObject	142
8.46	GeneralSetupWidget Class Reference	142
8.46.1	Detailed Description	142
8.46.2	Constructor & Destructor Documentation	143
8.46.2.1	GeneralSetupWidget	143
8.46.3	Member Data Documentation	143
8.46.3.1	generalSetupWidgetLayout	143
8.46.3.2	playerChoosesStartLocation	143
8.47	HotseatGameModel Class Reference	143
8.47.1	Detailed Description	144
8.47.2	Constructor & Destructor Documentation	144
8.47.2.1	HotseatGameModel	144
8.47.3	Member Function Documentation	144
8.47.3.1	beginTurn	144
8.47.3.2	changePlayerID	144
8.47.3.3	convertFromDifferentGameModel	144
8.47.3.4	deserialize	145
8.47.3.5	endTurn	145
8.47.3.6	firstTurn	145
8.47.3.7	getGameType	145
8.47.3.8	nextPlayerID	145
8.47.3.9	nextTurn	145
8.47.3.10	serialize	145
8.48	IdleEditorMouseState Class Reference	145
8.48.1	Detailed Description	146
8.48.2	Constructor & Destructor Documentation	146
8.48.2.1	IdleEditorMouseState	146
8.48.3	Member Function Documentation	146
8.48.3.1	nextState	146
8.49	IdleGameMouseState Class Reference	146
8.49.1	Detailed Description	147
8.49.2	Constructor & Destructor Documentation	147
8.49.2.1	IdleGameMouseState	147
8.49.3	Member Function Documentation	147

8.49.3.1	nextState	147
8.50	List Class Reference	148
8.50.1	Detailed Description	148
8.50.2	Constructor & Destructor Documentation	148
8.50.2.1	List	148
8.50.3	Member Function Documentation	148
8.50.3.1	mouseMoveEvent	148
8.50.3.2	mousePressEvent	148
8.51	LoadGame Class Reference	149
8.51.1	Detailed Description	150
8.51.2	Constructor & Destructor Documentation	150
8.51.2.1	LoadGame	150
8.51.3	Member Function Documentation	150
8.51.3.1	execute	150
8.51.3.2	init	150
8.51.3.3	loadNetworkGame	150
8.51.3.4	loadSoloHotseatGame	150
8.51.4	Member Data Documentation	150
8.51.4.1	fileManager	150
8.51.4.2	gameModelPointer	150
8.51.4.3	loadSetupDialog	150
8.51.4.4	loadSetupLayout	150
8.51.4.5	networkSetupWidget	150
8.51.4.6	saveFilePath	150
8.51.4.7	soloHotseatSetupWidget	151
8.52	LoadMap Class Reference	151
8.52.1	Detailed Description	151
8.52.2	Constructor & Destructor Documentation	152
8.52.2.1	LoadMap	152
8.52.3	Member Function Documentation	152
8.52.3.1	execute	152
8.52.4	Member Data Documentation	152
8.52.4.1	ask	152
8.52.4.2	fileManager	152
8.52.4.3	path	152
8.53	LoadMapByIndex Class Reference	152
8.53.1	Detailed Description	153
8.53.2	Constructor & Destructor Documentation	153
8.53.2.1	LoadMapByIndex	153
8.53.3	Member Function Documentation	153

8.53.3.1	execute	153
8.53.4	Member Data Documentation	154
8.53.4.1	ask	154
8.53.4.2	modelIndex	154
8.54	LoadStartMapFromIndex Class Reference	154
8.54.1	Detailed Description	155
8.54.2	Constructor & Destructor Documentation	155
8.54.2.1	LoadStartMapFromIndex	155
8.54.3	Member Function Documentation	155
8.54.3.1	execute	155
8.54.4	Member Data Documentation	155
8.54.4.1	startMapIndex	155
8.55	LoadWorld Class Reference	155
8.55.1	Detailed Description	156
8.55.2	Constructor & Destructor Documentation	157
8.55.2.1	LoadWorld	157
8.55.3	Member Function Documentation	157
8.55.3.1	execute	157
8.55.3.2	loadWorld	157
8.55.4	Member Data Documentation	157
8.55.4.1	fileManager	157
8.55.4.2	mayAsk	157
8.55.4.3	resetToStandardMap	157
8.56	LocalFilePaths Class Reference	157
8.56.1	Detailed Description	158
8.56.2	Constructor & Destructor Documentation	158
8.56.2.1	LocalFilePaths	158
8.56.2.2	LocalFilePaths	158
8.56.2.3	LocalFilePaths	158
8.56.2.4	LocalFilePaths	158
8.56.3	Member Function Documentation	158
8.56.3.1	convertToFileName	158
8.56.3.2	convertToFileNameVector	158
8.57	LocalGameModel Class Reference	158
8.57.1	Detailed Description	159
8.57.2	Constructor & Destructor Documentation	160
8.57.2.1	LocalGameModel	160
8.57.3	Member Function Documentation	160
8.57.3.1	addNewPlayer	160
8.57.3.2	addtoPreloadedMaps	160

8.57.3.3	convertFromDifferentGameModel	160
8.57.3.4	deavtivatePlayer	160
8.57.3.5	deserialize	160
8.57.3.6	getPreloadedMaps	161
8.57.3.7	isYourTurn	161
8.57.3.8	loadMap	161
8.57.3.9	refCurrentMap	161
8.57.3.10	refPlayer	161
8.57.3.11	refPlayerModel	161
8.57.3.12	refPreloadedMap	161
8.57.3.13	serialize	161
8.57.3.14	setCurrentMap	161
8.57.4	Member Data Documentation	162
8.57.4.1	preloadedMaps	162
8.57.4.2	processor	162
8.58	MailGameInfo Class Reference	162
8.58.1	Detailed Description	163
8.58.2	Constructor & Destructor Documentation	163
8.58.2.1	MailGameInfo	163
8.58.3	Member Function Documentation	163
8.58.3.1	authenticatePlayer	163
8.58.3.2	deserialize	163
8.58.3.3	deserialize	163
8.58.3.4	getEmailGameDir	163
8.58.3.5	getEncryptedText	163
8.58.3.6	getLocalEmailGamePath	163
8.58.3.7	getSaveDir	163
8.58.3.8	getSaveDirAbsolutePath	163
8.58.3.9	hasEmailGameDirSet	163
8.58.3.10	hasSaveDirSet	164
8.58.3.11	initializeMailGameInfo	164
8.58.3.12	serialize	164
8.58.3.13	serialize	164
8.58.3.14	setLocalEmailGamePath	164
8.58.3.15	setSaveDir	164
8.58.4	Member Data Documentation	164
8.58.4.1	cipherText	164
8.58.4.2	emailGameDir	164
8.58.4.3	emailGameDirSet	164
8.58.4.4	saveDir	164

8.58.4.5	saveDirSet	164
8.59	MainViewMap Class Reference	164
8.59.1	Detailed Description	165
8.59.2	Constructor & Destructor Documentation	165
8.59.2.1	MainViewMap	165
8.59.3	Member Function Documentation	165
8.59.3.1	createStackGrid	165
8.59.3.2	deleteStackGrid	166
8.59.3.3	mapSizeUpdated	166
8.59.3.4	paintEvent	166
8.59.3.5	paintGrid	166
8.59.3.6	showContextMenu	166
8.59.3.7	updateMap	166
8.59.4	Member Data Documentation	166
8.59.4.1	editorModel	166
8.59.4.2	gameModel	166
8.59.4.3	mapModel	166
8.59.4.4	mouseInput	166
8.59.4.5	processor	166
8.59.4.6	stackGrid	166
8.60	MainViewStack Class Reference	166
8.60.1	Detailed Description	168
8.60.2	Constructor & Destructor Documentation	168
8.60.2.1	MainViewStack	168
8.60.3	Member Function Documentation	168
8.60.3.1	drawDirectionIndicator	168
8.60.3.2	drawObject	168
8.60.3.3	mouseMoveEvent	168
8.60.3.4	mousePressEvent	168
8.60.3.5	mouseReleaseEvent	168
8.60.3.6	paintEvent	168
8.60.3.7	setStackModel	168
8.60.3.8	updateAll	168
8.60.3.9	updateSize	169
8.60.4	Member Data Documentation	169
8.60.4.1	editorModel	169
8.60.4.2	gameModel	169
8.60.4.3	mapModel	169
8.60.4.4	mouseInput	169
8.60.4.5	processor	169

8.60.4.6	stackModel	169
8.61	MainWindow Class Reference	169
8.61.1	Detailed Description	171
8.61.2	Constructor & Destructor Documentation	171
8.61.2.1	MainWindow	171
8.61.3	Member Function Documentation	171
8.61.3.1	closeEvent	171
8.61.3.2	createMainViewMap	171
8.61.3.3	init	171
8.61.3.4	initializeEditor	171
8.61.3.5	initializeGame	171
8.61.3.6	setNewMap	171
8.61.3.7	updateEditorSize	172
8.61.3.8	updateSize	172
8.61.3.9	updateWindowTitle	172
8.61.3.10	windowSizeChanged	172
8.61.4	Member Data Documentation	172
8.61.4.1	done	172
8.61.4.2	editorMenu	172
8.61.4.3	editorModel	172
8.61.4.4	editorTools	172
8.61.4.5	editorToolsDock	172
8.61.4.6	gameModel	172
8.61.4.7	mainViewMap	172
8.61.4.8	mapModel	172
8.61.4.9	processor	172
8.61.4.10	worldModel	172
8.61.4.11	worldView	172
8.61.4.12	worldViewDock	173
8.62	MapInfos Class Reference	173
8.62.1	Detailed Description	173
8.63	MapModel Class Reference	173
8.63.1	Detailed Description	176
8.63.2	Member Enumeration Documentation	176
8.63.2.1	Side	176
8.63.3	Constructor & Destructor Documentation	176
8.63.3.1	MapModel	176
8.63.3.2	MapModel	176
8.63.4	Member Function Documentation	177
8.63.4.1	addPlayerStack	177

8.63.4.2	addStartPositionStack	177
8.63.4.3	appendHeightLeft	177
8.63.4.4	appendHeightRight	177
8.63.4.5	appendHorizontalLeft	177
8.63.4.6	appendHorizontalRight	177
8.63.4.7	appendVerticalLeft	177
8.63.4.8	appendVerticalRight	178
8.63.4.9	clearPath	178
8.63.4.10	clearStackModelGrid	178
8.63.4.11	clipboardObjectSet	178
8.63.4.12	correctPositionEntries	178
8.63.4.13	createStackModelAt	178
8.63.4.14	deleteStackModelAt	178
8.63.4.15	deserialize	178
8.63.4.16	deserialize	179
8.63.4.17	firstTimeInit	179
8.63.4.18	getCurrentHeight	179
8.63.4.19	getCurrentPath	179
8.63.4.20	getGridLineSize	179
8.63.4.21	getHSquareCount	179
8.63.4.22	getLocalMapPath	179
8.63.4.23	getMapSize	179
8.63.4.24	getMaxHeight	179
8.63.4.25	getSquareSize	179
8.63.4.26	getSquareSizeZoomFactor	179
8.63.4.27	getVSquareCount	179
8.63.4.28	gridChanged	179
8.63.4.29	gridLineSizeChanged	179
8.63.4.30	hasStartPositions	179
8.63.4.31	heightSizeChanged	179
8.63.4.32	horizontalSizeChanged	179
8.63.4.33	initMapModel	179
8.63.4.34	mapPathChanged	179
8.63.4.35	refClipboardObject	179
8.63.4.36	refDestinationStackModel	179
8.63.4.37	refPlayerStack	179
8.63.4.38	refStackModel	179
8.63.4.39	refStackModel	179
8.63.4.40	refStackModel	179
8.63.4.41	refStartPositionStack	179

8.63.4.42	removeHeightLeft	180
8.63.4.43	removeHeightRight	181
8.63.4.44	removeHorizontalLeft	181
8.63.4.45	removeHorizontalRight	181
8.63.4.46	removePlayerStack	181
8.63.4.47	removeStackModelAt	181
8.63.4.48	removeStartPositionStack	181
8.63.4.49	removeVerticalLeft	181
8.63.4.50	removeVerticalRight	182
8.63.4.51	resizeStackModelGrid	182
8.63.4.52	serialize	182
8.63.4.53	serialize	182
8.63.4.54	setClipboardObject	182
8.63.4.55	setConvertedPath	182
8.63.4.56	setCurrentHeight	182
8.63.4.57	setGridLineSize	182
8.63.4.58	setHSquareCount	182
8.63.4.59	setLocalMapPath	182
8.63.4.60	setMaxHeight	183
8.63.4.61	setNewStandardMap	184
8.63.4.62	setSquareSizeZoomFactor	184
8.63.4.63	setVSquareCount	184
8.63.4.64	squareSizeChanged	184
8.63.4.65	updateSquareSize	184
8.63.4.66	verticalSizeChanged	184
8.63.5	Member Data Documentation	184
8.63.5.1	clipboardObject	184
8.63.5.2	currentHeight	184
8.63.5.3	currentPath	184
8.63.5.4	gridLineSize	184
8.63.5.5	localMapPath	184
8.63.5.6	playerStacks	184
8.63.5.7	squareSize	184
8.63.5.8	squareSizeZoomFactor	185
8.63.5.9	stackModelGrid	185
8.63.5.10	startPositionStacks	185
8.64	MapPropertiesDialog Class Reference	185
8.64.1	Constructor & Destructor Documentation	186
8.64.1.1	MapPropertiesDialog	186
8.64.2	Member Function Documentation	186

8.64.2.1	applyChanges	186
8.64.2.2	updateHorizontal	186
8.64.2.3	updateMaxHeight	186
8.64.2.4	updateVertical	186
8.64.3	Member Data Documentation	186
8.64.3.1	geometryPropeties	186
8.64.3.2	geometryPropetiesLayout	186
8.64.3.3	horizontalSquareSideComboBox	186
8.64.3.4	horizontalSquareSpinBox	186
8.64.3.5	maxHeightComboBox	186
8.64.3.6	maxHeightSpinBox	186
8.64.3.7	optionsLayout	186
8.64.3.8	processor	186
8.64.3.9	verticalSquareSideComboBox	186
8.64.3.10	verticalSquareSpinBox	186
8.65	MouseEditorInput Class Reference	186
8.65.1	Detailed Description	189
8.65.2	Constructor & Destructor Documentation	189
8.65.2.1	MouseEditorInput	189
8.65.3	Member Function Documentation	189
8.65.3.1	checkCharacterActions	189
8.65.3.2	checkFloorActions	189
8.65.3.3	checkItemsActions	189
8.65.3.4	checkPasteAction	189
8.65.3.5	checkRemoveAllAction	189
8.65.3.6	checkSpecialsActions	189
8.65.3.7	checkWallActions	190
8.65.3.8	copyCharacter	190
8.65.3.9	copyFloor	190
8.65.3.10	copyItems	190
8.65.3.11	copySpecials	190
8.65.3.12	copyWall	190
8.65.3.13	cutCharacter	190
8.65.3.14	cutFloor	190
8.65.3.15	cutItems	190
8.65.3.16	cutSpecials	190
8.65.3.17	cutWall	190
8.65.3.18	editCharacter	190
8.65.3.19	editFloor	191
8.65.3.20	editItems	191

8.65.3.21	editSpecials	191
8.65.3.22	editWall	191
8.65.3.23	nextState	191
8.65.3.24	pasteObject	191
8.65.3.25	removeAll	191
8.65.3.26	removeCharacter	191
8.65.3.27	removeFloor	191
8.65.3.28	removeItems	191
8.65.3.29	removeSpecials	191
8.65.3.30	removeWall	191
8.65.3.31	showEditorContextMenu	192
8.65.4	Member Data Documentation	192
8.65.4.1	characterMenu	192
8.65.4.2	contextMenuStack	192
8.65.4.3	copyCharacterAction	192
8.65.4.4	copyFloorAction	192
8.65.4.5	copyItemsAction	192
8.65.4.6	copySpecialsAction	192
8.65.4.7	copyWallAction	192
8.65.4.8	cutCharacterAction	192
8.65.4.9	cutFloorAction	192
8.65.4.10	cutItemsAction	192
8.65.4.11	cutSpecialsAction	192
8.65.4.12	cutWallAction	192
8.65.4.13	editCharacterAction	192
8.65.4.14	editFloorAction	192
8.65.4.15	editItemsAction	192
8.65.4.16	editSpecialsAction	192
8.65.4.17	editWallAction	192
8.65.4.18	floorMenu	192
8.65.4.19	itemsMenu	192
8.65.4.20	pasteObjectAction	192
8.65.4.21	removeAllAction	192
8.65.4.22	removeCharacterAction	192
8.65.4.23	removeFloorAction	192
8.65.4.24	removeItemsAction	192
8.65.4.25	removeSpecialsAction	192
8.65.4.26	removeWallAction	192
8.65.4.27	specialsMenu	193
8.65.4.28	wallMenu	193

8.66	MouseGameInput Class Reference	193
8.66.1	Detailed Description	193
8.66.2	Constructor & Destructor Documentation	194
8.66.2.1	MouseGameInput	194
8.66.3	Member Function Documentation	194
8.66.3.1	endTurn	194
8.66.3.2	nextState	194
8.66.3.3	showGameContextMenu	194
8.66.3.4	showGameOptions	194
8.67	MouseInput Class Reference	194
8.67.1	Detailed Description	195
8.67.2	Constructor & Destructor Documentation	195
8.67.2.1	MouseInput	195
8.67.2.2	~MouseInput	195
8.67.3	Member Function Documentation	195
8.67.3.1	deleteContextMenu	195
8.67.3.2	mousePressed	195
8.67.3.3	mouseReleased	195
8.67.3.4	nextState	195
8.67.3.5	showContextMenu	195
8.67.3.6	showContextMenu	195
8.67.4	Member Data Documentation	195
8.67.4.1	contextMenu	195
8.67.4.2	currentMouseState	196
8.67.4.3	gameModel	196
8.67.4.4	mapModel	196
8.67.4.5	processor	196
8.68	MouseState Class Reference	196
8.68.1	Detailed Description	197
8.68.2	Member Enumeration Documentation	197
8.68.2.1	MouseEventType	197
8.68.3	Constructor & Destructor Documentation	197
8.68.3.1	MouseState	197
8.68.3.2	~MouseState	197
8.68.4	Member Function Documentation	197
8.68.4.1	nextState	197
8.68.4.2	refTarget	198
8.68.4.3	setTarget	198
8.68.4.4	showContextMenu	198
8.68.5	Member Data Documentation	198

8.68.5.1	processor	198
8.68.5.2	target	198
8.69	MoveCharacter Class Reference	198
8.69.1	Detailed Description	199
8.69.2	Constructor & Destructor Documentation	199
8.69.2.1	MoveCharacter	199
8.69.3	Member Function Documentation	199
8.69.3.1	execute	199
8.69.4	Member Data Documentation	199
8.69.4.1	from	199
8.69.4.2	to	200
8.70	MoveFile Class Reference	200
8.70.1	Detailed Description	200
8.70.2	Constructor & Destructor Documentation	201
8.70.2.1	MoveFile	201
8.70.3	Member Function Documentation	201
8.70.3.1	execute	201
8.70.4	Member Data Documentation	201
8.70.4.1	fromIndex	201
8.70.4.2	toIndex	201
8.71	NetworkCommand Class Reference	201
8.71.1	Detailed Description	202
8.71.2	Constructor & Destructor Documentation	202
8.71.2.1	NetworkCommand	202
8.71.3	Member Function Documentation	202
8.71.3.1	execute	202
8.71.3.2	isExecutable	203
8.72	NetworkModel Class Reference	203
8.72.1	Detailed Description	204
8.72.2	Constructor & Destructor Documentation	204
8.72.2.1	NetworkModel	204
8.72.2.2	~NetworkModel	204
8.72.3	Member Function Documentation	204
8.72.3.1	allDataReceivedFromClient	204
8.72.3.2	allDataReceivedFromServer	204
8.72.3.3	changeClientID	204
8.72.3.4	clientDisconnect	205
8.72.3.5	connectionEstablished	205
8.72.3.6	getConnectedPlayerIDs	205
8.72.3.7	getConnectionList	205

8.72.3.8	isClient	205
8.72.3.9	isServer	205
8.72.3.10	receiveData	205
8.72.3.11	receiveData	205
8.72.3.12	receiveData	205
8.72.3.13	receiveData	205
8.72.3.14	receiveData	206
8.72.3.15	sendData	206
8.72.3.16	sendData	206
8.72.3.17	sendData	206
8.72.3.18	sendData	206
8.72.3.19	sendData	206
8.72.4	Member Data Documentation	206
8.72.4.1	port	206
8.73	NetworkSetupWidget Class Reference	206
8.73.1	Detailed Description	207
8.73.2	Constructor & Destructor Documentation	207
8.73.2.1	NetworkSetupWidget	207
8.73.3	Member Data Documentation	207
8.73.3.1	gameStartButton	207
8.73.3.2	networkWidgetLayout	207
8.73.3.3	portEdit	207
8.73.3.4	portLabel	207
8.73.3.5	portValidator	207
8.74	NewFolder Class Reference	207
8.74.1	Detailed Description	208
8.74.2	Constructor & Destructor Documentation	208
8.74.2.1	NewFolder	208
8.74.3	Member Function Documentation	208
8.74.3.1	execute	208
8.74.4	Member Data Documentation	209
8.74.4.1	index	209
8.75	NewMapFile Class Reference	209
8.75.1	Detailed Description	209
8.75.2	Constructor & Destructor Documentation	210
8.75.2.1	NewMapFile	210
8.75.3	Member Function Documentation	210
8.75.3.1	execute	210
8.75.4	Member Data Documentation	210
8.75.4.1	dir	210

8.75.4.2	file	210
8.75.4.3	fileManager	210
8.75.4.4	index	210
8.76	NewWorld Class Reference	210
8.76.1	Detailed Description	211
8.76.2	Constructor & Destructor Documentation	211
8.76.2.1	NewWorld	211
8.76.3	Member Function Documentation	211
8.76.3.1	execute	211
8.76.4	Member Data Documentation	212
8.76.4.1	fileManager	212
8.77	NextTurn Class Reference	212
8.77.1	Detailed Description	212
8.77.2	Constructor & Destructor Documentation	213
8.77.2.1	NextTurn	213
8.77.3	Member Function Documentation	213
8.77.3.1	execute	213
8.78	NotUndoable Class Reference	213
8.78.1	Detailed Description	215
8.78.2	Constructor & Destructor Documentation	215
8.78.2.1	NotUndoable	215
8.78.3	Member Function Documentation	215
8.78.3.1	execute	215
8.78.3.2	isClearingUndoRedo	215
8.78.3.3	isExecutable	216
8.78.3.4	isUndoable	216
8.78.3.5	redo	216
8.78.3.6	undo	216
8.78.4	Member Data Documentation	216
8.78.4.1	clearsUndoRedo	216
8.79	ObjectBase Class Reference	216
8.79.1	Detailed Description	219
8.79.2	Member Enumeration Documentation	219
8.79.2.1	ObjectType	219
8.79.3	Constructor & Destructor Documentation	219
8.79.3.1	ObjectBase	219
8.79.3.2	ObjectBase	219
8.79.3.3	ObjectBase	219
8.79.3.4	ObjectBase	219
8.79.4	Member Function Documentation	219

8.79.4.1	appendImage	219
8.79.4.2	appendImages	219
8.79.4.3	clearImageList	220
8.79.4.4	copyBase	220
8.79.4.5	createInformationWidget	220
8.79.4.6	deserialize	220
8.79.4.7	deserialize	220
8.79.4.8	getCommandList	220
8.79.4.9	getCurrentImagePath	220
8.79.4.10	getDescription	220
8.79.4.11	getImagePathList	220
8.79.4.12	getIsVisible	220
8.79.4.13	getObjectIDNumber	220
8.79.4.14	getObjectName	221
8.79.4.15	getSpawnID	221
8.79.4.16	getType	221
8.79.4.17	getUsedImageIndex	221
8.79.4.18	getX	221
8.79.4.19	getY	221
8.79.4.20	getZ	221
8.79.4.21	initObjectBase	221
8.79.4.22	initObjectBase	221
8.79.4.23	isSet	221
8.79.4.24	loadImage	221
8.79.4.25	loadImages	221
8.79.4.26	refImage	221
8.79.4.27	refImage	222
8.79.4.28	serialize	222
8.79.4.29	serialize	222
8.79.4.30	setDescription	222
8.79.4.31	setIsVisible	222
8.79.4.32	setPosition	222
8.79.4.33	setSpawnID	222
8.79.4.34	setUsedImageIndex	222
8.79.4.35	toggleVisibility	222
8.79.4.36	unsetPosition	222
8.79.4.37	update	222
8.79.4.38	updateBase	222
8.79.5	Member Data Documentation	223
8.79.5.1	description	223

8.79.5.2	imageList	223
8.79.5.3	isVisible	223
8.79.5.4	spawnID	223
8.79.5.5	usedImageIndex	223
8.79.5.6	x	223
8.79.5.7	y	223
8.79.5.8	z	223
8.80	ObjectCharacter Class Reference	223
8.80.1	Detailed Description	224
8.80.2	Constructor & Destructor Documentation	225
8.80.2.1	ObjectCharacter	225
8.80.2.2	ObjectCharacter	225
8.80.3	Member Function Documentation	225
8.80.3.1	copy	225
8.80.3.2	copyBase	225
8.80.3.3	deserialize	225
8.80.3.4	deserialize	225
8.80.3.5	getObjectID	225
8.80.3.6	getObjectIDNumber	225
8.80.3.7	getType	225
8.80.3.8	initObjectCharacter	225
8.80.3.9	serialize	225
8.80.3.10	serialize	226
8.80.3.11	update	226
8.80.3.12	updateCharacter	226
8.81	ObjectCommand Class Reference	226
8.81.1	Detailed Description	227
8.81.2	Constructor & Destructor Documentation	227
8.81.2.1	ObjectCommand	227
8.81.3	Member Function Documentation	227
8.81.3.1	execute	227
8.81.3.2	isExecutable	227
8.81.4	Member Data Documentation	227
8.81.4.1	object	227
8.82	ObjectID Class Reference	227
8.82.1	Detailed Description	228
8.82.2	Member Enumeration Documentation	228
8.82.2.1	CharacterID	228
8.82.2.2	ItemID	229
8.82.2.3	SpecialID	229

8.82.2.4	TileID	229
8.82.3	Constructor & Destructor Documentation	229
8.82.3.1	ObjectID	229
8.82.4	Member Function Documentation	229
8.82.4.1	deserialize	229
8.82.4.2	deserialize	229
8.82.4.3	getNewSpawnIDFor	229
8.82.4.4	getNewSpawnIDFor	229
8.82.4.5	getNewSpawnIDFor	229
8.82.4.6	getNewSpawnIDFor	229
8.82.4.7	getNewSpawnIDFromMap	229
8.82.4.8	serialize	229
8.82.4.9	serialize	229
8.82.5	Member Data Documentation	229
8.82.5.1	characterSpawnIDMap	229
8.82.5.2	itemSpawnIDMap	229
8.82.5.3	specialSpawnIDMap	229
8.82.5.4	tileSpawnIDMap	230
8.83	ObjectImage Class Reference	230
8.83.1	Detailed Description	230
8.83.2	Constructor & Destructor Documentation	230
8.83.2.1	ObjectImage	230
8.83.2.2	ObjectImage	230
8.83.3	Member Function Documentation	231
8.83.3.1	getPath	231
8.83.4	Member Data Documentation	231
8.83.4.1	path	231
8.84	ObjectItem Class Reference	231
8.84.1	Detailed Description	232
8.84.2	Constructor & Destructor Documentation	232
8.84.2.1	ObjectItem	232
8.84.2.2	ObjectItem	232
8.84.3	Member Function Documentation	232
8.84.3.1	copy	232
8.84.3.2	copyBase	232
8.84.3.3	deserialize	232
8.84.3.4	deserialize	232
8.84.3.5	getObjectID	232
8.84.3.6	getObjectIDNumber	232
8.84.3.7	getType	232

8.84.3.8	initObjectItem	232
8.84.3.9	serialize	233
8.84.3.10	serialize	233
8.84.3.11	update	233
8.84.3.12	updateItem	233
8.85	ObjectMenu Class Reference	233
8.85.1	Detailed Description	234
8.85.2	Constructor & Destructor Documentation	234
8.85.2.1	ObjectMenu	234
8.85.3	Member Function Documentation	234
8.85.3.1	triggerCommandByName	234
8.85.4	Member Data Documentation	234
8.85.4.1	object	234
8.85.4.2	processor	234
8.86	ObjectSpecial Class Reference	234
8.86.1	Constructor & Destructor Documentation	235
8.86.1.1	ObjectSpecial	235
8.86.1.2	ObjectSpecial	235
8.86.2	Member Function Documentation	235
8.86.2.1	copy	235
8.86.2.2	copyBase	235
8.86.2.3	deserialize	236
8.86.2.4	deserialize	236
8.86.2.5	getMaxAmountPerStack	236
8.86.2.6	getObjectID	236
8.86.2.7	getObjectIDNumber	236
8.86.2.8	getType	236
8.86.2.9	initObjectSpecial	236
8.86.2.10	isVisibleInGame	236
8.86.2.11	serialize	236
8.86.2.12	serialize	236
8.86.2.13	setMaxAmountPerStack	236
8.86.2.14	setVisibleInGame	236
8.86.2.15	update	236
8.86.2.16	updateSpecial	237
8.86.3	Member Data Documentation	237
8.86.3.1	maxAmountPerStack	237
8.86.3.2	visibleInGame	237
8.87	ObjectTile Class Reference	237
8.87.1	Detailed Description	238

8.87.2	Member Enumeration Documentation	238
8.87.2.1	TileType	238
8.87.3	Constructor & Destructor Documentation	238
8.87.3.1	ObjectTile	238
8.87.3.2	ObjectTile	238
8.87.4	Member Function Documentation	238
8.87.4.1	copy	238
8.87.4.2	copyBase	239
8.87.4.3	deserialize	239
8.87.4.4	deserialize	239
8.87.4.5	getObjectID	239
8.87.4.6	getObjectIDNumber	239
8.87.4.7	getTileType	239
8.87.4.8	getType	239
8.87.4.9	initObjectTile	239
8.87.4.10	serialize	239
8.87.4.11	serialize	239
8.87.4.12	setTileType	239
8.87.4.13	update	239
8.87.4.14	updateTile	240
8.87.5	Member Data Documentation	240
8.87.5.1	tileType	240
8.88	OptionsLayout Class Reference	240
8.88.1	Detailed Description	241
8.88.2	Constructor & Destructor Documentation	241
8.88.2.1	OptionsLayout	241
8.88.3	Member Function Documentation	241
8.88.3.1	addPropertyWidget	241
8.88.3.2	currentIndexChangedTo	241
8.88.3.3	refApplyButton	241
8.88.3.4	refOkButton	241
8.88.3.5	setPropertyListWidth	241
8.88.3.6	showCurrentWidget	241
8.88.4	Member Data Documentation	241
8.88.4.1	applyButton	241
8.88.4.2	okButton	241
8.88.4.3	propertyList	241
8.88.4.4	stackedProperties	241
8.88.4.5	stackLayout	241
8.89	PasteObject Class Reference	241

8.89.1	Detailed Description	242
8.89.2	Constructor & Destructor Documentation	242
8.89.2.1	PasteObject	242
8.89.3	Member Function Documentation	242
8.89.3.1	execute	242
8.89.4	Member Data Documentation	243
8.89.4.1	stackModel	243
8.90	PlaceEditorObject Class Reference	243
8.90.1	Constructor & Destructor Documentation	243
8.90.1.1	PlaceEditorObject	244
8.90.2	Member Function Documentation	244
8.90.2.1	execute	244
8.90.3	Member Data Documentation	244
8.90.3.1	placeableObject	244
8.90.3.2	stackModel	244
8.91	PlaceObject Class Reference	244
8.91.1	Detailed Description	244
8.92	PlaceObjectsState Class Reference	245
8.92.1	Detailed Description	245
8.92.2	Constructor & Destructor Documentation	245
8.92.2.1	PlaceObjectsState	245
8.92.3	Member Function Documentation	245
8.92.3.1	nextState	245
8.92.4	Member Data Documentation	246
8.92.4.1	alreadyPlaced	246
8.93	PlaceSpecial Class Reference	246
8.93.1	Detailed Description	247
8.93.2	Constructor & Destructor Documentation	247
8.93.2.1	PlaceSpecial	247
8.93.3	Member Function Documentation	247
8.93.3.1	execute	247
8.93.3.2	redo	247
8.93.3.3	undo	247
8.93.4	Member Data Documentation	248
8.93.4.1	special	248
8.93.4.2	stackModel	248
8.94	PlaceWall Class Reference	248
8.94.1	Detailed Description	249
8.94.2	Constructor & Destructor Documentation	249
8.94.2.1	PlaceWall	249

8.94.3	Member Function Documentation	249
8.94.3.1	execute	249
8.94.3.2	redo	249
8.94.3.3	undo	249
8.94.4	Member Data Documentation	249
8.94.4.1	oldWall	249
8.94.4.2	stackModel	249
8.94.4.3	wall	250
8.95	Player Class Reference	250
8.95.1	Detailed Description	250
8.95.2	Constructor & Destructor Documentation	251
8.95.2.1	Player	251
8.95.2.2	Player	251
8.95.3	Member Function Documentation	251
8.95.3.1	copy	251
8.95.3.2	createInformationWidget	251
8.95.3.3	deserialize	251
8.95.3.4	deserialize	251
8.95.3.5	getCommandList	251
8.95.3.6	getObjectID	251
8.95.3.7	initPlayer	252
8.95.3.8	serialize	252
8.95.3.9	serialize	252
8.95.3.10	update	252
8.96	PlayerModel Class Reference	252
8.96.1	Detailed Description	253
8.96.2	Constructor & Destructor Documentation	253
8.96.2.1	PlayerModel	253
8.96.2.2	PlayerModel	253
8.96.3	Member Function Documentation	253
8.96.3.1	deserialize	253
8.96.3.2	deserialize	253
8.96.3.3	getID	253
8.96.3.4	getLocalMapPath	253
8.96.3.5	hasCharacterSpawned	253
8.96.3.6	init	253
8.96.3.7	isActivePlayer	253
8.96.3.8	serialize	253
8.96.3.9	serialize	254
8.96.3.10	setActivePlayer	254

8.96.3.11	setCharacterSpawned	254
8.96.3.12	setPlayerMapPath	254
8.96.4	Member Data Documentation	254
8.96.4.1	activePlayer	254
8.96.4.2	characterSpawned	254
8.96.4.3	ID	254
8.96.4.4	playerMapPath	254
8.97	PreloadAllMaps Class Reference	254
8.97.1	Detailed Description	255
8.97.2	Constructor & Destructor Documentation	255
8.97.2.1	PreloadAllMaps	255
8.97.3	Member Function Documentation	255
8.97.3.1	execute	255
8.97.3.2	preload	255
8.97.4	Member Data Documentation	256
8.97.4.1	fileManager	256
8.98	PreloadWorld Class Reference	256
8.98.1	Detailed Description	257
8.98.2	Constructor & Destructor Documentation	257
8.98.2.1	PreloadWorld	257
8.98.3	Member Function Documentation	257
8.98.3.1	execute	257
8.98.4	Member Data Documentation	257
8.98.4.1	fileManager	257
8.99	Processor Class Reference	257
8.99.1	Detailed Description	258
8.99.2	Constructor & Destructor Documentation	258
8.99.2.1	Processor	258
8.99.3	Member Function Documentation	258
8.99.3.1	commandProcessed	258
8.99.3.2	execute	259
8.99.3.3	isExecutable	260
8.99.3.4	refUndoRedo	260
8.99.3.5	setEditorModel	260
8.99.3.6	setGameModel	260
8.99.3.7	setMapModel	260
8.99.3.8	setNetworkModel	260
8.99.3.9	setWorldModel	260
8.99.4	Member Data Documentation	260
8.99.4.1	editorModel	260

8.99.4.2	gameModel	260
8.99.4.3	mapModel	260
8.99.4.4	networkModel	260
8.99.4.5	undoRedo	260
8.99.4.6	worldModel	260
8.100	Quit Class Reference	261
8.100.1	Detailed Description	261
8.100.2	Constructor & Destructor Documentation	261
8.100.2.1	Quit	261
8.100.3	Member Function Documentation	262
8.100.3.1	execute	262
8.100.4	Member Data Documentation	262
8.100.4.1	askToSaveGame	262
8.101	ReceiveChangePlayerID Class Reference	262
8.101.1	Detailed Description	263
8.101.2	Constructor & Destructor Documentation	263
8.101.2.1	ReceiveChangePlayerID	263
8.101.3	Member Function Documentation	263
8.101.3.1	execute	263
8.102	ReceiveInitialPlayerInformation Class Reference	263
8.102.1	Detailed Description	264
8.102.2	Constructor & Destructor Documentation	264
8.102.2.1	ReceiveInitialPlayerInformation	264
8.102.3	Member Function Documentation	264
8.102.3.1	execute	264
8.103	ReceiveMap Class Reference	265
8.103.1	Detailed Description	265
8.103.2	Constructor & Destructor Documentation	266
8.103.2.1	ReceiveMap	266
8.103.3	Member Function Documentation	266
8.103.3.1	execute	266
8.104	ReceivePlayer Class Reference	266
8.104.1	Detailed Description	267
8.104.2	Constructor & Destructor Documentation	267
8.104.2.1	ReceivePlayer	267
8.104.3	Member Function Documentation	267
8.104.3.1	execute	267
8.105	ReceivePlayerModels Class Reference	267
8.105.1	Detailed Description	268
8.105.2	Constructor & Destructor Documentation	268

8.105.2.1 ReceivePlayerModels	268
8.105.3 Member Function Documentation	268
8.105.3.1 execute	268
8.106 ReceiveSpawnCharacterFrom Class Reference	269
8.106.1 Detailed Description	269
8.106.2 Constructor & Destructor Documentation	270
8.106.2.1 ReceiveSpawnCharacterFrom	270
8.106.3 Member Function Documentation	270
8.106.3.1 execute	270
8.106.4 Member Data Documentation	270
8.106.4.1 senderID	270
8.107 ReceiveUpdateCharacterFrom Class Reference	270
8.107.1 Detailed Description	271
8.107.2 Constructor & Destructor Documentation	271
8.107.2.1 ReceiveUpdateCharacterFrom	271
8.107.3 Member Function Documentation	271
8.107.3.1 execute	271
8.107.4 Member Data Documentation	271
8.107.4.1 senderID	271
8.108 RemoveAllObjects Class Reference	272
8.108.1 Detailed Description	272
8.108.2 Constructor & Destructor Documentation	273
8.108.2.1 RemoveAllObjects	273
8.108.3 Member Function Documentation	273
8.108.3.1 execute	273
8.108.3.2 redo	273
8.108.3.3 undo	273
8.108.4 Member Data Documentation	273
8.108.4.1 stackModel	273
8.108.4.2 undoCharacter	273
8.108.4.3 undoFloor	273
8.108.4.4 undoItemList	273
8.108.4.5 undoSpecialList	274
8.108.4.6 undoWall	274
8.109 RemoveFolder Class Reference	274
8.109.1 Detailed Description	275
8.109.2 Constructor & Destructor Documentation	275
8.109.2.1 RemoveFolder	275
8.109.3 Member Function Documentation	275
8.109.3.1 execute	275

8.109.3.2 removeDirRecursive	275
8.109.3.3 removeEmpty	275
8.109.4 Member Data Documentation	275
8.109.4.1 folder	275
8.110 RemoveFromChosenList Class Reference	276
8.110.1 Detailed Description	276
8.110.2 Constructor & Destructor Documentation	277
8.110.2.1 RemoveFromChosenList	277
8.110.3 Member Function Documentation	277
8.110.3.1 execute	277
8.110.4 Member Data Documentation	277
8.110.4.1 chosenListModel	277
8.110.4.2 currentIndex	277
8.111 RemoveObjectsState Class Reference	277
8.111.1 Detailed Description	278
8.111.2 Constructor & Destructor Documentation	278
8.111.2.1 RemoveObjectsState	278
8.111.3 Member Function Documentation	278
8.111.3.1 nextState	278
8.111.4 Member Data Documentation	278
8.111.4.1 alreadyPlaced	278
8.112 RemoveWall Class Reference	279
8.112.1 Detailed Description	279
8.112.2 Constructor & Destructor Documentation	280
8.112.2.1 RemoveWall	280
8.112.3 Member Function Documentation	280
8.112.3.1 execute	280
8.112.3.2 redo	280
8.112.3.3 undo	280
8.112.4 Member Data Documentation	280
8.112.4.1 oldWall	280
8.112.4.2 stackModel	280
8.113 RemoveWorldEntry Class Reference	280
8.113.1 Detailed Description	281
8.113.2 Constructor & Destructor Documentation	281
8.113.2.1 RemoveWorldEntry	281
8.113.3 Member Function Documentation	281
8.113.3.1 execute	281
8.113.4 Member Data Documentation	282
8.113.4.1 index	282

8.113.4.2 mayAsk	282
8.114RenameWorldEntry Class Reference	282
8.114.1 Detailed Description	283
8.114.2 Constructor & Destructor Documentation	283
8.114.2.1 RenameWorldEntry	283
8.114.3 Member Function Documentation	283
8.114.3.1 execute	283
8.114.4 Member Data Documentation	283
8.114.4.1 index	283
8.115ReplyToRequestForMap Class Reference	283
8.115.1 Detailed Description	284
8.115.2 Constructor & Destructor Documentation	284
8.115.2.1 ReplyToRequestForMap	284
8.115.3 Member Function Documentation	284
8.115.3.1 execute	284
8.116ReplyToRequestForPlayer Class Reference	285
8.116.1 Detailed Description	285
8.116.2 Constructor & Destructor Documentation	286
8.116.2.1 ReplyToRequestForPlayer	286
8.116.3 Member Function Documentation	286
8.116.3.1 execute	286
8.117ReplyToRequestForPlayerModels Class Reference	286
8.117.1 Detailed Description	287
8.117.2 Constructor & Destructor Documentation	287
8.117.2.1 ReplyToRequestForPlayerModels	287
8.117.3 Member Function Documentation	287
8.117.3.1 execute	287
8.118RequestMapFromServer Class Reference	287
8.118.1 Detailed Description	288
8.118.2 Constructor & Destructor Documentation	288
8.118.2.1 RequestMapFromServer	288
8.118.3 Member Function Documentation	288
8.118.3.1 execute	288
8.118.4 Member Data Documentation	289
8.118.4.1 localMapPath	289
8.119RequestPlayerFromServer Class Reference	289
8.119.1 Detailed Description	290
8.119.2 Constructor & Destructor Documentation	290
8.119.2.1 RequestPlayerFromServer	290
8.119.3 Member Function Documentation	290

8.119.3.1 execute	290
8.119.4 Member Data Documentation	290
8.119.4.1 playerId	290
8.120RequestPlayerModelsFromServer Class Reference	290
8.120.1 Detailed Description	291
8.120.2 Constructor & Destructor Documentation	291
8.120.2.1 RequestPlayerModelsFromServer	291
8.120.3 Member Function Documentation	291
8.120.3.1 execute	291
8.121ResumeEMailGame Class Reference	292
8.121.1 Detailed Description	292
8.121.2 Constructor & Destructor Documentation	293
8.121.2.1 ResumeEMailGame	293
8.121.3 Member Function Documentation	293
8.121.3.1 execute	293
8.121.4 Member Data Documentation	293
8.121.4.1 fileManager	293
8.122RSAMethod Class Reference	293
8.122.1 Detailed Description	294
8.122.2 Constructor & Destructor Documentation	294
8.122.2.1 RSAMethod	294
8.122.2.2 RSAMethod	294
8.122.3 Member Function Documentation	295
8.122.3.1 decryptString	295
8.122.3.2 encryptString	295
8.122.3.3 generateKeys	295
8.122.3.4 getMaxCipherTextLength	295
8.122.3.5 getMaxPlainTextLength	295
8.122.3.6 init	295
8.122.3.7 load	295
8.122.3.8 loadKey	295
8.122.3.9 loadKeys	295
8.122.3.10save	296
8.122.3.11saveKey	296
8.122.4 Member Data Documentation	296
8.122.4.1 maxCipherTextLength	296
8.122.4.2 maxPlainTextLength	296
8.122.4.3 privateKey	296
8.122.4.4 privateKeyFilePath	296
8.122.4.5 publicKey	296

8.122.4.6 publicKeyFilePath	296
8.122.4.7 randomNumberGenerator	296
8.123 SaveEMailGame Class Reference	296
8.123.1 Detailed Description	297
8.123.2 Constructor & Destructor Documentation	297
8.123.2.1 SaveEMailGame	297
8.123.3 Member Function Documentation	297
8.123.3.1 execute	297
8.123.4 Member Data Documentation	297
8.123.4.1 fileManager	297
8.123.4.2 mailGameInfo	297
8.123.4.3 playerName	297
8.123.4.4 turnCount	297
8.124 SaveGame Class Reference	298
8.124.1 Detailed Description	298
8.124.2 Constructor & Destructor Documentation	298
8.124.2.1 SaveGame	298
8.124.3 Member Function Documentation	299
8.124.3.1 execute	299
8.124.4 Member Data Documentation	299
8.124.4.1 fileManager	299
8.125 SaveGameFile Class Reference	299
8.125.1 Detailed Description	300
8.125.2 Constructor & Destructor Documentation	300
8.125.2.1 SaveGameFile	300
8.125.3 Member Function Documentation	300
8.125.3.1 deserialize	300
8.125.3.2 fromGameType	300
8.125.3.3 serialize	300
8.125.4 Member Data Documentation	300
8.125.4.1 gameModel	300
8.125.4.2 processor	300
8.125.4.3 worldModel	300
8.126 SaveMap Class Reference	301
8.126.1 Detailed Description	301
8.126.2 Constructor & Destructor Documentation	301
8.126.2.1 SaveMap	301
8.126.3 Member Function Documentation	301
8.126.3.1 execute	302
8.126.4 Member Data Documentation	302

8.126.4.1 fileManager	302
8.127SaveMapAs Class Reference	302
8.127.1 Detailed Description	303
8.127.2 Constructor & Destructor Documentation	303
8.127.2.1 SaveMapAs	303
8.127.3 Member Function Documentation	303
8.127.3.1 execute	303
8.127.4 Member Data Documentation	303
8.127.4.1 fileManager	303
8.128SaveWorld Class Reference	303
8.128.1 Detailed Description	304
8.128.2 Constructor & Destructor Documentation	304
8.128.2.1 SaveWorld	304
8.128.3 Member Function Documentation	304
8.128.3.1 execute	304
8.128.4 Member Data Documentation	305
8.128.4.1 fileManager	305
8.129SaveWorldAs Class Reference	305
8.129.1 Detailed Description	305
8.129.2 Constructor & Destructor Documentation	306
8.129.2.1 SaveWorldAs	306
8.129.3 Member Function Documentation	306
8.129.3.1 execute	306
8.129.4 Member Data Documentation	306
8.129.4.1 fileManager	306
8.130SendInitialPlayerInformation Class Reference	306
8.130.1 Detailed Description	307
8.130.2 Constructor & Destructor Documentation	307
8.130.2.1 SendInitialPlayerInformation	307
8.130.3 Member Function Documentation	307
8.130.3.1 execute	307
8.131SendSpawnCharacterToServer Class Reference	307
8.131.1 Detailed Description	308
8.131.2 Constructor & Destructor Documentation	308
8.131.2.1 SendSpawnCharacterToServer	308
8.131.3 Member Function Documentation	308
8.131.3.1 execute	308
8.131.4 Member Data Documentation	309
8.131.4.1 newCharacter	309
8.132SendToClientNextTurn Class Reference	309

8.132.1 Detailed Description	309
8.132.2 Constructor & Destructor Documentation	310
8.132.2.1 SendToClientNextTurn	310
8.132.3 Member Function Documentation	310
8.132.3.1 execute	310
8.133 SendToServerEndTurn Class Reference	310
8.133.1 Detailed Description	311
8.133.2 Constructor & Destructor Documentation	311
8.133.2.1 SendToServerEndTurn	311
8.133.3 Member Function Documentation	311
8.133.3.1 execute	311
8.134 SendUpdateCharacterToServer Class Reference	311
8.134.1 Detailed Description	312
8.134.2 Constructor & Destructor Documentation	312
8.134.2.1 SendUpdateCharacterToServer	312
8.134.3 Member Function Documentation	313
8.134.3.1 execute	313
8.134.4 Member Data Documentation	313
8.134.4.1 newCharacter	313
8.134.4.2 oldCharacter	313
8.135 Serialize Class Reference	313
8.135.1 Detailed Description	314
8.135.2 Constructor & Destructor Documentation	314
8.135.2.1 Serialize	314
8.135.3 Member Function Documentation	314
8.135.3.1 deserialize	314
8.135.3.2 getFileInfo	314
8.135.3.3 resetFileInfo	314
8.135.3.4 serialize	314
8.135.3.5 setFileInfo	314
8.135.3.6 setFileInfo	314
8.135.4 Member Data Documentation	314
8.135.4.1 fileInfo	314
8.136 ServerCommand Class Reference	315
8.136.1 Detailed Description	316
8.136.2 Constructor & Destructor Documentation	316
8.136.2.1 ServerCommand	316
8.136.3 Member Function Documentation	316
8.136.3.1 execute	316
8.136.3.2 isExecutable	316

8.136.4 Member Data Documentation	316
8.136.4.1 playerID	316
8.137 ServerGameModel Class Reference	317
8.137.1 Detailed Description	318
8.137.2 Constructor & Destructor Documentation	318
8.137.2.1 ServerGameModel	318
8.137.3 Member Function Documentation	318
8.137.3.1 beginTurn	318
8.137.3.2 convertFromDifferentGameModel	318
8.137.3.3 deactivatePlayer	318
8.137.3.4 deserialize	318
8.137.3.5 endTurn	319
8.137.3.6 firstTurn	319
8.137.3.7 getCurrentPlayerID	319
8.137.3.8 getGameType	319
8.137.3.9 isYourTurn	319
8.137.3.10 nextCurrentPlayer	319
8.137.3.11 nextTurn	319
8.137.3.12 refCurrentMap	319
8.137.3.13 serialize	319
8.137.3.14 setCurrentPlayerID	319
8.137.3.15 spawnCharacter	319
8.137.3.16 updateCharacter	320
8.137.4 Member Data Documentation	320
8.137.4.1 currentPlayerID	320
8.138 ServerModel Class Reference	320
8.138.1 Constructor & Destructor Documentation	321
8.138.1.1 ServerModel	321
8.138.1.2 ~ServerModel	321
8.138.2 Member Function Documentation	321
8.138.2.1 changeClientID	321
8.138.2.2 deleteDisconnectedClient	322
8.138.2.3 getConnectedPlayerIDs	322
8.138.2.4 getConnectionList	322
8.138.2.5 isClient	322
8.138.2.6 isServer	322
8.138.2.7 listen	322
8.138.2.8 newConnection	322
8.138.2.9 receiveData	322
8.138.2.10 receiveData	322

8.138.2.11	receiveData	322
8.138.2.12	receiveData	323
8.138.2.13	receiveData	323
8.138.2.14	receivePendingData	323
8.138.2.15	sendData	323
8.138.2.16	sendData	323
8.138.2.17	sendData	323
8.138.2.18	sendData	323
8.138.2.19	sendData	323
8.138.2.20	socketError	323
8.138.3	Member Data Documentation	323
8.138.3.1	clients	323
8.138.3.2	processor	323
8.138.3.3	server	323
8.138.3.4	worldModel	324
8.139	SetupNewGame Class Reference	324
8.139.1	Detailed Description	325
8.139.2	Constructor & Destructor Documentation	325
8.139.2.1	SetupNewGame	325
8.139.3	Member Function Documentation	325
8.139.3.1	applyGeneralProperties	325
8.139.3.2	execute	325
8.139.3.3	initEMailGame	325
8.139.3.4	initGameSetupDialog	326
8.139.3.5	initNetwork	326
8.139.3.6	initSoloHotseat	326
8.139.4	Member Data Documentation	326
8.139.4.1	emailSetupWidget	326
8.139.4.2	gameModelPointer	326
8.139.4.3	gameSetupDialog	326
8.139.4.4	gameSetupLayout	326
8.139.4.5	generalSetupWidget	326
8.139.4.6	networkSetupWidget	326
8.139.4.7	soloHotseatSetupWidget	326
8.140	ShowAbout Class Reference	326
8.140.1	Detailed Description	327
8.140.2	Constructor & Destructor Documentation	327
8.140.2.1	ShowAbout	327
8.140.3	Member Function Documentation	327
8.140.3.1	execute	327

8.141 ShowDescription Class Reference	328
8.141.1 Detailed Description	328
8.141.2 Constructor & Destructor Documentation	328
8.141.2.1 ShowDescription	328
8.141.3 Member Function Documentation	328
8.141.3.1 execute	328
8.142 ShowGameOptions Class Reference	329
8.142.1 Detailed Description	330
8.142.2 Constructor & Destructor Documentation	330
8.142.2.1 ShowGameOptions	330
8.142.3 Member Function Documentation	330
8.142.3.1 addHotseatPlayer	330
8.142.3.2 choseSaveDir	330
8.142.3.3 execute	330
8.142.3.4 init	330
8.142.3.5 leaveGame	330
8.142.3.6 saveGame	330
8.142.4 Member Data Documentation	331
8.142.4.1 addNewPlayerButton	331
8.142.4.2 gameOptionsDialog	331
8.142.4.3 gameOptionsLayout	331
8.142.4.4 leaveGameButton	331
8.142.4.5 saveDirChooseButton	331
8.142.4.6 saveDirLabel	331
8.142.4.7 saveDirLineEdit	331
8.142.4.8 saveGameButton	331
8.142.4.9 specificGameTypeLayout	331
8.142.4.10 specificGameTypeWidget	331
8.143 SoloHotseatSetupWidget Class Reference	331
8.143.1 Detailed Description	331
8.143.2 Constructor & Destructor Documentation	332
8.143.2.1 SoloHotseatSetupWidget	332
8.143.3 Member Data Documentation	332
8.143.3.1 gameStartButton	332
8.143.3.2 soloHotseatWidgetLayout	332
8.144 SpawnGameObject Class Reference	332
8.144.1 Detailed Description	333
8.144.2 Constructor & Destructor Documentation	333
8.144.2.1 SpawnGameObject	333
8.144.3 Member Function Documentation	333

8.144.3.1 execute	333
8.144.4 Member Data Documentation	333
8.144.4.1 spawnableObject	333
8.144.4.2 stackModel	333
8.145 SpawnPlayer Class Reference	334
8.145.1 Detailed Description	334
8.146 SpawnPlayerOnCurrentMap Class Reference	334
8.146.1 Constructor & Destructor Documentation	335
8.146.1.1 SpawnPlayerOnCurrentMap	335
8.146.2 Member Function Documentation	335
8.146.2.1 execute	335
8.146.3 Member Data Documentation	335
8.146.3.1 playerName	335
8.147 StackModel Class Reference	335
8.147.1 Detailed Description	338
8.147.2 Member Enumeration Documentation	338
8.147.2.1 PathIndication	338
8.147.3 Constructor & Destructor Documentation	339
8.147.3.1 StackModel	339
8.147.3.2 StackModel	339
8.147.3.3 StackModel	339
8.147.4 Member Function Documentation	339
8.147.4.1 characterChanged	339
8.147.4.2 characterExists	339
8.147.4.3 characterExists	339
8.147.4.4 deleteAll	339
8.147.4.5 deleteAllItems	339
8.147.4.6 deleteAllSpecials	339
8.147.4.7 deleteCharacter	339
8.147.4.8 deleteFloor	339
8.147.4.9 deleteItem	340
8.147.4.10 deleteSpecial	341
8.147.4.11 deleteWall	341
8.147.4.12 deserialize	341
8.147.4.13 floorChanged	341
8.147.4.14 getHeight	341
8.147.4.15 getPathIndication	341
8.147.4.16 getSquarePos	341
8.147.4.17 hasCharacter	341
8.147.4.18 hasFloor	341

8.147.4.19	hasSpaceForSpecial	341
8.147.4.20	hasWall	342
8.147.4.21	initStack	342
8.147.4.22	isEmpty	342
8.147.4.23	itemListChanged	342
8.147.4.24	itemListSize	342
8.147.4.25	pathIndicationChanged	342
8.147.4.26	playerRemoved	342
8.147.4.27	playerSet	342
8.147.4.28	prependItem	342
8.147.4.29	prependSpecial	342
8.147.4.30	refCharacter	342
8.147.4.31	refFloor	342
8.147.4.32	refItem	342
8.147.4.33	refItemList	342
8.147.4.34	refSpecial	343
8.147.4.35	refSpecialList	343
8.147.4.36	refWall	343
8.147.4.37	removeAll	343
8.147.4.38	removeAllItems	343
8.147.4.39	removeAllSpecials	343
8.147.4.40	removeCharacter	343
8.147.4.41	removeFloor	343
8.147.4.42	removeItem	343
8.147.4.43	removeSpecial	343
8.147.4.44	removeWall	343
8.147.4.45	replaceCharacter	343
8.147.4.46	replaceFloor	344
8.147.4.47	replaceWall	344
8.147.4.48	resetPathDirection	344
8.147.4.49	serialize	344
8.147.4.50	setCharacter	344
8.147.4.51	setFloor	344
8.147.4.52	setItemList	344
8.147.4.53	setPathIndication	344
8.147.4.54	setPosition	344
8.147.4.55	setSpecialList	344
8.147.4.56	setWall	344
8.147.4.57	specialIDExists	344
8.147.4.58	specialListChanged	345

8.147.4.59specialListSize	345
8.147.4.60startPositionRemoved	345
8.147.4.61startPositionSet	345
8.147.4.62takeAllItems	345
8.147.4.63takeAllSpecial	345
8.147.4.64takeCharacter	345
8.147.4.65takeFloor	345
8.147.4.66takeItem	346
8.147.4.67takeSpecial	347
8.147.4.68takeWall	347
8.147.4.69wallChanged	347
8.147.5 Member Data Documentation	347
8.147.5.1 character	347
8.147.5.2 floor	347
8.147.5.3 height	347
8.147.5.4 itemList	347
8.147.5.5 pathIndication	347
8.147.5.6 specialList	347
8.147.5.7 squarePos	347
8.147.5.8 wall	347
8.148StartPosition Class Reference	348
8.148.1 Detailed Description	348
8.148.2 Constructor & Destructor Documentation	349
8.148.2.1 StartPosition	349
8.148.2.2 StartPosition	349
8.148.3 Member Function Documentation	349
8.148.3.1 copy	349
8.148.3.2 createInformationWidget	349
8.148.3.3 deserialize	349
8.148.3.4 deserialize	349
8.148.3.5 getCommandList	349
8.148.3.6 getObjectID	349
8.148.3.7 serialize	349
8.148.3.8 serialize	349
8.148.3.9 update	350
8.149TargetChosenState Class Reference	350
8.149.1 Detailed Description	350
8.149.2 Constructor & Destructor Documentation	350
8.149.2.1 TargetChosenState	350
8.149.3 Member Function Documentation	350

8.149.3.1 nextState	350
8.150TargetConfirmedState Class Reference	351
8.150.1 Detailed Description	351
8.150.2 Constructor & Destructor Documentation	352
8.150.2.1 TargetConfirmedState	352
8.150.3 Member Function Documentation	352
8.150.3.1 nextState	352
8.151TargetMoveState Class Reference	352
8.151.1 Detailed Description	352
8.151.2 Constructor & Destructor Documentation	353
8.151.2.1 TargetMoveState	353
8.151.3 Member Function Documentation	353
8.151.3.1 nextState	353
8.152Undoable Class Reference	353
8.152.1 Detailed Description	354
8.152.2 Constructor & Destructor Documentation	354
8.152.2.1 Undoable	354
8.152.3 Member Function Documentation	354
8.152.3.1 execute	354
8.152.3.2 isClearingUndoRedo	355
8.152.3.3 isExecutable	355
8.152.3.4 isUndoable	355
8.152.3.5 redo	355
8.152.3.6 undo	355
8.153UndoRedo Class Reference	355
8.153.1 Detailed Description	357
8.153.2 Constructor & Destructor Documentation	357
8.153.2.1 UndoRedo	357
8.153.2.2 ~UndoRedo	357
8.153.3 Member Function Documentation	357
8.153.3.1 add	357
8.153.3.2 clear	357
8.153.3.3 clearRedoList	357
8.153.3.4 clearUndoList	357
8.153.3.5 hasOnlySavedMapChanges	357
8.153.3.6 hasOnlySavedWorldChanges	357
8.153.3.7 redo	357
8.153.3.8 redoChanged	358
8.153.3.9 setOnlySavedMapChanges	358
8.153.3.10setOnlySavedWorldChanges	358

8.153.3.11setPossibleToReachOriginalState	358
8.153.3.12setSavedStates	358
8.153.3.13undo	358
8.153.3.14undoChanged	358
8.153.4 Member Data Documentation	358
8.153.4.1 onlySavedMapChanges	358
8.153.4.2 onlySavedWorldChanges	358
8.153.4.3 possibleToReachOriginalState	358
8.153.4.4 redoList	358
8.153.4.5 redoListSavedState	358
8.153.4.6 undoList	358
8.153.4.7 undoListSavedState	358
8.154UseRedo Class Reference	359
8.154.1 Detailed Description	359
8.154.2 Constructor & Destructor Documentation	359
8.154.2.1 UseRedo	359
8.154.3 Member Function Documentation	359
8.154.3.1 execute	359
8.155UseUndo Class Reference	360
8.155.1 Detailed Description	360
8.155.2 Constructor & Destructor Documentation	361
8.155.2.1 UseUndo	361
8.155.3 Member Function Documentation	361
8.155.3.1 execute	361
8.156WaitingForDataDialog Class Reference	361
8.156.1 Constructor & Destructor Documentation	362
8.156.1.1 WaitingForDataDialog	362
8.156.1.2 WaitingForDataDialog	362
8.156.2 Member Function Documentation	362
8.156.2.1 dataReceived	362
8.156.2.2 dataReceived	362
8.156.2.3 waitFor	362
8.156.2.4 waitFor	362
8.156.3 Member Data Documentation	362
8.156.3.1 fromClient	362
8.156.3.2 fromServer	362
8.156.3.3 hasBeenReceived	363
8.156.3.4 messageBox	363
8.156.3.5 networkModel	363
8.157WorldFileTree Class Reference	363

8.157.1 Detailed Description	364
8.157.2 Constructor & Destructor Documentation	364
8.157.2.1 WorldFileTree	364
8.157.3 Member Function Documentation	364
8.157.3.1 dragMoveEvent	364
8.157.3.2 dropEvent	364
8.157.3.3 duplicate	364
8.157.3.4 indexDoubleClicked	364
8.157.3.5 mousePressEvent	365
8.157.3.6 mouseReleaseEvent	365
8.157.3.7 newFolder	365
8.157.3.8 newMap	365
8.157.3.9 refContextMenu	365
8.157.3.10 remove	365
8.157.3.11 rename	365
8.157.3.12 updateColumnWidth	365
8.157.3.13 updateRootIndex	365
8.157.4 Member Data Documentation	365
8.157.4.1 contextMenu	365
8.157.4.2 draggedIndex	365
8.157.4.3 duplicateAction	365
8.157.4.4 newFolderAction	365
8.157.4.5 newMapAction	365
8.157.4.6 processor	365
8.157.4.7 removeAction	365
8.157.4.8 renameAction	365
8.157.4.9 worldModel	366
8.158 WorldModel Class Reference	366
8.158.1 Detailed Description	367
8.158.2 Constructor & Destructor Documentation	367
8.158.2.1 WorldModel	367
8.158.3 Member Function Documentation	367
8.158.3.1 absoluteMapPathToLocalMapPath	367
8.158.3.2 deserialize	367
8.158.3.3 deserialize	367
8.158.3.4 fileChanged	367
8.158.3.5 getStartMapWorldPaths	367
8.158.3.6 localMapPathToAbsolutePath	367
8.158.3.7 refObjectID	367
8.158.3.8 refWorldFileInfo	367

8.158.3.9 serialize	367
8.158.3.10serialize	367
8.158.3.11setStartMaps	367
8.158.3.12setStartMaps	367
8.158.3.13setWorldFileInfo	367
8.158.4 Member Data Documentation	367
8.158.4.1 objectID	367
8.158.4.2 startMaps	368
8.158.4.3 worldFileInfo	368
8.159WorldPropertiesDialog Class Reference	368
8.159.1 Detailed Description	369
8.159.2 Constructor & Destructor Documentation	369
8.159.2.1 WorldPropertiesDialog	369
8.159.3 Member Function Documentation	369
8.159.3.1 addPathToChosen	369
8.159.3.2 applyChanges	369
8.159.3.3 availableTreeUpdateColumnWidth	369
8.159.3.4 exec	369
8.159.3.5 removePathFromChosen	369
8.159.3.6 updateAvailableFileSystemModel	369
8.159.3.7 updateChosenMaps	369
8.159.4 Member Data Documentation	369
8.159.4.1 addButton	369
8.159.4.2 availableFileSystemModel	369
8.159.4.3 availableTreeView	369
8.159.4.4 chosenListModel	369
8.159.4.5 chosenListView	369
8.159.4.6 manageStartingMaps	369
8.159.4.7 manageStartingMapsLayout	370
8.159.4.8 optionsLayout	370
8.159.4.9 processor	370
8.159.4.10removeButton	370
8.159.4.11worldModel	370
8.160WorldView Class Reference	370
8.160.1 Detailed Description	370
8.160.2 Constructor & Destructor Documentation	371
8.160.2.1 WorldView	371
8.160.3 Member Function Documentation	371
8.160.3.1 resizeWorldView	371
8.160.4 Member Data Documentation	371

8.160.4.1 fileTree	371
8.160.4.2 processor	371
8.160.4.3 worldModel	371
8.161 YesNoCancelDialog Class Reference	371
8.161.1 Detailed Description	372
8.161.2 Member Enumeration Documentation	372
8.161.2.1 ButtonClicked	372
8.161.3 Constructor & Destructor Documentation	372
8.161.3.1 YesNoCancelDialog	372
8.161.4 Member Function Documentation	372
8.161.4.1 getButtonClicked	372
8.161.4.2 showDialog	372
8.161.5 Member Data Documentation	372
8.161.5.1 buttonClicked	372
8.161.5.2 noButton	372
8.161.5.3 yesButton	373
8.162 YourTurnInformation Class Reference	373
8.162.1 Detailed Description	373
8.162.2 Constructor & Destructor Documentation	374
8.162.2.1 YourTurnInformation	374
8.162.3 Member Function Documentation	374
8.162.3.1 execute	374
9 File Documentation	375
9.1 /home/ryoga/git/pidgirl/pidgirl-engine/AStar/AStarMap.cpp File Reference	375
9.2 /home/ryoga/git/pidgirl/pidgirl-engine/AStar/AStarMap.h File Reference	375
9.3 /home/ryoga/git/pidgirl/pidgirl-engine/AStar/AStarMapEntry.cpp File Reference	375
9.4 /home/ryoga/git/pidgirl/pidgirl-engine/AStar/AStarMapEntry.h File Reference	375
9.5 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/BaseCommand.cpp File Reference	376
9.6 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/BaseCommand.h File Reference	376
9.7 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Command.cpp File Reference	376
9.8 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Command.h File Reference	376
9.9 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/CommonCommand.cpp File Reference	377
9.10 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/CommonCommand.h File Reference	377
9.11 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/CopyFolder.cpp File Reference	377
9.12 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/CopyFolder.h File Reference	377
9.13 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/Quit.cpp File Reference	378
9.14 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/Quit.h File Reference	378
9.15 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/RemoveFolder.cpp File Reference	378

9.16	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/RemoveFolder.h File Reference	378
9.17	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/AskForSave.cpp File Reference . .	379
9.18	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/AskForSave.h File Reference . . .	379
9.19	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/ApplyMapGeometryChanges.cpp File Reference	379
9.20	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/ApplyMapGeometryChanges.h File Reference	379
9.21	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/AskForSaveMap.cpp File Ref- erence	379
9.22	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/AskForSaveMap.h File Refer- ence	380
9.23	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/ChangeZoomFactor.cpp File Reference	380
9.24	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/ChangeZoomFactor.h File Reference	380
9.25	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/CopyWall.cpp File Reference	380
9.26	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/CopyWall.h File Reference . .	380
9.27	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/CutWall.cpp File Reference .	381
9.28	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/CutWall.h File Reference . . .	381
9.29	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/LoadMap.cpp File Reference	381
9.30	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/LoadMap.h File Reference . .	381
9.31	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/LoadMapByIndex.cpp File Reference	381
9.32	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/LoadMapByIndex.h File Ref- erence	382
9.33	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/PasteObject.cpp File Reference	382
9.34	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/PasteObject.h File Reference	382
9.35	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/PlaceEditorObject.cpp File Reference	382
9.36	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/PlaceEditorObject.h File Ref- erence	382
9.37	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/PlaceSpecial.cpp File Refer- ence	383
9.38	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/PlaceSpecial.h File Reference	383
9.39	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/PlaceWall.cpp File Reference	383
9.40	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/PlaceWall.h File Reference .	383
9.41	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/RemoveAllObjects.cpp File Reference	384
9.42	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/RemoveAllObjects.h File Ref- erence	384
9.43	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/RemoveWall.cpp File Reference	384
9.44	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/RemoveWall.h File Reference	384
9.45	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Tools/ChooseMainTool.cpp File Reference	384

9.46	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Tools/ChooseMainTool.h File Reference	385
9.47	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/AddToChosenList.cpp File Reference	385
9.48	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/AddToChosenList.h File Reference	385
9.49	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/ApplyWorldProperties-Changes.cpp File Reference	385
9.50	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/ApplyWorldProperties-Changes.h File Reference	386
9.51	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/AskForSaveWorld.cpp File Reference	386
9.52	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/AskForSaveWorld.h File Reference	386
9.53	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/DuplicateWorldEntry.cpp File Reference	386
9.54	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/DuplicateWorldEntry.h File Reference	386
9.55	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/LoadWorld.cpp File Reference	387
9.56	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/LoadWorld.h File Reference	387
9.57	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/MoveFile.cpp File Reference	387
9.58	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/MoveFile.h File Reference	387
9.59	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/NewFolder.cpp File Reference	388
9.60	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/NewFolder.h File Reference	388
9.61	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/NewMapFile.cpp File Reference	388
9.62	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/NewMapFile.h File Reference	388
9.63	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/NewWorld.cpp File Reference	389
9.64	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/NewWorld.h File Reference	389
9.65	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/RemoveFromChosen-List.cpp File Reference	389
9.66	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/RemoveFromChosenList.h File Reference	389
9.67	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/RemoveWorldEntry.cpp File Reference	390
9.68	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/RemoveWorldEntry.h File Reference	390
9.69	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/RenameWorldEntry.cpp File Reference	390
9.70	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/RenameWorldEntry.h File Reference	390
9.71	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/SaveMap.cpp File Reference	391
9.72	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/SaveMap.h File Reference	391
9.73	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/SaveMapAs.cpp File Reference	391

9.74	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/SaveMapAs.h File Reference	391
9.75	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/SaveWorld.cpp File Reference	392
9.76	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/SaveWorld.h File Reference	392
9.77	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/SaveWorldAs.cpp File Reference	392
9.78	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/SaveWorldAs.h File Reference	392
9.79	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/GameCommand.cpp File Reference	393
9.80	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/GameCommand.h File Reference	393
9.81	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/CreateCharacter.cpp File Reference	393
9.82	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/CreateCharacter.h File Reference	393
9.83	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/EndTurn.cpp File Reference	394
9.84	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/EndTurn.h File Reference	394
9.85	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/FirstTurn.cpp File Reference	394
9.86	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/FirstTurn.h File Reference	394
9.87	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/LoadStartMapFrom-Index.cpp File Reference	395
9.88	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/LoadStartMapFrom-Index.h File Reference	395
9.89	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/NextTurn.cpp File Reference	395
9.90	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/NextTurn.h File Reference	395
9.91	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/PreloadAllMaps.cpp File Reference	396
9.92	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/PreloadAllMaps.h File Reference	396
9.93	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/SaveEMailGame.cpp File Reference	396
9.94	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/SaveEMailGame.h File Reference	396
9.95	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/SaveGame.cpp File Reference	397
9.96	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/SaveGame.h File Reference	397
9.97	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/ShowGameOptions.cpp File Reference	397
9.98	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/ShowGameOptions.h File Reference	398
9.99	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/SpawnGameObject.cpp File Reference	398
9.100	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/SpawnGameObject.h File Reference	398
9.101	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/SpawnPlayerOnCurrent-Map.cpp File Reference	398
9.102	/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/SpawnPlayerOnCurrent-Map.h File Reference	399

9.103/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/YourTurnInformation.cpp File Reference	399
9.104/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/YourTurnInformation.h File Reference	399
9.105/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Map/ClearPath.cpp File Reference	399
9.106/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Map/ClearPath.h File Reference .	399
9.107/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Map/FindPath.cpp File Reference .	400
9.108/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Map/FindPath.h File Reference . .	400
9.109/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Map/MoveCharacter.cpp File Ref- erence	400
9.110/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Map/MoveCharacter.h File Refer- ence	400
9.111/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/ConnectTo.cpp File Reference	401
9.112/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/ConnectTo.h File Refer- ence	401
9.113/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/LoadGame.cpp File Reference	401
9.114/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/LoadGame.h File Ref- erence	402
9.115/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/PreloadWorld.cpp File Reference	402
9.116/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/PreloadWorld.h File Reference	402
9.117/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/ResumeEMailGame.cpp File Reference	403
9.118/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/ResumeEMailGame.h File Reference	403
9.119/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/SetupNewGame.cpp File Reference	403
9.120/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/SetupNewGame.h File Reference	404
9.121/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ChangePlayerIDOn- Server.cpp File Reference	404
9.122/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ChangePlayerIDOn- Server.h File Reference	404
9.123/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ClientCommand.cpp File Reference	404
9.124/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ClientCommand.h File Reference	405
9.125/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ReceiveInitialPlayer- Information.cpp File Reference	405
9.126/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ReceiveInitialPlayer- Information.h File Reference	405
9.127/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ReceiveMap.cpp File Ref- erence	405

9.128/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ReceiveMap.h File Reference	405
9.129/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ReceivePlayer.cpp File Reference	406
9.130/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ReceivePlayer.h File Reference	406
9.131/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ReceivePlayerModels.cpp File Reference	406
9.132/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ReceivePlayerModels.h File Reference	406
9.133/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/RequestMapFromServer.cpp File Reference	406
9.134/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/RequestMapFromServer.h File Reference	407
9.135/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/RequestPlayerFrom-Server.cpp File Reference	407
9.136/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/RequestPlayerFrom-Server.h File Reference	407
9.137/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/RequestPlayerModels-FromServer.cpp File Reference	407
9.138/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/RequestPlayerModels-FromServer.h File Reference	407
9.139/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/SendSpawnCharacterTo-Server.cpp File Reference	408
9.140/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/SendSpawnCharacterTo-Server.h File Reference	408
9.141/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/SendToServerEnd-Turn.cpp File Reference	408
9.142/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/SendToServerEndTurn.h File Reference	408
9.143/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/SendUpdateCharacterTo-Server.cpp File Reference	408
9.144/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/SendUpdateCharacterTo-Server.h File Reference	408
9.145/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/NetworkCommand.cpp File Reference	409
9.146/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/NetworkCommand.h File Reference	409
9.147/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/ReceiveSpawnCharacter-From.cpp File Reference	409
9.148/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/ReceiveSpawnCharacterFrom.h File Reference	409
9.149/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/ReceiveUpdateCharacter-From.cpp File Reference	409
9.150/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/ReceiveUpdateCharacterFrom.h File Reference	410

9.151/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Broadcast/Broadcast-Command.cpp File Reference	410
9.152/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Broadcast/Broadcast-Command.h File Reference	410
9.153/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Broadcast/Broadcast-SpawnCharacter.cpp File Reference	410
9.154/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Broadcast/Broadcast-SpawnCharacter.h File Reference	410
9.155/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Broadcast/Broadcast-UpdateCharacter.cpp File Reference	411
9.156/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Broadcast/Broadcast-UpdateCharacter.h File Reference	411
9.157/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/ClientDisconnected.cpp File Reference	411
9.158/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/ClientDisconnected.h File Reference	411
9.159/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/ReceiveChangePlayerID.cpp File Reference	411
9.160/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/ReceiveChangePlayerID.h File Reference	411
9.161/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/ReplyToRequestForMap.cpp File Reference	412
9.162/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/ReplyToRequestForMap.h File Reference	412
9.163/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/ReplyToRequestForPlayer.cpp File Reference	412
9.164/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/ReplyToRequestForPlayer.h File Reference	412
9.165/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/ReplyToRequestForPlayerModels.cpp File Reference	412
9.166/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/ReplyToRequestForPlayerModels.h File Reference	413
9.167/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/SendInitialPlayerInformation.cpp File Reference	413
9.168/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/SendInitialPlayerInformation.h File Reference	413
9.169/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/SendToClientNextTurn.cpp File Reference	413
9.170/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/SendToClientNextTurn.h File Reference	413
9.171/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/ServerCommand.cpp File Reference	414
9.172/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/ServerCommand.h File Reference	414
9.173/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/NotUndoable.cpp File Reference	414
9.174/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/NotUndoable.h File Reference	414
9.175/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Object/ObjectCommand.cpp File Reference	414

9.176/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Object/ObjectCommand.h File Reference	414
9.177/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Object/ShowDescription.cpp File Reference	415
9.178/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Object/ShowDescription.h File Reference	415
9.179/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/ShowAbout.cpp File Reference	415
9.180/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/ShowAbout.h File Reference	415
9.181/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Undoable.cpp File Reference	416
9.182/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Undoable.h File Reference	416
9.183/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/UseRedo.cpp File Reference	416
9.184/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/UseRedo.h File Reference	416
9.185/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/UseUndo.cpp File Reference	416
9.186/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/UseUndo.h File Reference	417
9.187/home/ryoga/git/pidgirl/pidgirl-engine/Command/Processor.cpp File Reference	417
9.188/home/ryoga/git/pidgirl/pidgirl-engine/Command/Processor.h File Reference	417
9.189/home/ryoga/git/pidgirl/pidgirl-engine/Command/UndoRedo.cpp File Reference	417
9.190/home/ryoga/git/pidgirl/pidgirl-engine/Command/UndoRedo.h File Reference	417
9.191/home/ryoga/git/pidgirl/pidgirl-engine/Common/Common.cpp File Reference	418
9.192/home/ryoga/git/pidgirl/pidgirl-engine/Common/Common.h File Reference	418
9.193/home/ryoga/git/pidgirl/pidgirl-engine/Common/FileManager.cpp File Reference	420
9.194/home/ryoga/git/pidgirl/pidgirl-engine/Common/FileManager.h File Reference	420
9.195/home/ryoga/git/pidgirl/pidgirl-engine/Common/FileNameConventions.cpp File Reference	421
9.196/home/ryoga/git/pidgirl/pidgirl-engine/Common/FileNameConventions.h File Reference	421
9.197/home/ryoga/git/pidgirl/pidgirl-engine/Common/OptionsLayout.cpp File Reference	422
9.198/home/ryoga/git/pidgirl/pidgirl-engine/Common/OptionsLayout.h File Reference	422
9.199/home/ryoga/git/pidgirl/pidgirl-engine/Common/Serialize.cpp File Reference	422
9.200/home/ryoga/git/pidgirl/pidgirl-engine/Common/Serialize.h File Reference	422
9.201/home/ryoga/git/pidgirl/pidgirl-engine/Common/YesNoCancelDialog.cpp File Reference	422
9.202/home/ryoga/git/pidgirl/pidgirl-engine/Common/YesNoCancelDialog.h File Reference	423
9.203/home/ryoga/git/pidgirl/pidgirl-engine/Editor/EditorCommon.h File Reference	423
9.204/home/ryoga/git/pidgirl/pidgirl-engine/Editor/EditorMenu.cpp File Reference	423
9.205/home/ryoga/git/pidgirl/pidgirl-engine/Editor/EditorMenu.h File Reference	424
9.206/home/ryoga/git/pidgirl/pidgirl-engine/Editor/EditorTools.cpp File Reference	424
9.207/home/ryoga/git/pidgirl/pidgirl-engine/Editor/EditorTools.h File Reference	424
9.208/home/ryoga/git/pidgirl/pidgirl-engine/Editor/MapPropertiesDialog.cpp File Reference	424
9.209/home/ryoga/git/pidgirl/pidgirl-engine/Editor/MapPropertiesDialog.h File Reference	425
9.210/home/ryoga/git/pidgirl/pidgirl-engine/Editor/Model/EditorModel.cpp File Reference	425
9.211/home/ryoga/git/pidgirl/pidgirl-engine/Editor/Model/EditorModel.h File Reference	425
9.212/home/ryoga/git/pidgirl/pidgirl-engine/Editor/Model/LocalFilePaths.cpp File Reference	426
9.213/home/ryoga/git/pidgirl/pidgirl-engine/Editor/Model/LocalFilePaths.h File Reference	426
9.214/home/ryoga/git/pidgirl/pidgirl-engine/Editor/Model/WorldModel.cpp File Reference	426
9.215/home/ryoga/git/pidgirl/pidgirl-engine/Editor/Model/WorldModel.h File Reference	426

9.216/home/ryoga/git/pidgirl/pidgirl-engine/Editor/WorldFileTree.cpp File Reference	427
9.217/home/ryoga/git/pidgirl/pidgirl-engine/Editor/WorldFileTree.h File Reference	427
9.218/home/ryoga/git/pidgirl/pidgirl-engine/Editor/WorldPropertiesDialog.cpp File Reference	427
9.219/home/ryoga/git/pidgirl/pidgirl-engine/Editor/WorldPropertiesDialog.h File Reference	428
9.220/home/ryoga/git/pidgirl/pidgirl-engine/Editor/WorldView.cpp File Reference	428
9.221/home/ryoga/git/pidgirl/pidgirl-engine/Editor/WorldView.h File Reference	428
9.222/home/ryoga/git/pidgirl/pidgirl-engine/Game/EMailSetupWidget.cpp File Reference	428
9.223/home/ryoga/git/pidgirl/pidgirl-engine/Game/EMailSetupWidget.h File Reference	429
9.224/home/ryoga/git/pidgirl/pidgirl-engine/Game/GeneralSetupWidget.cpp File Reference	429
9.225/home/ryoga/git/pidgirl/pidgirl-engine/Game/GeneralSetupWidget.h File Reference	429
9.226/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/ClientGameModel.cpp File Reference	429
9.227/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/ClientGameModel.h File Reference	430
9.228/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/EMailGameModel.cpp File Reference	430
9.229/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/EMailGameModel.h File Reference	430
9.230/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/GameModel.cpp File Reference	430
9.231/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/GameModel.h File Reference	431
9.232/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/HotseatGameModel.cpp File Reference	431
9.233/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/HotseatGameModel.h File Reference	431
9.234/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/LocalGameModel.cpp File Reference	432
9.235/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/LocalGameModel.h File Reference	432
9.236/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/MailGameInfo.cpp File Reference	432
9.237/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/MailGameInfo.h File Reference	432
9.238/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/PlayerModel.cpp File Reference	433
9.239/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/PlayerModel.h File Reference	433
9.240/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/SaveGameFile.cpp File Reference	433
9.241/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/SaveGameFile.h File Reference	433
9.242/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/ServerGameModel.cpp File Reference	434
9.243/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/ServerGameModel.h File Reference	434
9.244/home/ryoga/git/pidgirl/pidgirl-engine/Game/NetworkSetupWidget.cpp File Reference	434
9.245/home/ryoga/git/pidgirl/pidgirl-engine/Game/NetworkSetupWidget.h File Reference	434
9.246/home/ryoga/git/pidgirl/pidgirl-engine/Game/SoloHotseatSetupWidget.cpp File Reference	435
9.247/home/ryoga/git/pidgirl/pidgirl-engine/Game/SoloHotseatSetupWidget.h File Reference	435
9.248/home/ryoga/git/pidgirl/pidgirl-engine/main.cpp File Reference	435
9.248.1 Function Documentation	435
9.248.1.1 checkLocale	435
9.248.1.2 main	435
9.249/home/ryoga/git/pidgirl/pidgirl-engine/MainView/CommandAction.cpp File Reference	435
9.250/home/ryoga/git/pidgirl/pidgirl-engine/MainView/CommandAction.h File Reference	436
9.251/home/ryoga/git/pidgirl/pidgirl-engine/MainView/MainViewMap.cpp File Reference	436
9.252/home/ryoga/git/pidgirl/pidgirl-engine/MainView/MainViewMap.h File Reference	436

9.253/home/ryoga/git/pidgirl/pidgirl-engine/MainView/MainViewStack.cpp File Reference	436
9.254/home/ryoga/git/pidgirl/pidgirl-engine/MainView/MainViewStack.h File Reference	437
9.255/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MapModel.cpp File Reference	437
9.256/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MapModel.h File Reference	437
9.257/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ContextMenuMenuState.cpp File Reference	437
9.258/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ContextMenuMenuState.h File Reference	438
9.259/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ContextGameMenuState.cpp File Reference	438
9.260/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ContextGameMenuState.h File Reference	438
9.261/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/IdleEditorMouseState.cpp File Reference	438
9.262/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/IdleEditorMouseState.h File Reference	438
9.263/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/IdleGameMouseState.cpp File Reference	439
9.264/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/IdleGameMouseState.h File Reference	439
9.265/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/MouseEditorInput.cpp File Ref- erence	439
9.266/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/MouseEditorInput.h File Refer- ence	439
9.267/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/MouseGameInput.cpp File Ref- erence	440
9.268/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/MouseGameInput.h File Refer- ence	440
9.269/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/MouseInput.cpp File Reference	440
9.270/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/MouseInput.h File Reference . .	440
9.271/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/MouseState.cpp File Reference	441
9.272/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/MouseState.h File Reference . .	441
9.273/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/PlaceObjectsState.cpp File Ref- erence	441
9.274/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/PlaceObjectsState.h File Refer- ence	441
9.275/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/RemoveObjectsState.cpp File Reference	442
9.276/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/RemoveObjectsState.h File Ref- erence	442
9.277/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/TargetChosenState.cpp File Reference	442
9.278/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/TargetChosenState.h File Refer- ence	442
9.279/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/TargetConfirmedState.cpp File Reference	443

9.280/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/TargetConfirmedState.h File Reference	443
9.281/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/TargetMoveState.cpp File Reference	443
9.282/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/TargetMoveState.h File Reference	443
9.283/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/StackModel.cpp File Reference	444
9.284/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/StackModel.h File Reference	444
9.285/home/ryoga/git/pidgirl/pidgirl-engine/MainView/ObjectMenu.cpp File Reference	444
9.286/home/ryoga/git/pidgirl/pidgirl-engine/MainView/ObjectMenu.h File Reference	444
9.287/home/ryoga/git/pidgirl/pidgirl-engine/MainWindow.cpp File Reference	445
9.288/home/ryoga/git/pidgirl/pidgirl-engine/MainWindow.h File Reference	445
9.289/home/ryoga/git/pidgirl/pidgirl-engine/Network/ClientModel.cpp File Reference	446
9.290/home/ryoga/git/pidgirl/pidgirl-engine/Network/ClientModel.h File Reference	446
9.291/home/ryoga/git/pidgirl/pidgirl-engine/Network/ClientSocket.cpp File Reference	447
9.292/home/ryoga/git/pidgirl/pidgirl-engine/Network/ClientSocket.h File Reference	447
9.293/home/ryoga/git/pidgirl/pidgirl-engine/Network/NetworkModel.cpp File Reference	447
9.294/home/ryoga/git/pidgirl/pidgirl-engine/Network/NetworkModel.h File Reference	447
9.294.1 Enumeration Type Documentation	448
9.294.1.1 FromClient	448
9.294.1.2 FromServer	448
9.295/home/ryoga/git/pidgirl/pidgirl-engine/Network/ServerModel.cpp File Reference	448
9.296/home/ryoga/git/pidgirl/pidgirl-engine/Network/ServerModel.h File Reference	449
9.297/home/ryoga/git/pidgirl/pidgirl-engine/Network/WaitingForDataDialog.cpp File Reference	449
9.298/home/ryoga/git/pidgirl/pidgirl-engine/Network/WaitingForDataDialog.h File Reference	449
9.299/home/ryoga/git/pidgirl/pidgirl-engine/Object/Character/Player.cpp File Reference	449
9.300/home/ryoga/git/pidgirl/pidgirl-engine/Object/Character/Player.h File Reference	449
9.301/home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectBase.cpp File Reference	450
9.302/home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectBase.h File Reference	450
9.303/home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectCharacter.cpp File Reference	450
9.304/home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectCharacter.h File Reference	450
9.305/home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectFromID.cpp File Reference	451
9.306/home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectFromID.h File Reference	451
9.307/home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectID.cpp File Reference	451
9.308/home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectID.h File Reference	451
9.309/home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectImage.cpp File Reference	452
9.310/home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectImage.h File Reference	452
9.311/home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectItem.cpp File Reference	452
9.312/home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectItem.h File Reference	452
9.313/home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectSpecial.cpp File Reference	452
9.314/home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectSpecial.h File Reference	452

9.315/home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectTile.cpp File Reference	453
9.316/home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectTile.h File Reference	453
9.317/home/ryoga/git/pidgirl/pidgirl-engine/Object/Special/StartPosition.cpp File Reference	453
9.318/home/ryoga/git/pidgirl/pidgirl-engine/Object/Special/StartPosition.h File Reference	453
9.319/home/ryoga/git/pidgirl/pidgirl-engine/Object/Tile/BrickWall.cpp File Reference	453
9.320/home/ryoga/git/pidgirl/pidgirl-engine/Object/Tile/BrickWall.h File Reference	454
9.321/home/ryoga/git/pidgirl/pidgirl-engine/RSA/RSAMethod.cpp File Reference	454
9.322/home/ryoga/git/pidgirl/pidgirl-engine/RSA/RSAMethod.h File Reference	454

Index**455**

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

AStar	23
Commands	24
CommandProcessor	28
Common	29
GUI	30
Model	31
Game	32
State	33
Network	34
Object	35
RSA	36

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

ObjectFromID	37
PIDGIRL	
The PIDGIRL (p. 37) namespace	37

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

AplyWorldPropertiesChanges	49
AStarMap	56
AStarMapEntry	58
MapInfos	173
MouseState	196
ContextEditorMenuState	101
ContextGameMenuState	102
IdleEditorMouseState	145
IdleGameMouseState	146
PlaceObjectsState	245
RemoveObjectsState	277
TargetChosenState	350
TargetConfirmedState	351
TargetMoveState	352
ObjectID	227
PlaceObject	244
QAction	
CommandAction	96
QDialog	
MapPropertiesDialog	185
WorldPropertiesDialog	368
QFileSystemModel	
WorldModel	366
QGridLayout	
OptionsLayout	240
QImage	
ObjectImage	230
QListWidget	
List	148
QMainWindow	
MainWindow	169
QMenu	
ObjectMenu	233
QMenuBar	
EditorMenu	113
QMessageBox	
YesNoCancelDialog	371
QObject	

BaseCommand	63
Command	93
NotUndoable	213
AddToChosenList	47
ApplyMapGeometryChanges	49
ApplyWorldPropertiesChanges	51
AskForSave	52
AskForSaveMap	54
AskForSaveWorld	55
ChangeZoomFactor	75
ChooseMainTool	76
CommonCommand	97
CopyFolder	104
Quit	261
RemoveFolder	274
CopyWall	105
CutWall	110
DuplicateWorldEntry	111
GameCommand	131
ClearPath	78
ConnectTo	99
CreateCharacter	107
EndTurn	124
FindPath	127
FirstTurn	129
LoadGame	149
LoadStartMapFromIndex	154
MoveCharacter	198
NextTurn	212
PreloadAllMaps	254
PreloadWorld	256
ResumeEMailGame	292
SaveEMailGame	296
SaveGame	298
SetupNewGame	324
ShowGameOptions	329
SpawnGameObject	332
SpawnPlayerOnCurrentMap	334
YourTurnInformation	373
LoadMap	151
LoadMapByIndex	152
LoadWorld	155
MoveFile	200
NetworkCommand	201
ClientCommand	79
ChangePlayerIDOnServer	73
ReceiveInitialPlayerInformation	263
ReceiveMap	265
ReceivePlayer	266
ReceivePlayerModels	267
RequestMapFromServer	287
RequestPlayerFromServer	289
RequestPlayerModelsFromServer	290
SendSpawnCharacterToServer	307
SendToServerEndTurn	310
SendUpdateCharacterToServer	311
ReceiveSpawnCharacterFrom	269
ReceiveUpdateCharacterFrom	270

ServerCommand	315
BroadcastCommand	68
BroadcastSpawnCharacter	70
BroadcastUpdateCharacter	71
ClientDisconnected	80
ReceiveChangePlayerID	262
ReplyToRequestForMap	283
ReplyToRequestForPlayer	285
ReplyToRequestForPlayerModels	286
SendInitialPlayerInformation	306
SendToClientNextTurn	309
NewFolder	207
NewMapFile	209
NewWorld	210
ObjectCommand	226
ShowDescription	328
PasteObject	241
PlaceEditorObject	243
RemoveFromChosenList	276
RemoveWorldEntry	280
RenameWorldEntry	282
SaveMap	301
SaveMapAs	302
SaveWorld	303
SaveWorldAs	305
ShowAbout	326
UseRedo	359
UseUndo	360
Undoable	353
PlaceSpecial	246
PlaceWall	248
RemoveAllObjects	272
RemoveWall	279
ClientSocket	91
EditorModel	117
FileManager	125
GameModel	133
ClientGameModel	82
LocalGameModel	158
EMailGameModel	120
HotseatGameModel	143
ServerGameModel	317
MailGameInfo	162
MapModel	173
MouseInput	194
MouseEditorInput	186
MouseGameInput	193
NetworkModel	203
ClientModel	87
ServerModel	320
ObjectBase	216
ObjectCharacter	223
Player	250
ObjectItem	231
ObjectSpecial	234
StartPosition	348
ObjectTile	237

BrickWall	66
PlayerModel	252
Processor	257
SaveGameFile	299
StackModel	335
UndoRedo	355
WaitingForDataDialog	361
QTreeView	
WorldFileTree	363
QVector	
LocalFilePaths	157
QWidget	
EditorTools	118
EMailSetupWidget	123
GeneralSetupWidget	142
MainViewMap	164
MainViewStack	166
NetworkSetupWidget	206
SoloHotseatSetupWidget	331
WorldView	370
RSAMethod	293
Serialize	313
GameModel	133
MailGameInfo	162
MapModel	173
ObjectBase	216
PlayerModel	252
SaveGameFile	299
WorldModel	366
SpawnPlayer	334

Chapter 4

Class Index

4.1 Class List

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

AddToChosenList	
Add the current index to the chosen list	47
AplyWorldPropertiesChanges	
Apply the chosen properties to the world model	49
ApplyMapGeometryChanges	
Apply the geometrical changes to the currently loaded map model	49
ApplyWorldPropertiesChanges	51
AskForSave	
Ask if the world or the current map isn't saved, if you want to save	52
AskForSaveMap	54
AskForSaveWorld	
Ask whenever the world isn't saved, if you want to save	55
AStarMap	
The representation of the map for the A*-Algorithm	56
AStarMapEntry	58
BaseCommand	
Interface to ensure the least needed functions for the commands	63
BrickWall	
A wall of bricks	66
BroadcastCommand	
Broadcast information to all clients on the same map the information was sent from	68
BroadcastSpawnCharacter	
Broadcast the new character to be spawned	70
BroadcastUpdateCharacter	
Broadcast the character that has been changed	71
ChangePlayerIDOnServer	
Send the new player ID to the server that will replace the old one	73
ChangeZoomFactor	
Change the zoom factor for a map	75
ChooseMainTool	
Choose a Tool and set it in the editorModel	76
ClearPath	
Clear the found path to a destination	78
ClientCommand	
Interface for a command executed by network clients	79
ClientDisconnected	
Command (p. 93) is triggered whenever a client is disconnected	80

ClientGameModel	
The model representing a game ran as a client where most information is accessible remotely	82
ClientModel	
The client side implementation of the network model	87
ClientSocket	91
Command	
The base for all commands	93
CommandAction	
An action that will trigger a command	96
CommonCommand	
Interface for commonly used commands	97
ConnectTo	
Connect to a running PIDGIRL (p.37) Server	99
ContextEditorMenuState	
The state in which to show the editor context menu	101
ContextGameMenuState	
The state in which to show the game menu	102
CopyFolder	
Copy a folder recursively to another location	104
CopyWall	
Copy the wall to the clipboard	105
CreateCharacter	
Create a character and put his starting information into the game model	107
CutWall	
Copy and delete (Cut) the wall to the clipboard	110
DuplicateWorldEntry	
Make a copy of an entry in the world	111
EditorMenu	
Represents the menubar of the editor	113
EditorModel	
Represents the editor state	117
EditorTools	
The tools to be used in the editor	118
EmailGameModel	
The model representing an E-Mail game	120
EmailSetupWidget	
The widget to set up an E-Mail Game	123
EndTurn	
Things to do at the end of the turn and prepare to start the next turn	124
FileManager	
Manages files to open and close accordingly and other things	125
FindPath	
Creates a path that shows how to get from one stack to another	127
FirstTurn	
Things to do for the very first turn	129
GameCommand	
Interface for a command done in games	131
GameModel	
The model representing the game preferences and players	133
GeneralSetupWidget	
The widget to set up general options for every game before it begins	142
HotseatGameModel	
The model representing a hotseat game with one or more players	143
IdleEditorMouseState	
The editor mouse state when nothing is being done	145
IdleGameMouseState	
The game mouse state when nothing is being done	146

List	
The list for widgets with reimplemented mouse events	148
LoadGame	
Load a game for PIDGIRL (p. 37) Engine	149
LoadMap	
Load a map from a given path	151
LoadMapByIndex	
Load a map according to the given path in a model index	152
LoadStartMapFromIndex	
Load the map depending on the index given. Can be random as well	154
LoadWorld	
Choose the name of a file and load the world from this file	155
LocalFilePaths	
Locally in world folder represented file paths	157
LocalGameModel	
The model representing a local game where all the information is held locally	158
MailGameInfo	
Holds all the informations about your E-Mail Game	162
MainViewMap	
The main view showing every stack on an grid	164
MainViewStack	
Graphically represents a stack of a square in the game	166
MainWindow	
Holds all the models and GUIs and initializes them	169
MapInfos	
Information about maps	173
MapModel	
Represents the map	173
MapPropertiesDialog	185
MouseEditorInput	
The mouse input manager for the editor	186
MouseGameInput	
The mouse input manager for the game	193
MouseInput	
The base class that manages the flow of the events to be processed next	194
MouseState	
The interface used for any state concerning mouse inputs	196
MoveCharacter	
Move a character from stack to stack	198
MoveFile	
Move a file from an index to another one	200
NetworkCommand	
Command (p. 93) sent by Server or Client alike	201
NetworkModel	
The network model managing connections and interpretations of incoming messages to be converted to commands	203
NetworkSetupWidget	
The widget to set up an Network Game	206
NewFolder	
Create a new folder in the world	207
NewMapFile	
Create a new map file	209
NewWorld	
Choose the name of a file and create a new world at that location	210
NextTurn	
Begin the next turn	212
NotUndoable	
Abstract class to be inherited by Commands which are not undoable	213

ObjectBase	
The baseclass for the Objects to be placed into the game's mainView	216
ObjectCharacter	
The baseclass for the Characters to be placed into the game's mainView	223
ObjectCommand	
A command corresponding and holding to a given object	226
ObjectID	
Information and management of spawnIDs for the corresponding ObjectIDs	227
ObjectImage	
Representing and QImage which knows the path it has been loaded from	230
ObjectItem	
The baseclass for the Items to be placed into the game's mainView	231
ObjectMenu	
The menu which is created by the command actions created through the given object	233
ObjectSpecial	234
ObjectTile	
Represents a tile to be put into the mainView	237
OptionsLayout	
The layout for options to be chosen	240
PasteObject	
Paste an object from the clipboard	241
PlaceEditorObject	243
PlaceObject	
Place an object	244
PlaceObjectsState	
The state in which objects will be placed	245
PlaceSpecial	
Place a special on a chosen stack	246
PlaceWall	
Place a wall on a chosen stack	248
Player	
The player object representing the player	250
PlayerModel	
This model represents the informations about a specific player	252
PreloadAllMaps	
Preload the maps if it's a local game or a server	254
PreloadWorld	
Preload the world	256
Processor	
Process commands and manages them accordingly	257
Quit	
Command (p. 93) to quit the running application	261
ReceiveChangePlayerID	
Receive the player ID to which a player connection should change	262
ReceiveInitialPlayerInformation	
Receive the initial player information	263
ReceiveMap	
Receive and load a map from the server	265
ReceivePlayer	
Receive a player object from the server	266
ReceivePlayerModels	
Receive the requested player models	267
ReceiveSpawnCharacterFrom	
Receive the character that has been spawned	269
ReceiveUpdateCharacterFrom	
Receive the Updated Character from any given sender ID	270
RemoveAllObjects	
Remove all objects from a stackmodel	272

RemoveFolder	
Remove the a folder recursively	274
RemoveFromChosenList	
Remove entry of list model	276
RemoveObjectsState	
Remove objects from stacks	277
RemoveWall	
Remove a wall from a chosen stack	279
RemoveWorldEntry	
Remove an entry of the world	280
RenameWorldEntry	
Renames an entry of the world	282
ReplyToRequestForMap	
Reply a requested map to a player	283
ReplyToRequestForPlayer	
Reply a requested player object to a given player	285
ReplyToRequestForPlayerModels	
Reply requested player models to a player	286
RequestMapFromServer	
Request a specific map from the server	287
RequestPlayerFromServer	
Request a specific player from the server	289
RequestPlayerModelsFromServer	
Request all player models from the server	290
ResumeEMailGame	
Resume an E-Mail Game	292
RSAMethod	
The RSA Method implemented wrapped around the Crypto++ library. This is written to be completely independent of QT	293
SaveEMailGame	
Save the current EMail Game	296
SaveGame	
Save the current game and world models	298
SaveGameFile	
Represents everything needed to Serialize (p.313) and Deserialize a complete game save file	299
SaveMap	
Save a map. If it wasn't saved before, choose a place and file	301
SaveMapAs	
Save a map as a speicified file	302
SaveWorld	
Save the world. The world might not exist, so it's simply created anew	303
SaveWorldAs	
Save the current world as another world	305
SendInitialPlayerInformation	
Send the initial player informations to the player that connected	306
SendSpawnCharacterToServer	
Send the character that has been created to the server	307
SendToClientNextTurn	
Send to the next client the next turn to start the new turn for that player	309
SendToServerEndTurn	
Send End Turn to the server	310
SendUpdateCharacterToServer	
Send updated information about a character to the server	311
Serialize	
Interface for serializable classes	313
ServerCommand	
Command (p. 93) executed by a server	315

ServerGameModel	
The model representing a game that is also a server	317
ServerModel	320
SetupNewGame	
Setup a new game for PIDGIRL (p. 37) Engine	324
ShowAbout	
Show the about dialog	326
ShowDescription	
Shows the description of a given object	328
ShowGameOptions	
Show the options for the game	329
SoloHotseatSetupWidget	
The widget to set up an Solo / Hotseat Game	331
SpawnGameObject	
Generic Command (p. 93) to spawn an object	332
SpawnPlayer	
Spawn a player object	334
SpawnPlayerOnCurrentMap	334
StackModel	
A single stack (square) holding a stack of items and informations about the square which holds them	335
StartPosition	
The start position of the player	348
TargetChosenState	
The target is chosen	350
TargetConfirmedState	
The chosen state is now confirmed	351
TargetMoveState	
Move to the confirmed target	352
Undoable	
Abstract class to be inherited by Commands which are undoable	353
UndoRedo	
Manage the Undo and Redo functionalities	355
UseRedo	
Redo the last command	359
UseUndo	
Undo the last command	360
WaitingForDataDialog	361
WorldFileTree	
Showing the FileTree of an opened world	363
WorldModel	
Represents the world and all of its settings	366
WorldPropertiesDialog	
Shows the properties of a world to change it	368
WorldView	
The widget of the world Management	370
YesNoCancelDialog	
A wrapper to show yes no cancel dialogs more easily	371
YourTurnInformation	
Inform with an dialog about it being your turn	373

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

/home/ryoga/git/pidgirl/pidgirl-engine/ main.cpp	435
/home/ryoga/git/pidgirl/pidgirl-engine/ MainWindow.cpp	445
/home/ryoga/git/pidgirl/pidgirl-engine/ MainWindow.h	445
/home/ryoga/git/pidgirl/pidgirl-engine/AStar/ AStarMap.cpp	375
/home/ryoga/git/pidgirl/pidgirl-engine/AStar/ AStarMap.h	375
/home/ryoga/git/pidgirl/pidgirl-engine/AStar/ AStarMapEntry.cpp	375
/home/ryoga/git/pidgirl/pidgirl-engine/AStar/ AStarMapEntry.h	375
/home/ryoga/git/pidgirl/pidgirl-engine/Command/ Processor.cpp	417
/home/ryoga/git/pidgirl/pidgirl-engine/Command/ Processor.h	417
/home/ryoga/git/pidgirl/pidgirl-engine/Command/ UndoRedo.cpp	417
/home/ryoga/git/pidgirl/pidgirl-engine/Command/ UndoRedo.h	417
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/ BaseCommand.cpp	376
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/ BaseCommand.h	376
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/ Command.cpp	376
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/ Command.h	376
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/ NotUndoable.cpp	414
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/ NotUndoable.h	414
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/ ShowAbout.cpp	415
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/ ShowAbout.h	415
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/ Undoable.cpp	416
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/ Undoable.h	416
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/ UseRedo.cpp	416
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/ UseRedo.h	416
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/ UseUndo.cpp	416
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/ UseUndo.h	417
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/ CommonCommand.cpp	377
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/ CommonCommand.h	377
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/ CopyFolder.cpp	377
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/ CopyFolder.h	377
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/ Quit.cpp	378
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/ Quit.h	378
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/ RemoveFolder.cpp	378
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/ RemoveFolder.h	378
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/ AskForSave.cpp	379
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/ AskForSave.h	379
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/ ApplyMapGeometryChanges.cpp	379
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/ ApplyMapGeometryChanges.h	379
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/ AskForSaveMap.cpp	379

/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/AskForSaveMap.h	380
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/ChangeZoomFactor.cpp	380
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/ChangeZoomFactor.h	380
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/CopyWall.cpp	380
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/CopyWall.h	380
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/CutWall.cpp	381
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/CutWall.h	381
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/LoadMap.cpp	381
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/LoadMap.h	381
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/LoadMapByIndex.cpp	381
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/LoadMapByIndex.h	382
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/PasteObject.cpp	382
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/PasteObject.h	382
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/PlaceEditorObject.cpp	382
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/PlaceEditorObject.h	382
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/PlaceSpecial.cpp	383
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/PlaceSpecial.h	383
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/PlaceWall.cpp	383
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/PlaceWall.h	383
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/RemoveAllObjects.cpp	384
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/RemoveAllObjects.h	384
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/RemoveWall.cpp	384
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/RemoveWall.h	384
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Tools/ChooseMainTool.cpp	384
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Tools/ChooseMainTool.h	385
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/AddToChosenList.cpp	385
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/AddToChosenList.h	385
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/ApplyWorldPropertiesChanges.- cpp	385
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/ApplyWorldPropertiesChanges.- h	386
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/AskForSaveWorld.cpp	386
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/AskForSaveWorld.h	386
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/DuplicateWorldEntry.cpp	386
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/DuplicateWorldEntry.h	386
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/LoadWorld.cpp	387
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/LoadWorld.h	387
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/MoveFile.cpp	387
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/MoveFile.h	387
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/NewFolder.cpp	388
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/NewFolder.h	388
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/NewMapFile.cpp	388
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/NewMapFile.h	388
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/NewWorld.cpp	389
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/NewWorld.h	389
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/RemoveFromChosenList.cpp	389
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/RemoveFromChosenList.h	389
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/RemoveWorldEntry.cpp	390
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/RemoveWorldEntry.h	390
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/RenameWorldEntry.cpp	390
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/RenameWorldEntry.h	390
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/SaveMap.cpp	391
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/SaveMap.h	391
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/SaveMapAs.cpp	391
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/SaveMapAs.h	391
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/SaveWorld.cpp	392
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/SaveWorld.h	392
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/SaveWorldAs.cpp	392

/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/SaveWorldAs.h	392
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/GameCommand.cpp	393
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/GameCommand.h	393
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/CreateCharacter.cpp	393
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/CreateCharacter.h	393
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/EndTurn.cpp	394
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/EndTurn.h	394
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/FirstTurn.cpp	394
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/FirstTurn.h	394
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/LoadStartMapFromIndex.cpp	395
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/LoadStartMapFromIndex.h	395
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/NextTurn.cpp	395
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/NextTurn.h	395
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/PreloadAllMaps.cpp	396
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/PreloadAllMaps.h	396
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/SaveEMailGame.cpp	396
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/SaveEMailGame.h	396
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/SaveGame.cpp	397
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/SaveGame.h	397
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/ShowGameOptions.cpp	397
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/ShowGameOptions.h	398
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/SpawnGameObject.cpp	398
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/SpawnGameObject.h	398
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/SpawnPlayerOnCurrentMap.- cpp	398
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/SpawnPlayerOnCurrentMap.h	399
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/YourTurnInformation.cpp	399
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/YourTurnInformation.h	399
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Map/ClearPath.cpp	399
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Map/ClearPath.h	399
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Map/FindPath.cpp	400
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Map/FindPath.h	400
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Map/MoveCharacter.cpp	400
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Map/MoveCharacter.h	400
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/ConnectTo.cpp	401
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/ConnectTo.h	401
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/LoadGame.cpp	401
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/LoadGame.h	402
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/PreloadWorld.cpp	402
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/PreloadWorld.h	402
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/ResumeEMailGame.cpp	403
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/ResumeEMailGame.h	403
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/SetupNewGame.cpp	403
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/SetupNewGame.h	404
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/NetworkCommand.cpp	409
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/NetworkCommand.h	409
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/ReceiveSpawnCharacterFrom.cpp	409
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/ReceiveSpawnCharacterFrom.h	409
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/ReceiveUpdateCharacterFrom.cpp	409
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/ReceiveUpdateCharacterFrom.h	410
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ChangePlayerIDOnServer.cpp	404
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ChangePlayerIDOnServer.h	404
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ClientCommand.cpp	404
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ClientCommand.h	405
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ReceiveInitialPlayerInformation.- cpp	405
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ReceiveInitialPlayerInformation.- h	405

/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ ReceiveMap.cpp	405
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ ReceiveMap.h	405
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ ReceivePlayer.cpp	406
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ ReceivePlayer.h	406
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ ReceivePlayerModels.cpp	406
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ ReceivePlayerModels.h	406
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ RequestMapFromServer.cpp	406
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ RequestMapFromServer.h	407
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ RequestPlayerFromServer.- cpp	407
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ RequestPlayerFromServer.h	407
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ RequestPlayerModelsFrom- Server.cpp	407
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ RequestPlayerModelsFrom- Server.h	407
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ SendSpawnCharacterTo- Server.cpp	408
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ SendSpawnCharacterTo- Server.h	408
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ SendToServerEndTurn.cpp	408
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ SendToServerEndTurn.h	408
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ SendUpdateCharacterTo- Server.cpp	408
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ SendUpdateCharacterTo- Server.h	408
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/ ClientDisconnected.cpp	411
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/ ClientDisconnected.h	411
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/ ReceiveChangePlayerID.cpp	411
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/ ReceiveChangePlayerID.h	411
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/ ReplyToRequestForMap.cpp	412
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/ ReplyToRequestForMap.h	412
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/ ReplyToRequestForPlayer.- cpp	412
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/ ReplyToRequestForPlayer.h	412
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/ ReplyToRequestForPlayer- Models.cpp	412
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/ ReplyToRequestForPlayer- Models.h	413
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/ SendInitialPlayerInformation.- cpp	413
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/ SendInitialPlayerInformation.- h	413
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/ SendToClientNextTurn.cpp	413
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/ SendToClientNextTurn.h	413
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/ ServerCommand.cpp	414
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/ ServerCommand.h	414
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Broadcast/ BroadcastCommand.- cpp	410
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Broadcast/ BroadcastCommand.- h	410
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Broadcast/ BroadcastSpawn- Character.cpp	410
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Broadcast/ BroadcastSpawn- Character.h	410
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Broadcast/ BroadcastUpdate- Character.cpp	411
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Broadcast/ BroadcastUpdate- Character.h	411

/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Object/ ObjectCommand.cpp	414
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Object/ ObjectCommand.h	414
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Object/ ShowDescription.cpp	415
/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Object/ ShowDescription.h	415
/home/ryoga/git/pidgirl/pidgirl-engine/Common/ Common.cpp	418
/home/ryoga/git/pidgirl/pidgirl-engine/Common/ Common.h	418
/home/ryoga/git/pidgirl/pidgirl-engine/Common/ FileManager.cpp	420
/home/ryoga/git/pidgirl/pidgirl-engine/Common/ FileManager.h	420
/home/ryoga/git/pidgirl/pidgirl-engine/Common/ FileNameConventions.cpp	421
/home/ryoga/git/pidgirl/pidgirl-engine/Common/ FileNameConventions.h	421
/home/ryoga/git/pidgirl/pidgirl-engine/Common/ OptionsLayout.cpp	422
/home/ryoga/git/pidgirl/pidgirl-engine/Common/ OptionsLayout.h	422
/home/ryoga/git/pidgirl/pidgirl-engine/Common/ Serialize.cpp	422
/home/ryoga/git/pidgirl/pidgirl-engine/Common/ Serialize.h	422
/home/ryoga/git/pidgirl/pidgirl-engine/Common/ YesNoCancelDialog.cpp	422
/home/ryoga/git/pidgirl/pidgirl-engine/Common/ YesNoCancelDialog.h	423
/home/ryoga/git/pidgirl/pidgirl-engine/Editor/ EditorCommon.h	423
/home/ryoga/git/pidgirl/pidgirl-engine/Editor/ EditorMenu.cpp	423
/home/ryoga/git/pidgirl/pidgirl-engine/Editor/ EditorMenu.h	424
/home/ryoga/git/pidgirl/pidgirl-engine/Editor/ EditorTools.cpp	424
/home/ryoga/git/pidgirl/pidgirl-engine/Editor/ EditorTools.h	424
/home/ryoga/git/pidgirl/pidgirl-engine/Editor/ MapPropertiesDialog.cpp	424
/home/ryoga/git/pidgirl/pidgirl-engine/Editor/ MapPropertiesDialog.h	425
/home/ryoga/git/pidgirl/pidgirl-engine/Editor/ WorldFileTree.cpp	427
/home/ryoga/git/pidgirl/pidgirl-engine/Editor/ WorldFileTree.h	427
/home/ryoga/git/pidgirl/pidgirl-engine/Editor/ WorldPropertiesDialog.cpp	427
/home/ryoga/git/pidgirl/pidgirl-engine/Editor/ WorldPropertiesDialog.h	428
/home/ryoga/git/pidgirl/pidgirl-engine/Editor/ WorldView.cpp	428
/home/ryoga/git/pidgirl/pidgirl-engine/Editor/ WorldView.h	428
/home/ryoga/git/pidgirl/pidgirl-engine/Editor/Model/ EditorModel.cpp	425
/home/ryoga/git/pidgirl/pidgirl-engine/Editor/Model/ EditorModel.h	425
/home/ryoga/git/pidgirl/pidgirl-engine/Editor/Model/ LocalFilePaths.cpp	426
/home/ryoga/git/pidgirl/pidgirl-engine/Editor/Model/ LocalFilePaths.h	426
/home/ryoga/git/pidgirl/pidgirl-engine/Editor/Model/ WorldModel.cpp	426
/home/ryoga/git/pidgirl/pidgirl-engine/Editor/Model/ WorldModel.h	426
/home/ryoga/git/pidgirl/pidgirl-engine/Game/ EmailSetupWidget.cpp	428
/home/ryoga/git/pidgirl/pidgirl-engine/Game/ EmailSetupWidget.h	429
/home/ryoga/git/pidgirl/pidgirl-engine/Game/ GeneralSetupWidget.cpp	429
/home/ryoga/git/pidgirl/pidgirl-engine/Game/ GeneralSetupWidget.h	429
/home/ryoga/git/pidgirl/pidgirl-engine/Game/ NetworkSetupWidget.cpp	434
/home/ryoga/git/pidgirl/pidgirl-engine/Game/ NetworkSetupWidget.h	434
/home/ryoga/git/pidgirl/pidgirl-engine/Game/ SoloHotseatSetupWidget.cpp	435
/home/ryoga/git/pidgirl/pidgirl-engine/Game/ SoloHotseatSetupWidget.h	435
/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/ ClientGameModel.cpp	429
/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/ ClientGameModel.h	430
/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/ EmailGameModel.cpp	430
/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/ EmailGameModel.h	430
/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/ GameModel.cpp	430
/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/ GameModel.h	431
/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/ HotseatGameModel.cpp	431
/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/ HotseatGameModel.h	431
/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/ LocalGameModel.cpp	432
/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/ LocalGameModel.h	432
/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/ MailGameInfo.cpp	432
/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/ MailGameInfo.h	432
/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/ PlayerModel.cpp	433
/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/ PlayerModel.h	433
/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/ SaveGameFile.cpp	433

/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/ SaveGameFile.h	433
/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/ ServerGameModel.cpp	434
/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/ ServerGameModel.h	434
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/ CommandAction.cpp	435
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/ CommandAction.h	436
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/ MainViewMap.cpp	436
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/ MainViewMap.h	436
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/ MainViewStack.cpp	436
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/ MainViewStack.h	437
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/ ObjectMenu.cpp	444
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/ ObjectMenu.h	444
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/ MapModel.cpp	437
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/ MapModel.h	437
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/ StackModel.cpp	444
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/ StackModel.h	444
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ ContextEditorMenuState.cpp	437
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ ContextEditorMenuState.h	438
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ ContextGameMenuState.cpp	438
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ ContextGameMenuState.h	438
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ IdleEditorMouseState.cpp	438
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ IdleEditorMouseState.h	438
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ IdleGameMouseState.cpp	439
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ IdleGameMouseState.h	439
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ MouseEditorInput.cpp	439
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ MouseEditorInput.h	439
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ MouseGameInput.cpp	440
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ MouseGameInput.h	440
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ MouseInput.cpp	440
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ MouseInput.h	440
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ MouseState.cpp	441
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ MouseState.h	441
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ PlaceObjectsState.cpp	441
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ PlaceObjectsState.h	441
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ RemoveObjectsState.cpp	442
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ RemoveObjectsState.h	442
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ TargetChosenState.cpp	442
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ TargetChosenState.h	442
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ TargetConfirmedState.cpp	443
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ TargetConfirmedState.h	443
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ TargetMoveState.cpp	443
/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ TargetMoveState.h	443
/home/ryoga/git/pidgirl/pidgirl-engine/Network/ ClientModel.cpp	446
/home/ryoga/git/pidgirl/pidgirl-engine/Network/ ClientModel.h	446
/home/ryoga/git/pidgirl/pidgirl-engine/Network/ ClientSocket.cpp	447
/home/ryoga/git/pidgirl/pidgirl-engine/Network/ ClientSocket.h	447
/home/ryoga/git/pidgirl/pidgirl-engine/Network/ NetworkModel.cpp	447
/home/ryoga/git/pidgirl/pidgirl-engine/Network/ NetworkModel.h	447
/home/ryoga/git/pidgirl/pidgirl-engine/Network/ ServerModel.cpp	448
/home/ryoga/git/pidgirl/pidgirl-engine/Network/ ServerModel.h	449
/home/ryoga/git/pidgirl/pidgirl-engine/Network/ WaitingForDataDialog.cpp	449
/home/ryoga/git/pidgirl/pidgirl-engine/Network/ WaitingForDataDialog.h	449
/home/ryoga/git/pidgirl/pidgirl-engine/Object/ ObjectBase.cpp	450
/home/ryoga/git/pidgirl/pidgirl-engine/Object/ ObjectBase.h	450
/home/ryoga/git/pidgirl/pidgirl-engine/Object/ ObjectCharacter.cpp	450
/home/ryoga/git/pidgirl/pidgirl-engine/Object/ ObjectCharacter.h	450
/home/ryoga/git/pidgirl/pidgirl-engine/Object/ ObjectFromID.cpp	451
/home/ryoga/git/pidgirl/pidgirl-engine/Object/ ObjectFromID.h	451
/home/ryoga/git/pidgirl/pidgirl-engine/Object/ ObjectID.cpp	451

/home/ryoga/git/pidgirl/pidgirl-engine/Object/ ObjectID.h	451
/home/ryoga/git/pidgirl/pidgirl-engine/Object/ ObjectImage.cpp	452
/home/ryoga/git/pidgirl/pidgirl-engine/Object/ ObjectImage.h	452
/home/ryoga/git/pidgirl/pidgirl-engine/Object/ ObjectItem.cpp	452
/home/ryoga/git/pidgirl/pidgirl-engine/Object/ ObjectItem.h	452
/home/ryoga/git/pidgirl/pidgirl-engine/Object/ ObjectSpecial.cpp	452
/home/ryoga/git/pidgirl/pidgirl-engine/Object/ ObjectSpecial.h	452
/home/ryoga/git/pidgirl/pidgirl-engine/Object/ ObjectTile.cpp	453
/home/ryoga/git/pidgirl/pidgirl-engine/Object/ ObjectTile.h	453
/home/ryoga/git/pidgirl/pidgirl-engine/Object/Character/ Player.cpp	449
/home/ryoga/git/pidgirl/pidgirl-engine/Object/Character/ Player.h	449
/home/ryoga/git/pidgirl/pidgirl-engine/Object/Special/ StartPosition.cpp	453
/home/ryoga/git/pidgirl/pidgirl-engine/Object/Special/ StartPosition.h	453
/home/ryoga/git/pidgirl/pidgirl-engine/Object/Tile/ BrickWall.cpp	453
/home/ryoga/git/pidgirl/pidgirl-engine/Object/Tile/ BrickWall.h	454
/home/ryoga/git/pidgirl/pidgirl-engine/RSA/ RSAMethod.cpp	454
/home/ryoga/git/pidgirl/pidgirl-engine/RSA/ RSAMethod.h	454

Chapter 6

Module Documentation

6.1 AStar

Classes

- class **AStarMap**
The representation of the map for the A-Algorithm.*
- class **FindPath**
Creates a path that shows how to get from one stack to another.

6.1.1 Detailed Description

6.2 Commands

Classes

- class **BaseCommand**
Interface to ensure the least needed functions for the commands.
- class **Command**
The base for all commands.
- class **CommonCommand**
Interface for commonly used commands.
- class **CopyFolder**
Copy a folder recursively to another location.
- class **Quit**
***Command** (p. 93) to quit the running application.*
- class **RemoveFolder**
Remove the a folder recursively.
- class **AskForSave**
Ask if the world or the current map isn't saved, if you want to save.
- class **ApplyMapGeometryChanges**
Apply the geometrical changes to the currently loaded map model.
- class **ChangeZoomFactor**
Change the zoom factor for a map.
- class **CopyWall**
Copy the wall to the clipboard.
- class **CutWall**
Copy and delete (Cut) the wall to the clipboard.
- class **LoadMap**
Load a map from a given path.
- class **LoadMapByIndex**
Load a map according to the given path in a model index.
- class **PasteObject**
Paste an object from the clipboard.
- class **PlaceObject**
Place an object.
- class **PlaceSpecial**
Place a special on a chosen stack.
- class **PlaceWall**
Place a wall on a chosen stack.
- class **RemoveAllObjects**
Remove all objects from a stackmodel.
- class **RemoveWall**
Remove a wall from a chosen stack.
- class **ChooseMainTool**
Choose a Tool and set it in the editorModel.
- class **AddToChosenList**
Add the current index to the chosen list.
- class **ApplyWorldPropertiesChanges**
Apply the chosen properties to the world model.
- class **AskForSaveWorld**
Ask whenever the world isn't saved, if you want to save.
- class **DuplicateWorldEntry**

- Make a copy of an entry in the world.*

 - class **LoadWorld**

Choose the name of a file and load the world from this file.
- class **MoveFile**

Move a file from an index to another one.
- class **NewFolder**

Create a new folder in the world.
- class **NewMapFile**

Create a new map file.
- class **NewWorld**

Choose the name of a file and create a new world at that location.
- class **RemoveFromChosenList**

Remove entry of list model.
- class **RemoveWorldEntry**

Remove an entry of the world.
- class **RenameWorldEntry**

Renames an entry of the world.
- class **SaveMap**

Save a map. If it wasn't saved before, choose a place and file.
- class **SaveMapAs**

Save a map as a specified file.
- class **SaveWorld**

Save the world. The world might not exist, so it's simply created anew.
- class **SaveWorldAs**

Save the current world as another world.
- class **GameCommand**

Interface for a command done in games.
- class **CreateCharacter**

Create a character and put his starting information into the game model.
- class **EndTurn**

Things to do at the end of the turn and prepare to start the next turn.
- class **FirstTurn**

Things to do for the very first turn.
- class **LoadStartMapFromIndex**

Load the map depending on the index given. Can be random as well.
- class **NextTurn**

Begin the next turn.
- class **PreloadAllMaps**

Preload the maps if it's a local game or a server.
- class **SaveEMailGame**

Save the current EMail Game.
- class **SaveGame**

Save the current game and world models.
- class **ShowGameOptions**

Show the options for the game.
- class **SpawnGameObject**

*Generic **Command** (p. 93) to spawn an object.*
- class **SpawnPlayer**

Spawn a player object.
- class **YourTurnInformation**

Inform with an dialog about it being your turn.

- class **ClearPath**
Clear the found path to a destination.
- class **FindPath**
Creates a path that shows how to get from one stack to another.
- class **MoveCharacter**
Move a character from stack to stack.
- class **ConnectTo**
*Connect to a running **PIDGIRL** (p. 37) Server.*
- class **LoadGame**
*Load a game for **PIDGIRL** (p. 37) Engine.*
- class **PreloadWorld**
Preload the world.
- class **ResumeEMailGame**
Resume an E-Mail Game.
- class **SetupNewGame**
*Setup a new game for **PIDGIRL** (p. 37) Engine.*
- class **ChangePlayerIDOnServer**
Send the new player ID to the server that will replace the old one.
- class **ClientCommand**
Interface for a command executed by network clients.
- class **ReceiveInitialPlayerInformation**
Receive the initial player information.
- class **ReceiveMap**
Receive and load a map from the server.
- class **ReceivePlayer**
Receive a player object from the server.
- class **ReceivePlayerModels**
Receive the requested player models.
- class **RequestMapFromServer**
Request a specific map from the server.
- class **RequestPlayerFromServer**
Request a specific player from the server.
- class **RequestPlayerModelsFromServer**
Request all player models from the server.
- class **SendSpawnCharacterToServer**
Send the character that has been created to the server.
- class **SendToServerEndTurn**
Send End Turn to the server.
- class **SendUpdateCharacterToServer**
Send updated information about a character to the server.
- class **NetworkCommand**
***Command** (p. 93) sent by Server or Client alike.*
- class **ReceiveSpawnCharacterFrom**
Receive the character that has been spawned.
- class **ReceiveUpdateCharacterFrom**
Receive the Updated Character from any given sender ID.
- class **BroadcastCommand**
Broadcast information to all clients on the same map the information was sent from.
- class **BroadcastSpawnCharacter**
Broadcast the new character to be spawned.
- class **BroadcastUpdateCharacter**

- Broadcast the character that has been changed.*
- class **ClientDisconnected**
 - Command** (p. 93) is triggered whenever a client is disconnected.*
- class **ReceiveChangePlayerID**
 - Receive the player ID to which a player connection should change.*
- class **ReplyToRequestForMap**
 - Reply a requested map to a player.*
- class **ReplyToRequestForPlayer**
 - Reply a requested player object to a given player.*
- class **ReplyToRequestForPlayerModels**
 - Reply requested player models to a player.*
- class **SendInitialPlayerInformation**
 - Send the initial player informations to the player that connected.*
- class **SendToClientNextTurn**
 - Send to the next client the next turn to start the new turn for that player.*
- class **ServerCommand**
 - Command** (p. 93) executed by a server.*
- class **NotUndoable**
 - Abstract class to be inherited by Commands which are not undoable.*
- class **ObjectCommand**
 - A command corresponding and holding to a given object.*
- class **ShowDescription**
 - Shows the description of a given object.*
- class **ShowAbout**
 - Show the about dialog.*
- class **Undoable**
 - Abstract class to be inherited by Commands which are undoable.*
- class **UseRedo**
 - Redo the last command.*
- class **UseUndo**
 - Undo the last command.*

6.2.1 Detailed Description

6.3 CommandProcessor

Classes

- class **Processor**
Process commands and manages them accordingly.
- class **UndoRedo**
Manage the Undo and Redo functionalities.

6.3.1 Detailed Description

6.4 Common

Classes

- class **FileManager**
Manages files to open and close accordingly and other things.
- class **List**
The list for widgets with reimplemented mouse events.
- class **OptionsLayout**
The layout for options to be chosen.
- class **Serialize**
Interface for serializable classes.
- class **YesNoCancelDialog**
A wrapper to show yes no cancel dialogs more easily.

6.4.1 Detailed Description

6.5 GUI

Classes

- class **EditorMenu**
Represents the menubar of the editor.
- class **EditorTools**
The tools to be used in the editor.
- class **MapInfos**
Information about maps.
- class **WorldFileTree**
Showing the FileTree of an opened world.
- class **WorldPropertiesDialog**
Shows the properties of a world to change it.
- class **WorldView**
The widget of the world Management.
- class **E-MailSetupWidget**
The widget to set up an E-Mail Game.
- class **GeneralSetupWidget**
The widget to set up general options for every game before it begins.
- class **NetworkSetupWidget**
The widget to set up an Network Game.
- class **SoloHotseatSetupWidget**
The widget to set up an Solo / Hotseat Game.
- class **CommandAction**
An action that will trigger a command.
- class **MainViewMap**
The main view showing every stack on an grid.
- class **MainViewStack**
Graphically represents a stack of a square in the game.
- class **ObjectMenu**
The menu which is created by the command actions created through the given object.
- class **MainWindow**
Holds all the models and GUIs and initializes them.

6.5.1 Detailed Description

6.6 Model

Classes

- class **EditorModel**
Represents the editor state.
- class **LocalFilePaths**
Locally in world folder represented file paths.
- class **WorldModel**
Represents the world and all of its settings.
- class **MapModel**
Represents the map.
- class **StackModel**
A single stack (square) holding a stack of items and informations about the square which holds them.

6.6.1 Detailed Description

6.7 Game

Classes

- class **ClientGameModel**
The model representing a game ran as a client where most information is accessible remotely.
- class **EMailGameModel**
The model representing an E-Mail game.
- class **GameModel**
The model representing the game preferences and players.
- class **HotseatGameModel**
The model representing a hotseat game with one or more players.
- class **LocalGameModel**
The model representing a local game where all the information is held locally.
- class **MailGameInfo**
Holds all the informations about your E-Mail Game.
- class **PlayerModel**
This model represents the informations about a specific player.
- class **SaveGameFile**
*Represents everything needed to **Serialize** (p. 313) and Deserialize a complete game save file.*
- class **ServerGameModel**
The model representing a game that is also a server.

6.7.1 Detailed Description

6.8 State

Classes

- class **ContextEditorMenuState**
The state in which to show the editor context menu.
- class **ContextGameMenuState**
The state in which to show the game menu.
- class **IdleEditorMouseState**
The editor mouse state when nothing is being done.
- class **IdleGameMouseState**
The game mouse state when nothing is being done.
- class **MouseEditorInput**
The mouse input manager for the editor.
- class **MouseGameInput**
The mouse input manager for the game.
- class **MouseInput**
The base class that manages the flow of the events to be processed next.
- class **MouseState**
The interface used for any state concerning mouse inputs.
- class **PlaceObjectsState**
The state in which objects will be placed.
- class **RemoveObjectsState**
Remove objects from stacks.
- class **TargetChosenState**
The target is chosen.
- class **TargetConfirmedState**
The chosen state is now confirmed.
- class **TargetMoveState**
Move to the confirmed target.

6.8.1 Detailed Description

6.9 Network

Classes

- class **ClientModel**
The client side implementation of the network model.
- class **NetworkModel**
The network model managing connections and interpretations of incoming messages to be converted to commands.

6.9.1 Detailed Description

6.10 Object

Classes

- class **Player**
The player object representing the player.
- class **ObjectBase**
The baseclass for the Objects to be placed into the game's mainView.
- class **ObjectCharacter**
The baseclass for the Characters to be placed into the game's mainView.
- class **ObjectID**
Information and management of spawnIDs for the corresponding ObjectIDs.
- class **ObjectImage**
Representing and QImage which knows the path it has been loaded from.
- class **ObjectItem**
The baseclass for the Items to be placed into the game's mainView.
- class **ObjectTile**
Represents a tile to be put into the mainView.
- class **StartPosition**
The start position of the player.
- class **BrickWall**
A wall of bricks.

6.10.1 Detailed Description

6.11 RSA

Classes

- class **RSAMethod**

The RSA Method implemented wrapped around the Crypto++ library. This is written to be completely independent of QT.

6.11.1 Detailed Description

Chapter 7

Namespace Documentation

7.1 ObjectFromID Namespace Reference

Functions

- **ObjectCharacter * objectFrom (ObjectID::CharacterID id, QObject *parent=0)**
Create object from an enum.
- **ObjectItem * objectFrom (ObjectID::ItemID id, QObject *parent=0)**
- **ObjectSpecial * objectFrom (ObjectID::SpecialID id, QObject *parent=0)**
- **ObjectTile * objectFrom (ObjectID::TileID id, QObject *parent=0)**

7.1.1 Function Documentation

7.1.1.1 ObjectCharacter * ObjectFromID::objectFrom (ObjectID::CharacterID *id*, QObject * *parent* = 0)

Create object from an enum.

Parameters

<i>id</i>	The id the character gets.
<i>parent</i>	The parent to be set for that object.

Returns

Returns the created object.

7.1.1.2 ObjectItem * ObjectFromID::objectFrom (ObjectID::ItemID *id*, QObject * *parent* = 0)

7.1.1.3 ObjectSpecial * ObjectFromID::objectFrom (ObjectID::SpecialID *id*, QObject * *parent* = 0)

7.1.1.4 ObjectTile * ObjectFromID::objectFrom (ObjectID::TileID *id*, QObject * *parent* = 0)

7.2 PIDGIRL Namespace Reference

The **PIDGIRL** (p. 37) namespace.

Enumerations

- enum **MainToolButton** { **START_POSITION** =0, **BRICKWALL** }

Represents every maintool button.

- enum **GameType** { **HOTSEAT** = 0, **NETWORK**, **EMAIL** }

Functions

- void **showUsage** ()
*Show the usage of **PIDGIRL** (p. 37).*
- void **writeLog** (const QString &s)
Write a log output in the outputbuffer, so 6 characters are left to be written.
- void **writeLogDone** (const QString &s="[DONE]")
Write a certain string.
- void **writeLogFail** (const QString &error="")
Something failed and a given error is written after [FAIL] is written.
- void **newSeed** ()
Create a new seed for random numbers.
- unsigned int **nextID** (unsigned int id, const unsigned int lowestID=0)
Get the next higher possible ID.

Variables

- const int **MAX_UNDO_REDO_SIZE** = 5
- const QString **PIDGIRL_VERSION** = "0.5"
Version Number.
- const int **SECONDS_UNTIL_RETRYING_FILE_ACCESS** = 1
Specifies the milliseconds until a file access should be retried. 1 sec = 1 millisecond.
- const int **MAX_FILE_ACCESS_RETRIES** = 5
Specifies how many times a fileaccess should be retried until it's canceled.
- const int **MIN_HORIZONTAL_SQUARE_COUNT** = 3
The minimum horizontal square count.
- const int **MAX_HORIZONTAL_SQUARE_COUNT** = 30
The maximum horizontal square count.
- const int **STANDARD_HORIZONTAL_SQUARE_COUNT** = 10
The standard horizontal square count.
- const int **MIN_VERTICAL_SQUARE_COUNT** = 3
The minimum vertical square count.
- const int **MAX_VERTICAL_SQUARE_COUNT** = 20
The maximum vertical square count.
- const int **STANDARD_VERTICAL_SQUARE_COUNT** = 10
The standard vertical square count.
- const int **MIN_HEIGHT** = 1
The minimum height.
- const int **MAX_HEIGHT** = 30
The maximum height.
- const int **STANDARD_HEIGHT** = 1
The standard height.
- const int **MIN_ZOOM_FACTOR** = 1
The minimum zoomfactor for the square size.
- const int **MAX_ZOOM_FACTOR** = 3
The maximum zoomfactor for the square size.
- const int **STANDARD_ZOOM_FACTOR** = 1

- The standard zoomfactor for the square size.*

 - const int **MIN_SQUARE_SIZE** = **MIN_ZOOM_FACTOR***30
- The minimum size of a square in SIZE x SIZE pixel.*

 - const int **MAX_SQUARE_SIZE** = **MAX_ZOOM_FACTOR*****MIN_SQUARE_SIZE**
- The maximum size of a square in SIZE x SIZE pixel.*

 - const int **STANDARD_SQUARE_SIZE** = **MIN_SQUARE_SIZE**
- The standard size of a square in SIZE x SIZE pixel.*

 - const int **MIN_GRID_LINE_SIZE** = 1
- The minimum size of a grid line.*

 - const int **MAX_GRID_LINE_SIZE** = 8
- The maximum size of a grid line.*

 - const int **STANDARD_GRID_LINE_SIZE** = 1
- The standard size of a grid line.*

 - const int **MIN_PLAYER_NAME_LENGTH** = 4
- The minimum of characters that has to be used when naming a character.*

 - const int **MAX_PLAYER_NAME_LENGTH** = 30
- The maximum of characters that can be used when naming a character.*

 - const QString **NAME_REG_EXP_STRING** = QString("\\s\\w{%1,%2}").arg(PIDGIRL::MIN_PLAYER_NAME_LENGTH).arg(PIDGIRL::MAX_PLAYER_NAME_LENGTH)
- Regular expression for names the player can enter.*

 - const QString **IP_ADDRESS_REG_EXP_STRING** = QString("[0-9]|[a-z][A-Z])([0-9]|-|.|[a-z][A-Z])*")
- Regular experssion for IP Addresses. According to RFC952 and RFC1123.*

 - const int **SEND_DATA_RETRIES** = 3
- The amount of retries before canceling the sending process.*

 - const unsigned int **SERVER_PLAYER_ID** = 0
- Represents the player ID used by the server.*

 - const int **MAX_EMAIL_PLAYER_COUNT** = 128
- The initial maximum amount of players in an E-Mail Game.*

 - const QString **MAP_SUFFIX** = ".tmap"
- The map suffix.*

 - const QString **WORLD_FOLDER_NAME** = "worlds"
- The World Folder Name.*

 - const QString **WORLD_SUFFIX** = ".tworld"
- The world suffix.*

 - const QString **GAME_INFO_SUFFIX** = ".tgameinfo"
- The game information suffix.*

 - const QString **SAVE_SUFFIX** = ".tsave"
- The save suffix.*

 - const QString **SAVE_FOLDER_NAME** = "saves"
- The Save Folder Name.*

 - const QString **EMAIL_SUFFIX** = ".tmail"
- The email file suffix.*

 - const QString **EMAIL_FOLDER_NAME** = "email"
- The EMail Folder Name.*

 - const QString **RSA_FOLDER_NAME** = "RSA-Encryption"
- The RSA Folder Name.*

 - const QString **RSA_PRIVATE_KEY_FILE_NAME** = "private.key"
- The RSA Private Key File Name.*

 - const QString **RSA_PUBLIC_KEY_FILE_NAME** = "public.key"
- The RSA Public Key File Name.*

 - const QString **RSA_PLAINTEXT_FILE_NAME** = "plain.text"

The Plaintext File Name.

- const int **EDITOR_VIEW_WIDTH** = 200

The width of the Editor View.

- const int **EDITOR_VIEW_HEIGHT** = 250

The minimal height of the Editor View.

- const int **TOOL_BUTTON_COUNT_IN_ROW** = 4

The amount of tool buttons per row.

- bool **done** = false

*Needed for canceling the program before the application has been started yet (Before `app.exec()` in `main()` (p. 435)).
Only set with the **Quit** (p. 261) **Command** (p. 93).*

7.2.1 Detailed Description

The **PIDGIRL** (p. 37) namespace.

7.2.2 Enumeration Type Documentation

7.2.2.1 enum PIDGIRL::GameType

Enumerator

HOTSEAT

NETWORK

EMAIL

7.2.2.2 enum PIDGIRL::MainToolButton

Represents every maintool button.

Enumerator

START_POSITION

BRICKWALL

7.2.3 Function Documentation

7.2.3.1 void PIDGIRL::newSeed () [inline]

Create a new seed for random numbers.

7.2.3.2 unsigned int PIDGIRL::nextID (unsigned int *id*, const unsigned int *lowestID* = 0) [inline]

Get the next higher possible ID.

7.2.3.3 void PIDGIRL::showUsage () [inline]

Show the usage of **PIDGIRL** (p. 37).

7.2.3.4 void PIDGIRL::writeLog (const QString & *s*) [inline]

Write a log output in the outputbuffer, so 6 characters are left to be written.

Parameters

<i>s</i>	
----------	--

7.2.3.5 void PIDGIRL::writeLogDone (const QString & *s* = " [DONE] ") [inline]

Write a certain string.

Parameters

<i>s</i>	Is [DONE] if nothing is given.
----------	--------------------------------

7.2.3.6 void PIDGIRL::writeLogFail (const QString & *error* = " ") [inline]

Something failed and a given error is written after [FAIL] is written.

Parameters

<i>error</i>	The error to be written out.
--------------	------------------------------

7.2.4 Variable Documentation

7.2.4.1 bool PIDGIRL::done = false

Needed for canceling the program before the application has been started yet (Before app.exec() in **main()** (p. 435)). Only set with the **Quit** (p. 261) **Command** (p. 93).

7.2.4.2 const int PIDGIRL::EDITOR_VIEW_HEIGHT = 250

The minimal height of the Editor View.

7.2.4.3 const int PIDGIRL::EDITOR_VIEW_WIDTH = 200

The width of the Editor View.

7.2.4.4 const QString PIDGIRL::EMAIL_FOLDER_NAME = "email"

The EMail Folder Name.

7.2.4.5 const QString PIDGIRL::EMAIL_SUFFIX = ".tmail"

The email file suffix.

7.2.4.6 const QString PIDGIRL::GAME_INFO_SUFFIX = ".tgameinfo"

The game information suffix.

7.2.4.7 const QString PIDGIRL::IP_ADDRESS_REG_EXP_STRING = QString("([0-9]|[a-z]|[A-Z])([0-9]|-|.|[a-z]|[A-Z])*")

Regular experssion for IP Addresses. According to RFC952 and RFC1123.

7.2.4.8 `const QString PIDGIRL::MAP_SUFFIX = ".tmap"`

The map suffix.

7.2.4.9 `const int PIDGIRL::MAX_EMAIL_PLAYER_COUNT = 128`

The initial maximum amount of players in an E-Mail Game.

7.2.4.10 `const int PIDGIRL::MAX_FILE_ACCESS_RETRIES = 5`

Specifies how many times a fileaccess should be retried until it's canceled.

7.2.4.11 `const int PIDGIRL::MAX_GRID_LINE_SIZE = 8`

The maximum size of a grid line.

7.2.4.12 `const int PIDGIRL::MAX_HEIGHT = 30`

The maximum height.

7.2.4.13 `const int PIDGIRL::MAX_HORIZONTAL_SQUARE_COUNT = 30`

The maximum horizontal square count.

7.2.4.14 `const int PIDGIRL::MAX_PLAYER_NAME_LENGTH = 30`

The maximum of characters that can be used when naming a character.

7.2.4.15 `const int PIDGIRL::MAX_SQUARE_SIZE = MAX_ZOOM_FACTOR*MIN_SQUARE_SIZE`

The maximum size of a square in SIZE x SIZE pixel.

7.2.4.16 `const int PIDGIRL::MAX_UNDO_REDO_SIZE = 5`

7.2.4.17 `const int PIDGIRL::MAX_VERTICAL_SQUARE_COUNT = 20`

The maximum vertical square count.

7.2.4.18 `const int PIDGIRL::MAX_ZOOM_FACTOR = 3`

The maximum zoomfactor for the square size.

7.2.4.19 `const int PIDGIRL::MIN_GRID_LINE_SIZE = 1`

The minimum size of a grid line.

7.2.4.20 `const int PIDGIRL::MIN_HEIGHT = 1`

The minimum height.

7.2.4.21 `const int PIDGIRL::MIN_HORIZONTAL_SQUARE_COUNT = 3`

The minimum horizontal square count.

7.2.4.22 `const int PIDGIRL::MIN_PLAYER_NAME_LENGTH = 4`

The minimum of characters that has to be used when naming a character.

7.2.4.23 `const int PIDGIRL::MIN_SQUARE_SIZE = MIN_ZOOM_FACTOR*30`

The minimum size of a square in SIZE x SIZE pixel.

7.2.4.24 `const int PIDGIRL::MIN_VERTICAL_SQUARE_COUNT = 3`

The minimum vertical square count.

7.2.4.25 `const int PIDGIRL::MIN_ZOOM_FACTOR = 1`

The minimum zoomfactor for the square size.

7.2.4.26 `const QString PIDGIRL::NAME_REG_EXP_STRING = QString("\\s\\w){%1,%2}").arg(PIDGIRL::MIN_PLAYER_NAME_LENGTH).arg(PIDGIRL::MAX_PLAYER_NAME_LENGTH)`

Regular expression for names the player can enter.

7.2.4.27 `const QString PIDGIRL::PIDGIRL_VERSION = "0.5"`

Version Number.

7.2.4.28 `const QString PIDGIRL::RSA_FOLDER_NAME = "RSA-Encryption"`

The RSA Folder Name.

7.2.4.29 `const QString PIDGIRL::RSA_PLAINTEXT_FILE_NAME = "plain.text"`

The Plaintext File Name.

7.2.4.30 `const QString PIDGIRL::RSA_PRIVATE_KEY_FILE_NAME = "private.key"`

The RSA Private Key File Name.

7.2.4.31 `const QString PIDGIRL::RSA_PUBLIC_KEY_FILE_NAME = "public.key"`

The RSA Public Key File Name.

7.2.4.32 `const QString PIDGIRL::SAVE_FOLDER_NAME = "saves"`

The Save Folder Name.

7.2.4.33 `const QString PIDGIRL::SAVE_SUFFIX = ".tsave"`

The save suffix.

7.2.4.34 `const int PIDGIRL::SECONDS_UNTIL_RETRYING_FILE_ACCESS = 1`

Specifies the milliseconds until a file access should be retried. 1 sec = 1 millisecond.

7.2.4.35 `const int PIDGIRL::SEND_DATA_RETRIES = 3`

The amount of retries before canceling the sending process.

7.2.4.36 `const unsigned int PIDGIRL::SERVER_PLAYER_ID = 0`

Represents the player ID used by the server.

7.2.4.37 `const int PIDGIRL::STANDARD_GRID_LINE_SIZE = 1`

The standard size of a grid line.

7.2.4.38 `const int PIDGIRL::STANDARD_HEIGHT = 1`

The standard height.

7.2.4.39 `const int PIDGIRL::STANDARD_HORIZONTAL_SQUARE_COUNT = 10`

The standard horizontal square count.

7.2.4.40 `const int PIDGIRL::STANDARD_SQUARE_SIZE = MIN_SQUARE_SIZE`

The standard size of a square in SIZE x SIZE pixel.

7.2.4.41 `const int PIDGIRL::STANDARD_VERTICAL_SQUARE_COUNT = 10`

The standard vertical square count.

7.2.4.42 `const int PIDGIRL::STANDARD_ZOOM_FACTOR = 1`

The standard zoomfactor for the square size.

7.2.4.43 `const int PIDGIRL::TOOL_BUTTON_COUNT_IN_ROW = 4`

The amount of tool buttons per row.

7.2.4.44 `const QString PIDGIRL::WORLD_FOLDER_NAME = "worlds"`

The World Folder Name.

7.2.4.45 `const QString PIDGIRL::WORLD_SUFFIX = ".tworld"`

The world suffix.

Chapter 8

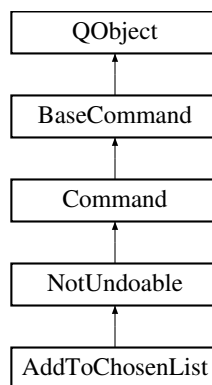
Class Documentation

8.1 AddToChosenList Class Reference

Add the current index to the chosen list.

```
#include <AddToChosenList.h>
```

Inheritance diagram for AddToChosenList:



Public Member Functions

- **AddToChosenList** (QFileSystemModel ***availableFileSystemModel**, QStandardItemModel ***chosenListModel**, const QModelIndex &**currentIndex**, QObject *parent=0)
- virtual bool **execute** ()

Executes the command.

Private Attributes

- QFileSystemModel * **availableFileSystemModel**
The file system model of which to get the path from.
- QStandardItemModel * **chosenListModel**
The list model representing the already chosen entries.
- QModelIndex **currentIndex**
The current index to be added to the chosen list model.

Additional Inherited Members

8.1.1 Detailed Description

Add the current index to the chosen list.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.1.2 Constructor & Destructor Documentation

8.1.2.1 **AddToChosenList::AddToChosenList** (*QFileSystemModel* * *availableFileSystemModel*, *QStandardItemModel* * *chosenListModel*, const *QModelIndex* & *currentIndex*, *QObject* * *parent* = 0)

8.1.3 Member Function Documentation

8.1.3.1 **bool AddToChosenList::execute** () [virtual]

Executes the command.

Returns

True, if successful.

Implements **NotUndoable** (p. 215).

8.1.4 Member Data Documentation

8.1.4.1 **QFileSystemModel*** **AddToChosenList::availableFileSystemModel** [private]

The file system model of which to get the path from.

8.1.4.2 **QStandardItemModel*** **AddToChosenList::chosenListModel** [private]

The list model representing the already chosen entries.

8.1.4.3 **QModelIndex** **AddToChosenList::currentIndex** [private]

The current index to be added to the chosen list model.

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

- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/**AddToChosenList.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/**AddToChosenList.cpp**

8.2 AplyWorldPropertiesChanges Class Reference

Apply the chosen properties to the world model.

```
#include <AplyWorldPropertiesChanges.h>
```

8.2.1 Detailed Description

Apply the chosen properties to the world model.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

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

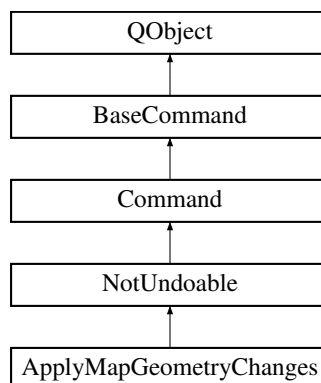
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/**AplyWorldPropertiesChanges.h**

8.3 ApplyMapGeometryChanges Class Reference

Apply the geometrical changes to the currently loaded map model.

```
#include <ApplyMapGeometryChanges.h>
```

Inheritance diagram for ApplyMapGeometryChanges:



Public Member Functions

- **ApplyMapGeometryChanges** (int **horizontalNewValue**, int **horizontalDirection**, int **verticalNewValue**, int **verticalDirection**, int **maxHeightNewValue**, int **maxHeightDirection**, QObject *parent=0)
- virtual bool **execute** ()
Executes the command.

Private Member Functions

- bool **applyGeometryChanges** ()
Apply the geometry changes to the current map.

Private Attributes

- int **horizontalNewValue**
Horizontal new value.
- **MapModel::Side horizontalDirection**
Horizontal direction in which to apply the new value.
- int **verticalNewValue**
Vertical new value.
- **MapModel::Side verticalDirection**
Vertical direction in which to apply the new value.
- int **maxHeightNewValue**
Max Height new value.
- **MapModel::Side maxHeightDirection**
Max Height direction in which to apply the new value.

Additional Inherited Members

8.3.1 Detailed Description

Apply the geometrical changes to the currently loaded map model.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.3.2 Constructor & Destructor Documentation

8.3.2.1 `ApplyMapGeometryChanges::ApplyMapGeometryChanges (int horizontalNewValue, int horizontalDirection, int verticalNewValue, int verticalDirection, int maxHeightNewValue, int maxHeightDirection, QObject * parent = 0)`

8.3.3 Member Function Documentation

8.3.3.1 `bool ApplyMapGeometryChanges::applyGeometryChanges () [private]`

Apply the geometry changes to the current map.

Returns

True, if changes were applied. Otherwise, false.

8.3.3.2 `bool ApplyMapGeometryChanges::execute () [virtual]`

Executes the command.

Returns

True, if successful.

Implements **NotUndoable** (p. 215).

8.3.4 Member Data Documentation

8.3.4.1 `MapModel::Side ApplyMapGeometryChanges::horizontalDirection` [private]

Horizontal direction in which to apply the new value.

8.3.4.2 `int ApplyMapGeometryChanges::horizontalNewValue` [private]

Horizontal new value.

8.3.4.3 `MapModel::Side ApplyMapGeometryChanges::maxHeightDirection` [private]

Max Height direction in which to apply the new value.

8.3.4.4 `int ApplyMapGeometryChanges::maxHeightNewValue` [private]

Max Height new value.

8.3.4.5 `MapModel::Side ApplyMapGeometryChanges::verticalDirection` [private]

Vetical direction in which to apply the new value.

8.3.4.6 `int ApplyMapGeometryChanges::verticalNewValue` [private]

Vertical new value.

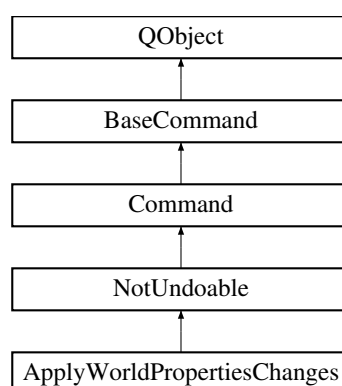
The documentation for this class was generated from the following files:

- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/ApplyMapGeometryChanges.h`
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/ApplyMapGeometryChanges.cpp`

8.4 ApplyWorldPropertiesChanges Class Reference

```
#include <ApplyWorldPropertiesChanges.h>
```

Inheritance diagram for ApplyWorldPropertiesChanges:



Public Member Functions

- **ApplyWorldPropertiesChanges** (QStandardItemModel ***chosenListModel**, QObject *parent=0)
- virtual bool **execute** ()
Executes the command.

Private Member Functions

- bool **applyChosenStarterMaps** ()

Private Attributes

- QStandardItemModel * **chosenListModel**
The given chosen list model with paths chosen.

Additional Inherited Members

8.4.1 Constructor & Destructor Documentation

- 8.4.1.1 **ApplyWorldPropertiesChanges::ApplyWorldPropertiesChanges** (QStandardItemModel * *chosenListModel*, QObject * *parent* = 0)

8.4.2 Member Function Documentation

- 8.4.2.1 bool **ApplyWorldPropertiesChanges::applyChosenStarterMaps** () [private]

- 8.4.2.2 bool **ApplyWorldPropertiesChanges::execute** () [virtual]

Executes the command.

Returns

True, if successful.

Implements **NotUndoable** (p. 215).

8.4.3 Member Data Documentation

- 8.4.3.1 QStandardItemModel* **ApplyWorldPropertiesChanges::chosenListModel** [private]

The given chosen list model with paths chosen.

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

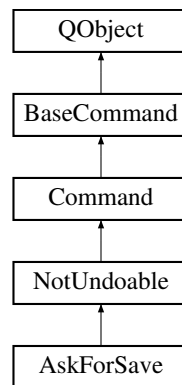
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/**ApplyWorldPropertiesChanges.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/**ApplyWorldPropertiesChanges.cpp**

8.5 AskForSave Class Reference

Ask if the world or the current map isn't saved, if you want to save.

```
#include <AskForSave.h>
```

Inheritance diagram for AskForSave:



Public Member Functions

- **AskForSave** (QObject *parent=0)
- virtual bool **execute** ()
Executes the command.

Additional Inherited Members

8.5.1 Detailed Description

Ask if the world or the current map isn't saved, if you want to save.

Ask if the current map isn't saved, if you want to save.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.5.2 Constructor & Destructor Documentation

8.5.2.1 **AskForSave::AskForSave** (QObject * *parent* = 0)

8.5.3 Member Function Documentation

8.5.3.1 **bool AskForSave::execute** () [virtual]

Executes the command.

Returns

True, if successful.

Implements **NotUndoable** (p. 215).

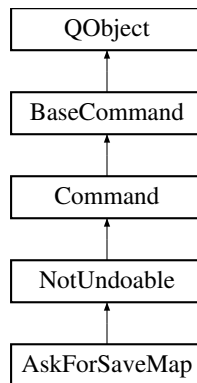
The documentation for this class was generated from the following files:

- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/**AskForSave.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/**AskForSave.cpp**

8.6 AskForSaveMap Class Reference

```
#include <AskForSaveMap.h>
```

Inheritance diagram for AskForSaveMap:



Public Member Functions

- **AskForSaveMap** (QObject *parent=0)
- virtual bool **execute** ()

Executes the command.

Private Attributes

- **YesNoCancelDialog yncDialog**

Dialog to ask for the current map to be saved.

Additional Inherited Members

8.6.1 Constructor & Destructor Documentation

8.6.1.1 **AskForSaveMap::AskForSaveMap** (QObject * *parent* = 0)

8.6.2 Member Function Documentation

8.6.2.1 bool **AskForSaveMap::execute** () [virtual]

Executes the command.

Returns

True, if successful.

Implements **NotUndoable** (p. 215).

8.6.3 Member Data Documentation

8.6.3.1 YesNoCancelDialog AskForSaveMap::yncDialog [private]

Dialog to ask for the current map to be saved.

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

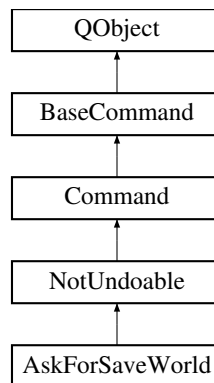
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/**AskForSaveMap.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/**AskForSaveMap.cpp**

8.7 AskForSaveWorld Class Reference

Ask whenever the world isn't saved, if you want to save.

```
#include <AskForSaveWorld.h>
```

Inheritance diagram for AskForSaveWorld:



Public Member Functions

- **AskForSaveWorld** (QObject *parent=0)
- virtual bool **execute** ()

Executes the command.

Private Attributes

- **YesNoCancelDialog yncDialog**

Dialog to ask for the current world to be saved.

Additional Inherited Members

8.7.1 Detailed Description

Ask whenever the world isn't saved, if you want to save.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.7.2 Constructor & Destructor Documentation

8.7.2.1 AskForSaveWorld::AskForSaveWorld (QObject * *parent* = 0)

8.7.3 Member Function Documentation

8.7.3.1 bool AskForSaveWorld::execute () [virtual]

Executes the command.

Returns

True, if successful.

Implements **NotUndoable** (p. 215).

8.7.4 Member Data Documentation

8.7.4.1 YesNoCancelDialog AskForSaveWorld::yncDialog [private]

Dialog to ask for the current world to be saved.

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

- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/**AskForSaveWorld.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/**AskForSaveWorld.cpp**

8.8 AStarMap Class Reference

The representation of the map for the A*-Algorithm.

#include <AStarMap.h>

Public Member Functions

- **AStarMap** ()
- **~AStarMap** ()
- void **fromMapModel** (MapModel *mapModel)
*Creates a **AStarMap** (p. 56) from a map model.*
- bool **calculatePath** (const QPoint &from, const QPoint &to)
Calculate the best path from a point to another.
- QVector< **AStarMapEntry** > **getPath** () const
- int **getMapWidth** () const
- int **getMapHeight** () const
- **AStarMapEntry** * **refEntry** (const QPoint &position) const
- **AStarMapEntry** * **refEntry** (int x, int y) const

Private Member Functions

- `QVector< AStarMapEntry * > getPossibleNeighborsOf (AStarMapEntry *entry)`
Get a list of neighboring entries.
- `void retracePath (AStarMapEntry *startEntry, AStarMapEntry *endEntry)`
Retraces calculated path from end to start.

Private Attributes

- `QVector< QVector< AStarMapEntry * > > map`
The weighted map over which to calculate a path from given points.
- `QVector< AStarMapEntry > path`
The found path.

8.8.1 Detailed Description

The representation of the map for the A*-Algorithm.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.8.2 Constructor & Destructor Documentation

8.8.2.1 `AStarMap::AStarMap ()`

8.8.2.2 `AStarMap::~~AStarMap ()`

8.8.3 Member Function Documentation

8.8.3.1 `bool AStarMap::calculatePath (const QPoint & from, const QPoint & to)`

Calculate the best path from a point to another.

Parameters

<i>from</i>	From where to start.
<i>to</i>	Where to go.

Returns

True, if a path was successfully created.

8.8.3.2 `void AStarMap::fromMapModel (MapModel * mapModel)`

Creates a **AStarMap** (p. 56) from a map model.

Parameters

<i>mapModel</i>	The map model used to create the AStarMap (p. 56).
-----------------	---

8.8.3.3 `int AStarMap::getMapHeight () const`

8.8.3.4 `int AStarMap::getMapWidth () const`

8.8.3.5 `QVector< AStarMapEntry > AStarMap::getPath () const`

8.8.3.6 `QVector< AStarMapEntry * > AStarMap::getPossibleNeighborsOf (AStarMapEntry * entry)` [private]

Get a list of neighboring entries.

Parameters

<i>entry</i>	The entry in the middle of those neigbmoring it.
--------------	--

Returns

A list of neighboring entries.

8.8.3.7 `AStarMapEntry * AStarMap::refEntry (const QPoint & position) const`

8.8.3.8 `AStarMapEntry * AStarMap::refEntry (int x, int y) const`

8.8.3.9 `void AStarMap::retracePath (AStarMapEntry * startEntry, AStarMapEntry * endEntry)` [private]

Retraces calculated path from end to start.

Parameters

<i>startEntry</i>	Where to trace to.
<i>endEntry</i>	Where to start tracing.

8.8.4 Member Data Documentation

8.8.4.1 `QVector< QVector <AStarMapEntry *> > AStarMap::map` [private]

The weighted map over which to calculate a path from given points.

8.8.4.2 `QVector< AStarMapEntry > AStarMap::path` [private]

The found path.

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

- /home/ryoga/git/pidgirl/pidgirl-engine/AStar/**AStarMap.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/AStar/**AStarMap.cpp**

8.9 AStarMapEntry Class Reference

```
#include <AStarMapEntry.h>
```

Public Member Functions

- **AStarMapEntry** ()
- **AStarMapEntry** (**StackModel** *stackModel)
- **AStarMapEntry** (const **QPoint** &position)
- **AStarMapEntry** (const **AStarMapEntry** &other)
- void **fromStackModel** (**StackModel** *stackModel)
*Create a **AStarMapEntry** (p. 58) out of a Stack Model.*
- double **heuristicDistanceTo** (**AStarMapEntry** *startEntry, **AStarMapEntry** *endEntry)
Calculate heuristical distance to the end.
- void **calculateF** (**AStarMapEntry** *startEntry, **AStarMapEntry** *endEntry)
Calculate heuristical distance to the end from the start.
- double **getWeight** () const
- bool **hasInfinityWeight** () const
- **QPoint** **getPosition** () const
- double **getG** () const
- double **getF** () const
- void **setWeight** (double weight)
- void **setWeightToInfinity** ()
- void **setG** (double g)
- void **setF** (double f)
- void **setCameFrom** (**AStarMapEntry** *cameFrom)
- **AStarMapEntry** * **refCameFrom** () const
- **AStarMapEntry** & **operator=** (const **AStarMapEntry** &other)
- bool **operator==** (const **AStarMapEntry** &other) const
- bool **operator!=** (const **AStarMapEntry** &other) const

Private Member Functions

- void **copy** (const **AStarMapEntry** &other)
*Create a copy of another **AStarMapEntry** (p. 58).*
- void **init** (double weight, const **QPoint** &position, double f=std::numeric_limits< double >::infinity(), double g=std::numeric_limits< double >::infinity())
*Initialized the **AStarMapEntry** (p. 58).*

Private Attributes

- double **weight**
The weight for the given map entry.
- **QPoint** **position**
The current position.
- double **g**
The shortest distance to the start.
- double **f**
The shortest distance calculated with the heuristical distance from start to end.
- **AStarMapEntry** * **cameFrom**
The entry where the shortest path came from.

8.9.1 Constructor & Destructor Documentation

8.9.1.1 `AStarMapEntry::AStarMapEntry ()`

8.9.1.2 `AStarMapEntry::AStarMapEntry (StackModel * stackModel)`

8.9.1.3 `AStarMapEntry::AStarMapEntry (const QPoint & position)`

8.9.1.4 `AStarMapEntry::AStarMapEntry (const AStarMapEntry & other)`

8.9.2 Member Function Documentation

8.9.2.1 `void AStarMapEntry::calculateF (AStarMapEntry * startEntry, AStarMapEntry * endEntry)`

Calculate heuristical distance to the end from the start.

Parameters

<i>endEntry</i>	The end entry where to heuristically determine the distance.
-----------------	--

8.9.2.2 `void AStarMapEntry::copy (const AStarMapEntry & other)` `[private]`

Create a copy of another **AStarMapEntry** (p. 58).

Parameters

<i>other</i>	The AStarMapEntry (p. 58) to be copied.
--------------	--

8.9.2.3 `void AStarMapEntry::fromStackModel (StackModel * stackModel)`

Create a **AStarMapEntry** (p. 58) out of a Stack Model.

Parameters

<i>stackModel</i>	The StackModel (p. 335) which is used to create the AStarMapEntry (p. 58).
-------------------	--

8.9.2.4 `double AStarMapEntry::getF () const`

8.9.2.5 `double AStarMapEntry::getG () const`

8.9.2.6 `QPoint AStarMapEntry::getPosition () const`

8.9.2.7 `double AStarMapEntry::getWeight () const`

8.9.2.8 `bool AStarMapEntry::hasInfinityWeight () const`

8.9.2.9 `double AStarMapEntry::heuristicalDistanceTo (AStarMapEntry * startEntry, AStarMapEntry * endEntry)`

Calculate heuristical distance to the end.

Parameters

<i>endEntry</i>	The end entry where to heuristically determine the distance.
-----------------	--

```
8.9.2.10 void AStarMapEntry::init ( double weight, const QPoint & position, double f =  
    std::numeric_limits<double>::infinity(), double g = std::numeric_  
    limits<double>::infinity() ) [private]
```

Initialized the **AStarMapEntry** (p. 58).

Parameters

<i>weight</i>	The initial weight.
<i>position</i>	The initial position.
<i>g</i>	Distance to start is initialized as -1 (Infinity).
<i>h</i>	Heuristical distance to end is initialized as -1 (Infinity).

8.9.2.11 `bool AStarMapEntry::operator!= (const AStarMapEntry & other) const`

8.9.2.12 `AStarMapEntry & AStarMapEntry::operator= (const AStarMapEntry & other)`

8.9.2.13 `bool AStarMapEntry::operator== (const AStarMapEntry & other) const`

8.9.2.14 `AStarMapEntry * AStarMapEntry::refCameFrom () const`

8.9.2.15 `void AStarMapEntry::setCameFrom (AStarMapEntry * cameFrom)`

8.9.2.16 `void AStarMapEntry::setF (double f)`

8.9.2.17 `void AStarMapEntry::setG (double g)`

8.9.2.18 `void AStarMapEntry::setWeight (double weight)`

8.9.2.19 `void AStarMapEntry::setWeightToInfinity ()`

8.9.3 Member Data Documentation

8.9.3.1 `AStarMapEntry* AStarMapEntry::cameFrom` [private]

The entry where the shortest path came from.

8.9.3.2 `double AStarMapEntry::f` [private]

The shortest distance calculated with the heuristical distance from start to end.

8.9.3.3 `double AStarMapEntry::g` [private]

The shortest distance to the start.

8.9.3.4 `QPoint AStarMapEntry::position` [private]

The current position.

8.9.3.5 `double AStarMapEntry::weight` [private]

The weight for the given map entry.

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

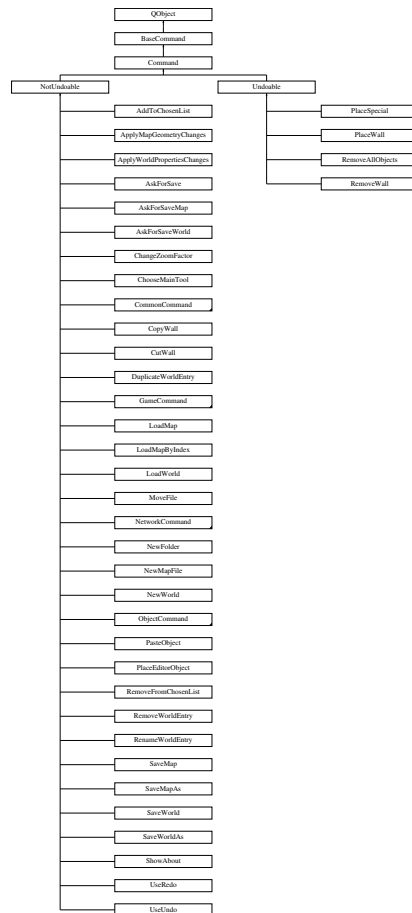
- `/home/ryoga/git/pidgirl/pidgirl-engine/AStar/AStarMapEntry.h`
- `/home/ryoga/git/pidgirl/pidgirl-engine/AStar/AStarMapEntry.cpp`

8.10 BaseCommand Class Reference

Interface to ensure the least needed functions for the commands.

```
#include <BaseCommand.h>
```

Inheritance diagram for BaseCommand:



Public Member Functions

- **BaseCommand** (QObject *parent=0)
- virtual bool **isExecutable** () const =0
Check if the command is executable.
- virtual bool **execute** ()=0
Executes the command.
- virtual void **undo** ()=0
Undo changes made by the command.
- virtual void **redo** ()=0
Redo changes made by the undoing of the command.
- virtual bool **isUndoable** ()=0
Determines undoability.
- virtual bool **isClearingUndoRedo** ()=0
Determines, if undo and redo should be cleared.
- bool **isErrorEnabled** () const
- virtual const QIcon **getCommandIcon** () const

- virtual void **setModels** (QObject *processor, QObject *undoRedo, QObject *editorModel, QObject *worldModel, QObject *gameModel, QObject *mapModel, QObject *networkModel)=0
- void **setErrorEnabled** (bool **errorEnabled**)

Private Attributes

- bool **errorEnabled**

8.10.1 Detailed Description

Interface to ensure the least needed functions for the commands.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.10.2 Constructor & Destructor Documentation

8.10.2.1 **BaseCommand::BaseCommand** (QObject * *parent* = 0) [explicit]

8.10.3 Member Function Documentation

8.10.3.1 virtual bool **BaseCommand::execute** () [pure virtual]

Executes the command.

Returns

True, if successful.

Implemented in **CreateCharacter** (p. 109), **SetupNewGame** (p. 325), **ShowGameOptions** (p. 330), **ConnectTo** (p. 100), **LoadGame** (p. 150), **FirstTurn** (p. 130), **Command** (p. 94), **RemoveAllObjects** (p. 273), **BroadcastCommand** (p. 69), **SaveEMailGame** (p. 297), **ApplyMapGeometryChanges** (p. 50), **CopyFolder** (p. 105), **AddToChosenList** (p. 48), **NewMapFile** (p. 210), **BroadcastUpdateCharacter** (p. 73), **RemoveFolder** (p. 275), **FindPath** (p. 129), **LoadMap** (p. 152), **LoadMapByIndex** (p. 153), **PlaceEditorObject** (p. 244), **PlaceWall** (p. 249), **LoadWorld** (p. 157), **MoveFile** (p. 201), **RemoveFromChosenList** (p. 277), **MoveCharacter** (p. 199), **ObjectCommand** (p. 227), **CopyWall** (p. 106), **PlaceSpecial** (p. 247), **RemoveWall** (p. 280), **DuplicateWorldEntry** (p. 112), **RemoveWorldEntry** (p. 281), **SpawnGameObject** (p. 333), **PreloadWorld** (p. 257), **ResumeEMailGame** (p. 293), **AskForSaveMap** (p. 54), **ApplyWorldPropertiesChanges** (p. 52), **AskForSaveWorld** (p. 56), **NewFolder** (p. 208), **NewWorld** (p. 211), **RenameWorldEntry** (p. 283), **SaveMap** (p. 302), **SaveMapAs** (p. 303), **SaveWorld** (p. 304), **SaveWorldAs** (p. 306), **PreloadAllMaps** (p. 255), **SaveGame** (p. 299), **SpawnPlayerOnCurrentMap** (p. 335), **CutWall** (p. 111), **PasteObject** (p. 242), **ChooseMainTool** (p. 77), **SendUpdateCharacterToServer** (p. 313), **ServerCommand** (p. 316), **UseRedo** (p. 359), **Quit** (p. 262), **ChangeZoomFactor** (p. 76), **LoadStartMapFromIndex** (p. 155), **ChangePlayerIDOnServer** (p. 74), **RequestMapFromServer** (p. 288), **RequestPlayerFromServer** (p. 290), **SendSpawnCharacterToServer** (p. 308), **ReceiveSpawnCharacterFrom** (p. 270), **BroadcastSpawnCharacter** (p. 71), **NotUndoable** (p. 215), **CommonCommand** (p. 98), **GameCommand** (p. 133), **ClientCommand** (p. 80), **NetworkCommand** (p. 202), **ReceiveUpdateCharacterFrom** (p. 271), **Undoable** (p. 354), **UseUndo** (p. 361), **AskForSave** (p. 53), **EndTurn** (p. 125), **NextTurn** (p. 213), **YourTurnInformation** (p. 374), **ClearPath** (p. 78), **ReceiveInitialPlayerInformation** (p. 264), **ReceiveMap** (p. 266),

ReceivePlayer (p. 267), **ReceivePlayerModels** (p. 268), **RequestPlayerModelsFromServer** (p. 291), **SendToServerEndTurn** (p. 311), **ClientDisconnected** (p. 81), **ReceiveChangePlayerID** (p. 263), **ReplyToRequestForMap** (p. 284), **ReplyToRequestForPlayer** (p. 286), **ReplyToRequestForPlayerModels** (p. 287), **SendInitialPlayerInformation** (p. 307), **SendToClientNextTurn** (p. 310), **ShowAbout** (p. 327), and **ShowDescription** (p. 328).

8.10.3.2 `const QIcon BaseCommand::getCommandIcon () const [virtual]`

8.10.3.3 `virtual bool BaseCommand::isClearingUndoRedo () [pure virtual]`

Determines, if undo and redo should be cleared.

Returns

True, if undo and redo should be cleared.

Implemented in **Command** (p. 95), **NotUndoable** (p. 215), and **Undoable** (p. 355).

8.10.3.4 `bool BaseCommand::isErrorEnabled () const`

8.10.3.5 `virtual bool BaseCommand::isExecutable () const [pure virtual]`

Check if the command is executable.

Returns

If the command is executable.

Implemented in **Command** (p. 95), **ObjectCommand** (p. 227), **ServerCommand** (p. 316), **NotUndoable** (p. 216), **CommonCommand** (p. 99), **GameCommand** (p. 133), **ClientCommand** (p. 80), **NetworkCommand** (p. 203), and **Undoable** (p. 355).

8.10.3.6 `virtual bool BaseCommand::isUndoable () [pure virtual]`

Determines undoability.

Returns

True, if this command can be undone.

Implemented in **Command** (p. 95), **NotUndoable** (p. 216), and **Undoable** (p. 355).

8.10.3.7 `virtual void BaseCommand::redo () [pure virtual]`

Redo changes made by the undoing of the command.

Implemented in **Command** (p. 95), **RemoveAllObjects** (p. 273), **PlaceWall** (p. 249), **PlaceSpecial** (p. 247), **RemoveWall** (p. 280), **NotUndoable** (p. 216), and **Undoable** (p. 355).

8.10.3.8 `void BaseCommand::setErrorEnabled (bool errorEnabled)`

8.10.3.9 `virtual void BaseCommand::setModels (QObject * processor, QObject * undoRedo, QObject * editorModel, QObject * worldModel, QObject * gameModel, QObject * mapModel, QObject * networkModel) [pure virtual]`

Implemented in **Command** (p. 96).

8.10.3.10 virtual void BaseCommand::undo () [pure virtual]

Undo changes made by the command.

Implemented in **Command** (p.96), **RemoveAllObjects** (p.273), **PlaceWall** (p.249), **PlaceSpecial** (p.247), **RemoveWall** (p.280), **NotUndoable** (p.216), and **Undoable** (p.355).

8.10.4 Member Data Documentation

8.10.4.1 bool BaseCommand::errorEnabled [private]

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

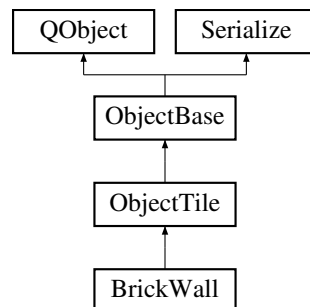
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/**BaseCommand.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/**BaseCommand.cpp**

8.11 BrickWall Class Reference

A wall of bricks.

```
#include <BrickWall.h>
```

Inheritance diagram for BrickWall:



Public Member Functions

- **BrickWall** (QObject *parent=0)
- **BrickWall** (const **BrickWall** &brickWall)
- QWidget * **createInformationWidget** (QWidget *parent) const
Create an information widget from an object to show informations about it.
- virtual **ObjectTile** * **copy** () const
- virtual void **update** (const **ObjectBase** &object)
- virtual **ObjectID::TileID** **getObjectID** () const
- virtual QMap< QString, **BaseCommand** * > **getCommandList** () const
- virtual void **serialize** (QDataStream &dataStream) const
- virtual void **serialize** (QIODevice *device) const
- virtual void **deserialize** (QDataStream &dataStream)
- virtual void **deserialize** (QIODevice *device)

Additional Inherited Members

8.11.1 Detailed Description

A wall of bricks.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.11.2 Constructor & Destructor Documentation

8.11.2.1 `BrickWall::BrickWall (QObject * parent = 0)`

8.11.2.2 `BrickWall::BrickWall (const BrickWall & brickWall)`

8.11.3 Member Function Documentation

8.11.3.1 `ObjectTile * BrickWall::copy () const` [virtual]

Implements **ObjectTile** (p. 238).

8.11.3.2 `QWidget * BrickWall::createInformationWidget (QWidget * parent) const` [virtual]

Create an information widget from an object to show informations about it.

Parameters

<i>parent</i>	The parent for the created information widget.
---------------	--

Returns

The widget that represents this object.

Implements **ObjectBase** (p. 220).

8.11.3.3 `void BrickWall::deserialize (QDataStream & dataStream)` [virtual]

Reimplemented from **ObjectTile** (p. 239).

8.11.3.4 `void BrickWall::deserialize (QIODevice * device)` [virtual]

Implements **ObjectTile** (p. 239).

8.11.3.5 `QMap< QString, BaseCommand * > BrickWall::getCommandList () const` [virtual]

Implements **ObjectBase** (p. 220).

8.11.3.6 **ObjectID::TileID** **BrickWall::getObjectID () const** [virtual]

Implements **ObjectTile** (p. 239).

8.11.3.7 **void** **BrickWall::serialize (QDataStream & *dataStream*) const** [virtual]

Reimplemented from **ObjectTile** (p. 239).

8.11.3.8 **void** **BrickWall::serialize (QIODevice * *device*) const** [virtual]

Implements **ObjectTile** (p. 239).

8.11.3.9 **void** **BrickWall::update (const ObjectBase & *object*)** [virtual]

Implements **ObjectTile** (p. 239).

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

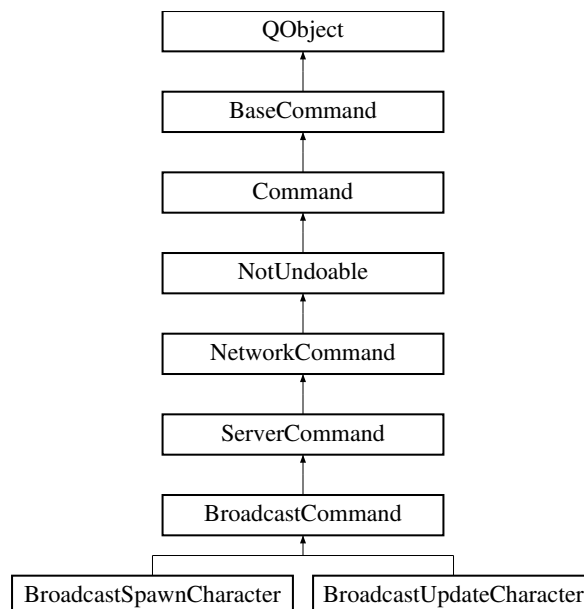
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/Tile/**BrickWall.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/Tile/**BrickWall.cpp**

8.12 BroadcastCommand Class Reference

Broadcast information to all clients on the same map the information was sent from.

```
#include <BroadcastCommand.h>
```

Inheritance diagram for BroadcastCommand:



Public Member Functions

- **BroadcastCommand** (unsigned int receivedByID, const QList< **PlayerModel** * > &playerList, const QString &localMapPath, QObject *parent=0)
Broadcast the command.

- virtual bool **execute** ()=0
Executes the command.

Protected Attributes

- QList< unsigned int > **broadcastPlayerIDs**
*The **Player** (p. 250) IDs to which information has to be sent to. The IDs are all from the same player map where the information changed.*

8.12.1 Detailed Description

Broadcast information to all clients on the same map the information was sent from.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.12.2 Constructor & Destructor Documentation

- 8.12.2.1 **BroadcastCommand::BroadcastCommand** (unsigned int *receivedByID*, const QList< **PlayerModel** * > & *playerList*, const QString & *localMapPath*, QObject * *parent* = 0)

Broadcast the command.

Parameters

<i>receivedByID</i>	The ID from which the command was received and does not have to be sent to.
<i>playerList</i>	The list of players.
<i>localMapPath</i>	The map that the information changed in.

8.12.3 Member Function Documentation

- 8.12.3.1 virtual bool **BroadcastCommand::execute** () [pure virtual]

Executes the command.

Returns

True, if successful.

Implements **ServerCommand** (p. 316).

Implemented in **BroadcastUpdateCharacter** (p. 73), and **BroadcastSpawnCharacter** (p. 71).

8.12.4 Member Data Documentation

- 8.12.4.1 QList<unsigned int> **BroadcastCommand::broadcastPlayerIDs** [protected]

The **Player** (p. 250) IDs to which information has to be sent to. The IDs are all from the same player map where the information changed.

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

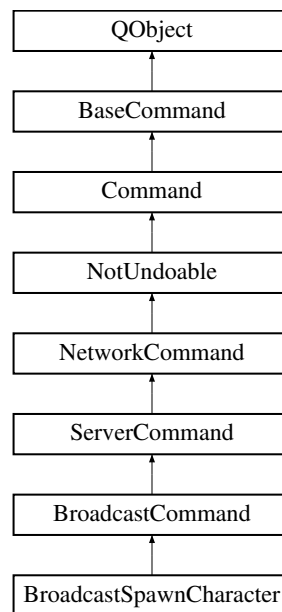
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Broadcast/**BroadcastCommand.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Broadcast/**BroadcastCommand.cpp**

8.13 BroadcastSpawnCharacter Class Reference

Broadcast the new character to be spawned.

```
#include <BroadcastSpawnCharacter.h>
```

Inheritance diagram for BroadcastSpawnCharacter:



Public Member Functions

- **BroadcastSpawnCharacter** (unsigned int receivedByID, const QList< **PlayerModel** * > &playerList, const QString &localMapPath, const **ObjectCharacter** &**newCharacter**, QObject *parent=0)
- virtual bool **execute** ()
Executes the command.

Private Attributes

- const **ObjectCharacter** & **newCharacter**
The new character that will be spawned.

Additional Inherited Members

8.13.1 Detailed Description

Broadcast the new character to be spawned.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.13.2 Constructor & Destructor Documentation

8.13.2.1 `BroadcastSpawnCharacter::BroadcastSpawnCharacter (unsigned int receivedByID, const QList< PlayerModel * > & playerList, const QString & localMapPath, const ObjectCharacter & newCharacter, QObject * parent = 0)`

8.13.3 Member Function Documentation

8.13.3.1 `bool BroadcastSpawnCharacter::execute () [virtual]`

Executes the command.

Returns

True, if successful.

Implements **BroadcastCommand** (p. 69).

8.13.4 Member Data Documentation

8.13.4.1 `const ObjectCharacter& BroadcastSpawnCharacter::newCharacter [private]`

The new character that will be spawned.

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

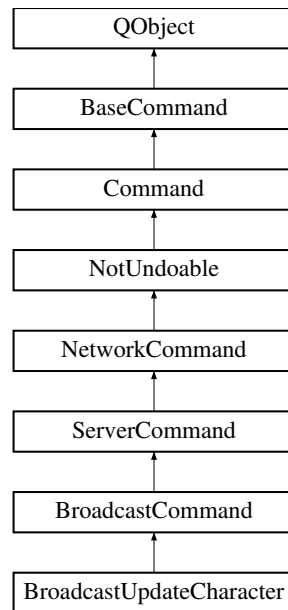
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Broadcast/BroadcastSpawnCharacter.h`
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Broadcast/BroadcastSpawnCharacter.cpp`

8.14 BroadcastUpdateCharacter Class Reference

Broadcast the character that has been changed.

```
#include <BroadcastUpdateCharacter.h>
```

Inheritance diagram for BroadcastUpdateCharacter:



Public Member Functions

- **BroadcastUpdateCharacter** (unsigned int receivedByID, const QList< **PlayerModel** * > &playerList, const QString &localMapPath, const **ObjectCharacter** &oldCharacter, const **ObjectCharacter** &newCharacter, QObject *parent=0)
- virtual bool **execute** ()
Executes the command.

Private Attributes

- const **ObjectCharacter** & **oldCharacter**
The old character that will be updated.
- const **ObjectCharacter** & **newCharacter**
The new character which has the informations to replace the old character.

Additional Inherited Members

8.14.1 Detailed Description

Broadcast the character that has been changed.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.14.2 Constructor & Destructor Documentation

8.14.2.1 `BroadcastUpdateCharacter::BroadcastUpdateCharacter (unsigned int receivedByID, const QList< PlayerModel * > & playerList, const QString & localMapPath, const ObjectCharacter & oldCharacter, const ObjectCharacter & newCharacter, QObject * parent = 0)`

8.14.3 Member Function Documentation

8.14.3.1 `bool BroadcastUpdateCharacter::execute () [virtual]`

Executes the command.

Returns

True, if successful.

Implements **BroadcastCommand** (p. 69).

8.14.4 Member Data Documentation

8.14.4.1 `const ObjectCharacter& BroadcastUpdateCharacter::newCharacter [private]`

The new character which has the informations to replace the old character.

8.14.4.2 `const ObjectCharacter& BroadcastUpdateCharacter::oldCharacter [private]`

The old character that will be updated.

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

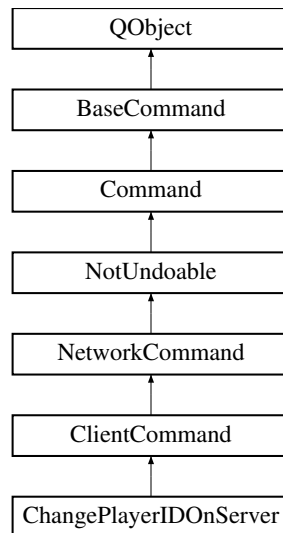
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Broadcast/BroadcastUpdateCharacter.h`
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Broadcast/BroadcastUpdateCharacter.cpp`

8.15 ChangePlayerIDOnServer Class Reference

Send the new player ID to the server that will replace the old one.

```
#include <ChangePlayerIDOnServer.h>
```

Inheritance diagram for ChangePlayerIDOnServer:



Public Member Functions

- **ChangePlayerIDOnServer** (unsigned int **newID**, QObject *parent=0)
- virtual bool **execute** ()
Executes the command.

Private Attributes

- unsigned int **newID**

Additional Inherited Members

8.15.1 Detailed Description

Send the new player ID to the server that will replace the old one.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.15.2 Constructor & Destructor Documentation

8.15.2.1 **ChangePlayerIDOnServer::ChangePlayerIDOnServer** (unsigned int *newID*, QObject * *parent* = 0)

8.15.3 Member Function Documentation

8.15.3.1 bool **ChangePlayerIDOnServer::execute** () [virtual]

Executes the command.

Returns

True, if successful.

Implements **ClientCommand** (p. 80).

8.15.4 Member Data Documentation**8.15.4.1** unsigned int ChangePlayerIDOnServer::newID [private]

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

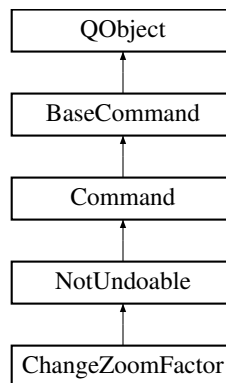
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/**ChangePlayerIDOnServer.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/**ChangePlayerIDOnServer.cpp**

8.16 ChangeZoomFactor Class Reference

Change the zoom factor for a map.

```
#include <ChangeZoomFactor.h>
```

Inheritance diagram for ChangeZoomFactor:

**Public Member Functions**

- **ChangeZoomFactor** (int **newFactor**, QObject *parent=0)
- virtual bool **execute** ()

Executes the command.

Private Attributes

- int **newFactor**

New Factor for zooming the view.

Additional Inherited Members**8.16.1 Detailed Description**

Change the zoom factor for a map.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.16.2 Constructor & Destructor Documentation**8.16.2.1** `ChangeZoomFactor::ChangeZoomFactor (int newFactor, QObject * parent = 0)`**8.16.3 Member Function Documentation****8.16.3.1** `bool ChangeZoomFactor::execute ()` [virtual]

Executes the command.

Returns

True, if successful.

Implements **NotUndoable** (p. 215).**8.16.4 Member Data Documentation****8.16.4.1** `int ChangeZoomFactor::newFactor` [private]

New Factor for zooming the view.

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

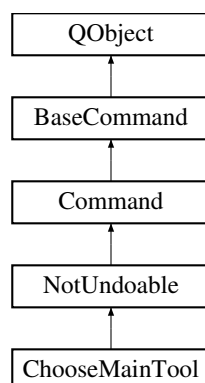
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/**ChangeZoomFactor.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/**ChangeZoomFactor.cpp**

8.17 ChooseMainTool Class Reference

Choose a Tool and set it in the editorModel.

`#include <ChooseMainTool.h>`

Inheritance diagram for ChooseMainTool:



Public Member Functions

- **ChooseMainTool** (**PIDGIRL::MainToolButton** *button*, *QObject *parent=0*)
- virtual bool **execute** ()
Executes the command.

Private Attributes

- **PIDGIRL::MainToolButton** *button*
The representation of the object to be placed.

Additional Inherited Members

8.17.1 Detailed Description

Choose a Tool and set it in the editorModel.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.17.2 Constructor & Destructor Documentation

8.17.2.1 **ChooseMainTool::ChooseMainTool** (**PIDGIRL::MainToolButton** *button*, *QObject *parent = 0*)

8.17.3 Member Function Documentation

8.17.3.1 bool **ChooseMainTool::execute** () *[virtual]*

Executes the command.

Returns

True, if successful.

Implements **NotUndoable** (p. 215).

8.17.4 Member Data Documentation

8.17.4.1 **PIDGIRL::MainToolButton** **ChooseMainTool::button** *[private]*

The representation of the object to be placed.

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

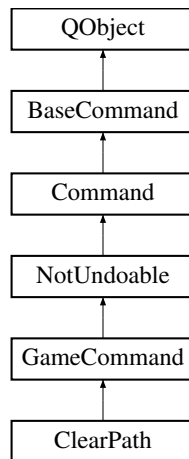
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Tools/**ChooseMainTool.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Tools/**ChooseMainTool.cpp**

8.18 ClearPath Class Reference

Clear the found path to a destination.

```
#include <ClearPath.h>
```

Inheritance diagram for ClearPath:



Public Member Functions

- **ClearPath** (QObject *parent=0)
- virtual bool **execute** ()
Executes the command.

Additional Inherited Members

8.18.1 Detailed Description

Clear the found path to a destination.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.18.2 Constructor & Destructor Documentation

8.18.2.1 ClearPath::ClearPath (QObject * parent = 0)

8.18.3 Member Function Documentation

8.18.3.1 bool ClearPath::execute () [virtual]

Executes the command.

Returns

True, if successful.

Implements **GameCommand** (p. 133).

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

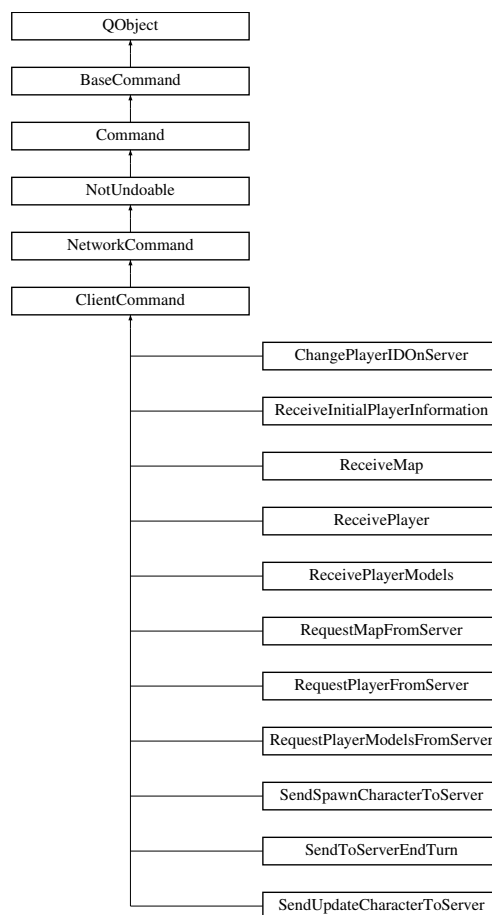
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Map/**ClearPath.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Map/**ClearPath.cpp**

8.19 ClientCommand Class Reference

Interface for a command executed by network clients.

```
#include <ClientCommand.h>
```

Inheritance diagram for ClientCommand:

**Public Member Functions**

- **ClientCommand** (QObject *parent=0)
- virtual bool **isExecutable** () const
Check if the command is executable.
- virtual bool **execute** ()=0
Executes the command.

Additional Inherited Members

8.19.1 Detailed Description

Interface for a command executed by network clients.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.19.2 Constructor & Destructor Documentation

8.19.2.1 `ClientCommand::ClientCommand (QObject * parent = 0)`

8.19.3 Member Function Documentation

8.19.3.1 `virtual bool ClientCommand::execute () [pure virtual]`

Executes the command.

Returns

True, if successful.

Implements **NetworkCommand** (p. 202).

Implemented in **SendUpdateCharacterToServer** (p. 313), **ChangePlayerIDOnServer** (p. 74), **RequestMapFromServer** (p. 288), **RequestPlayerFromServer** (p. 290), **SendSpawnCharacterToServer** (p. 308), **ReceiveInitialPlayerInformation** (p. 264), **ReceiveMap** (p. 266), **ReceivePlayer** (p. 267), **ReceivePlayerModels** (p. 268), **RequestPlayerModelsFromServer** (p. 291), and **SendToServerEndTurn** (p. 311).

8.19.3.2 `bool ClientCommand::isExecutable () const [virtual]`

Check if the command is executable.

Returns

If the command is executable.

Reimplemented from **NetworkCommand** (p. 203).

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

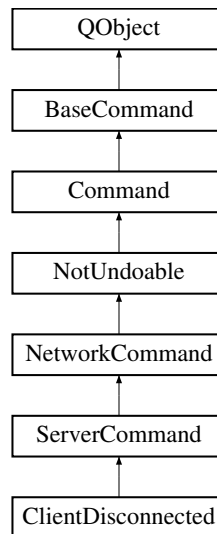
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/**ClientCommand.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/**ClientCommand.cpp**

8.20 ClientDisconnected Class Reference

Command (p. 93) is triggered whenever a client is disconnected.


```
#include <ClientDisconnected.h>
```

Inheritance diagram for ClientDisconnected:



Public Member Functions

- **ClientDisconnected** (unsigned int **playerID**, QObject *parent=0)
- virtual bool **execute** ()
Executes the command.

Additional Inherited Members

8.20.1 Detailed Description

Command (p. 93) is triggered whenever a client is disconnected.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.20.2 Constructor & Destructor Documentation

8.20.2.1 ClientDisconnected::ClientDisconnected (unsigned int *playerID*, QObject * *parent* = 0)

8.20.3 Member Function Documentation

8.20.3.1 bool ClientDisconnected::execute () [virtual]

Executes the command.

Returns

True, if successful.

Implements **ServerCommand** (p. 316).

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

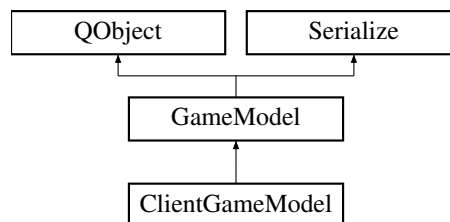
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/**ClientDisconnected.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/**ClientDisconnected.cpp**

8.21 ClientGameModel Class Reference

The model representing a game ran as a client where most information is accessible remotely.

```
#include <ClientGameModel.h>
```

Inheritance diagram for ClientGameModel:



Public Member Functions

- **ClientGameModel** (**Processor** *processor, QObject *parent)
- virtual bool **addNewPlayer** (unsigned int **playerID**, **PlayerModel** *playerModel)
Add a new player to the game.
- virtual bool **nextTurn** ()
Prepare for the next turn.
- virtual bool **beginTurn** ()
Begin the turn.
- virtual bool **firstTurn** ()
Prepare for the first turn.
- virtual bool **endTurn** ()
End your turn.
- virtual void **deactivatePlayer** (unsigned int **playerID**)
Deactivate a player.
- virtual void **addtoPreloadedMaps** (const QString &localMapPath, **MapModel** *map)
Add a preloaded map to the preloaded maps.
- virtual bool **loadMap** (const QString &localMapPath)
Load a map from a local map path.
- virtual bool **loadMap** (**MapModel** *map)
Load a map directly from a map model.
- virtual bool **spawnCharacter** (unsigned int **playerID**, **MapModel** *map, const **ObjectCharacter** &newCharacter)
Spawn a given character on a given map.
- virtual bool **updateCharacter** (unsigned int **playerID**, **MapModel** *map, const **ObjectCharacter** &oldCharacter, const **ObjectCharacter** &newCharacter)

- Update a given character with the information of a new character on a given map.*
- virtual void **changePlayerID** (unsigned int newID)
- Change the player ID.*
- virtual void **convertFromDifferentGameModel** (GameModel *gameModel)
- Update this game model with a game model of different type.*
- virtual **PIDGIRL::GameType** **getGameType** () const
- virtual bool **isYourTurn** () const
- virtual QMap< unsigned int, **PlayerModel** * > **getPlayerModels** () const
- virtual QMap< QString, **MapModel** * > **getPreloadedMaps** () const
- virtual **PlayerModel** * **refPlayerModel** (unsigned int **playerID**) const
- virtual **MapModel** * **refPreloadedMap** (const QString &localMapPath) const
- virtual **MapModel** * **refCurrentMap** () const
- virtual **Player** * **refPlayer** (unsigned int **playerID**) const
- virtual void **serialize** (QIODevice *device) const
- virtual void **deserialize** (QIODevice *device)

Private Member Functions

- void **setYourTurn** (bool **yourTurn**)
- virtual void **setCurrentMap** (**MapModel** *map)

Private Attributes

- **Processor** * **processor**
- **MapModel** * **currentMap**
- The current map loaded.*
- bool **yourTurn**
- True, if it is your turn.*

Additional Inherited Members

8.21.1 Detailed Description

The model representing a game ran as a client where most information is accessible remotely.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.21.2 Constructor & Destructor Documentation

8.21.2.1 **ClientGameModel::ClientGameModel** (**Processor** * *processor*, **QObject** * *parent*)

8.21.3 Member Function Documentation

8.21.3.1 bool **ClientGameModel::addNewPlayer** (unsigned int *playerID*, **PlayerModel** * *playerModel*) [virtual]

Add a new player to the game.

Parameters

<i>playerID</i>	The playerID to be added.
<i>playerModel</i>	The player model representing the player to be added.

Returns

True, if the player has been added. Otherwise, false.

Implements **GameModel** (p. 136).

8.21.3.2 `void ClientGameModel::addtoPreloadedMaps (const QString & localMapPath, MapModel * map) [virtual]`

Add a preloaded map to the preloaded maps.

Parameters

<i>localMapPath</i>	Local file path.
<i>map</i>	The preloaded map.

Implements **GameModel** (p. 137).

8.21.3.3 `bool ClientGameModel::beginTurn () [virtual]`

Begin the turn.

Implements **GameModel** (p. 137).

8.21.3.4 `void ClientGameModel::changePlayerID (unsigned int newID) [virtual]`

Change the player ID.

Parameters

<i>newID</i>	The new ID of the player.
--------------	---------------------------

Reimplemented from **GameModel** (p. 137).

8.21.3.5 `void ClientGameModel::convertFromDifferentGameModel (GameModel * gameModel) [virtual]`

Update this game model with a game model of different type.

Parameters

<i>gameModel</i>	Game Model with a different type than the one being updated.
------------------	--

Reimplemented from **GameModel** (p. 137).

8.21.3.6 `void ClientGameModel::deavtivatePlayer (unsigned int playerID) [virtual]`

Deactivate a player.

Parameters

<i>playerID</i>	The playerID of which to deactivate the character of.
-----------------	---

Implements **GameModel** (p. 137).

8.21.3.7 `void ClientGameModel::deserialize (QIODevice * device) [virtual]`

Implements **Serialize** (p. 314).

8.21.3.8 `bool ClientGameModel::endTurn () [virtual]`

End your turn.

Implements **GameModel** (p. 138).

8.21.3.9 `bool ClientGameModel::firstTurn () [virtual]`

Prepare for the first turn.

Implements **GameModel** (p. 138).

8.21.3.10 `PIDGIRL::GameType ClientGameModel::getGameType () const [virtual]`

Implements **GameModel** (p. 138).

8.21.3.11 `QMap< unsigned int, PlayerModel * > ClientGameModel::getPlayerModels () const [virtual]`

Reimplemented from **GameModel** (p. 138).

8.21.3.12 `QMap< QString, MapModel * > ClientGameModel::getPreloadedMaps () const [virtual]`

Implements **GameModel** (p. 138).

8.21.3.13 `bool ClientGameModel::isYourTurn () const [virtual]`

Implements **GameModel** (p. 138).

8.21.3.14 `bool ClientGameModel::loadMap (const QString & localMapPath) [virtual]`

Load a map from a local map path.

Parameters

<i>localMapPath</i>	The local map path from which to load a map.
---------------------	--

Returns

True, if the map has been loaded. Otherwise. false.

Implements **GameModel** (p. 139).

8.21.3.15 `bool ClientGameModel::loadMap (MapModel * map) [virtual]`

Load a map directly from a map model.

Parameters

<i>map</i>	The map model to be loaded.
------------	-----------------------------

Returns

True, if the map model has been loaded. Otherwise, false.

Reimplemented from **GameModel** (p. 140).

8.21.3.16 **bool** ClientGameModel::nextTurn () [virtual]

Prepare for the next turn.

Implements **GameModel** (p. 140).

8.21.3.17 **MapModel *** ClientGameModel::refCurrentMap () const [virtual]

Implements **GameModel** (p. 140).

8.21.3.18 **Player *** ClientGameModel::refPlayer (unsigned int *playerID*) const [virtual]

Implements **GameModel** (p. 140).

8.21.3.19 **PlayerModel *** ClientGameModel::refPlayerModel (unsigned int *playerID*) const [virtual]

Implements **GameModel** (p. 140).

8.21.3.20 **MapModel *** ClientGameModel::refPreloadedMap (const QString & *localMapPath*) const [virtual]

Implements **GameModel** (p. 140).

8.21.3.21 **void** ClientGameModel::serialize (QIODevice * *device*) const [virtual]

Implements **Serialize** (p. 314).

8.21.3.22 **void** ClientGameModel::setCurrentMap (**MapModel *** *map*) [private],[virtual]

Implements **GameModel** (p. 140).

8.21.3.23 **void** ClientGameModel::setYourTurn (bool *yourTurn*) [private]

8.21.3.24 **bool** ClientGameModel::spawnCharacter (unsigned int *playerID*, **MapModel *** *map*, const **ObjectCharacter &** *newCharacter*) [virtual]

Spawn a given character on a given map.

Parameters

<i>playerID</i>	The playerID of the player that wants to spawn a character.
<i>map</i>	The map where to spawn the character on.
<i>newCharacter</i>	The character to be spawned with given coordinates already set.

Returns

True, if the character has been spawned. Otherwise, false.

Reimplemented from **GameModel** (p. 141).

8.21.3.25 `bool ClientGameModel::updateCharacter (unsigned int playerID, MapModel * map, const ObjectCharacter & oldCharacter, const ObjectCharacter & newCharacter) [virtual]`

Update a given character with the information of a new character on a given map.

Parameters

<i>playerID</i>	The playerID of the player that wants to update a character.
<i>map</i>	The map where to update the character.
<i>oldCharacter</i>	The old character that will be updated.
<i>newCharacter</i>	The new character with the updated information.

Returns

True, if the character has been updated. Otherwise, false.

Reimplemented from **GameModel** (p. 141).

8.21.4 Member Data Documentation

8.21.4.1 `MapModel* ClientGameModel::currentMap [private]`

The current map loaded.

8.21.4.2 `Processor* ClientGameModel::processor [private]`

8.21.4.3 `bool ClientGameModel::yourTurn [private]`

True, if it is your turn.

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

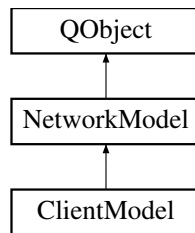
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/**ClientGameModel.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/**ClientGameModel.cpp**

8.22 ClientModel Class Reference

The client side implementation of the network model.

```
#include <ClientModel.h>
```

Inheritance diagram for ClientModel:



Public Member Functions

- **ClientModel** (const QString &ipAddress, int port, Processor *processor)
- virtual ~**ClientModel** ()
- virtual bool **isClient** ()
- virtual bool **isServer** ()
- virtual bool **changeClientID** (unsigned int fromID, unsigned int toID)
Change a client ID.
- bool **connectTo** ()
Connect to the given ipAddress with the given port. It will also set up world and game model with the informations given by the server.
- virtual void **sendData** (unsigned int data, unsigned int playerId=0)
Sends the given data to a playerId.
- virtual void **sendData** (int data, unsigned int playerId=0)
- virtual void **sendData** (bool data, unsigned int playerId=0)
- virtual void **sendData** (const **Serialize** &data, unsigned int playerId=0)
- virtual void **sendData** (const QString &data, unsigned int playerId=0)
- virtual void **receiveData** (unsigned int &data, unsigned int playerId=0)
Receive data from a client with the given playerId.
- virtual void **receiveData** (int &data, unsigned int playerId=0)
- virtual void **receiveData** (bool &data, unsigned int playerId=0)
- virtual void **receiveData** (**Serialize** &data, unsigned int playerId=0)
- virtual void **receiveData** (QString &data, unsigned int playerId=0)
- virtual QVector< QString > **getConnectionList** ()
- virtual QVector< unsigned int > **getConnectedPlayerIDs** ()

Private Slots

- void **receivePendingData** ()
- void **socketError** (QAbstractSocket::SocketError socketError)
- void **disconnectedFromServer** ()

Private Attributes

- **Processor** * **processor**
- QString **ipAddress**
The IPAddress of the Server.
- QTcpSocket **connection**
The connection to the server.

Additional Inherited Members

8.22.1 Detailed Description

The client side implementation of the network model.

The server side implementation of the network model.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.22.2 Constructor & Destructor Documentation

8.22.2.1 `ClientModel::ClientModel (const QString & ipAddress, int port, Processor * processor)`

8.22.2.2 `ClientModel::~~ClientModel ()` [virtual]

8.22.3 Member Function Documentation

8.22.3.1 `bool ClientModel::changeClientID (unsigned int fromID, unsigned int toID)` [virtual]

Change a client ID.

Parameters

<i>fromID</i>	The client connection ID from which to change.
<i>toID</i>	The client ID to which to change to.

Returns

True, if successfully changed. Otherwise, false.

Implements **NetworkModel** (p. 204).

8.22.3.2 `bool ClientModel::connectTo ()`

Connect to the given *ipAddress* with the given port. It will also set up world and game model with the informations given by the server.

Returns

True, if connection was established. Otherwise, false.

8.22.3.3 `void ClientModel::disconnectedFromServer ()` [private],[slot]

8.22.3.4 `QVector< unsigned int > ClientModel::getConnectedPlayerIDs ()` [virtual]

Implements **NetworkModel** (p. 205).

8.22.3.5 `QVector< QString > ClientModel::getConnectionList ()` [virtual]

Implements **NetworkModel** (p. 205).

8.22.3.6 `bool ClientModel::isClient ()` [virtual]

Implements **NetworkModel** (p. 205).

8.22.3.7 `bool ClientModel::isServer ()` [virtual]

Implements **NetworkModel** (p. 205).

8.22.3.8 `void ClientModel::receiveData (unsigned int & data, unsigned int playerID = 0)` [virtual]

Receive data from a client with the given *playerID*.

Parameters

<i>data</i>	The data to hold the data received.
<i>playerID</i>	The ID of the client data is received from.

Implements **NetworkModel** (p. 205).

8.22.3.9 `void ClientModel::receiveData (int & data, unsigned int playerID = 0)` [virtual]

Implements **NetworkModel** (p. 205).

8.22.3.10 `void ClientModel::receiveData (bool & data, unsigned int playerID = 0)` [virtual]

Implements **NetworkModel** (p. 205).

8.22.3.11 `void ClientModel::receiveData (Serialize & data, unsigned int playerID = 0)` [virtual]

Implements **NetworkModel** (p. 205).

8.22.3.12 `void ClientModel::receiveData (QString & data, unsigned int playerID = 0)` [virtual]

Implements **NetworkModel** (p. 206).

8.22.3.13 `void ClientModel::receivePendingData ()` [private],[slot]

8.22.3.14 `void ClientModel::sendData (unsigned int data, unsigned int playerID = 0)` [virtual]

Sends the given data to a *playerID*.

Parameters

<i>data</i>	The data to be sent.
<i>playerID</i>	The ID the data needs to be sent to.

Implements **NetworkModel** (p. 206).

8.22.3.15 void ClientModel::sendData (int *data*, unsigned int *playerID* = 0) [virtual]

Implements **NetworkModel** (p. 206).

8.22.3.16 void ClientModel::sendData (bool *data*, unsigned int *playerID* = 0) [virtual]

Implements **NetworkModel** (p. 206).

8.22.3.17 void ClientModel::sendData (const **Serialize** & *data*, unsigned int *playerID* = 0) [virtual]

Implements **NetworkModel** (p. 206).

8.22.3.18 void ClientModel::sendData (const QString & *data*, unsigned int *playerID* = 0) [virtual]

Implements **NetworkModel** (p. 206).

8.22.3.19 void ClientModel::socketError (QAbstractSocket::SocketError *socketError*) [private],[slot]

8.22.4 Member Data Documentation

8.22.4.1 QTcpSocket ClientModel::connection [private]

The connection to the server.

8.22.4.2 QString ClientModel::ipAddress [private]

The IPAddress of the Server.

8.22.4.3 Processor* ClientModel::processor [private]

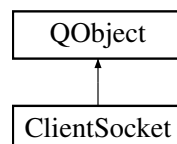
The documentation for this class was generated from the following files:

- /home/ryoga/git/pidgirl/pidgirl-engine/Network/**ClientModel.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Network/**ClientModel.cpp**

8.23 ClientSocket Class Reference

```
#include <ClientSocket.h>
```

Inheritance diagram for ClientSocket:



Signals

- void **readyRead** (**ClientSocket** **socket*)
- void **disconnected** (**ClientSocket** **socket*)

Public Member Functions

- **ClientSocket** (unsigned int **playerID**, QTcpSocket ***socket**, QObject *parent=0)
- **~ClientSocket** ()
- unsigned int **getPlayerID** () const
- void **setPlayerID** (unsigned int **playerID**)
- QTcpSocket * **refSocket** () const

Private Slots

- void **gotReadyRead** ()
- void **gotDisconnected** ()

Private Attributes

- unsigned int **playerID**
Consant unique ID connecting this connection with a given player model.
- QTcpSocket * **socket**
The socket connection to this player client.

8.23.1 Constructor & Destructor Documentation

8.23.1.1 **ClientSocket::ClientSocket** (unsigned int *playerID*, QTcpSocket * *socket*, QObject * *parent* = 0) [explicit]

8.23.1.2 **ClientSocket::~~ClientSocket** ()

8.23.2 Member Function Documentation

8.23.2.1 void **ClientSocket::disconnected** (**ClientSocket** * *socket*) [signal]

8.23.2.2 unsigned int **ClientSocket::getPlayerID** () const

8.23.2.3 void **ClientSocket::gotDisconnected** () [private],[slot]

8.23.2.4 void **ClientSocket::gotReadyRead** () [private],[slot]

8.23.2.5 void **ClientSocket::readyRead** (**ClientSocket** * *socket*) [signal]

8.23.2.6 QTcpSocket * **ClientSocket::refSocket** () const

8.23.2.7 void **ClientSocket::setPlayerID** (unsigned int *playerID*)

8.23.3 Member Data Documentation

8.23.3.1 unsigned int **ClientSocket::playerID** [private]

Consant unique ID connecting this connection with a given player model.

8.23.3.2 QTcpSocket* **ClientSocket::socket** [private]

The socket connection to this player client.

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

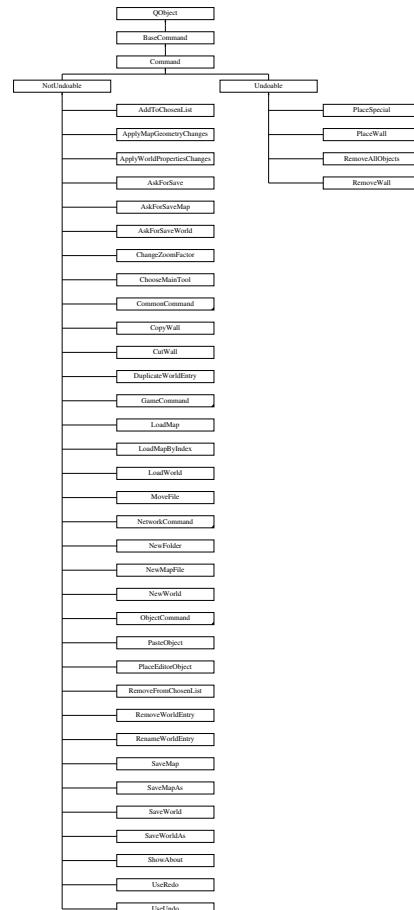
- /home/ryoga/git/pidgirl/pidgirl-engine/Network/**ClientSocket.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Network/**ClientSocket.cpp**

8.24 Command Class Reference

The base for all commands.

```
#include <Command.h>
```

Inheritance diagram for Command:



Public Member Functions

- **Command** (QObject *parent=0)
- virtual bool **isExecutable** () const =0
Check if the command is executable.
- virtual bool **execute** ()=0
Executes the command.
- virtual void **undo** ()=0
Undo changes made by the command.
- virtual void **redo** ()=0
Redo changes made by the undoing of the command.
- virtual bool **isUndoable** ()=0
Determines undoability.
- virtual bool **isClearingUndoRedo** ()=0

Determines, if undo and redo should be cleared.

- virtual void **setModels** (QObject ***processor**, QObject ***undoRedo**, QObject ***editorModel**, QObject ***worldModel**, QObject ***gameModel**, QObject ***mapModel**, QObject ***networkModel**)

Protected Attributes

- **Processor** * **processor**
- **UndoRedo** * **undoRedo**
- **EditorModel** * **editorModel**
- **WorldModel** * **worldModel**
- **GameModel** * **gameModel**
- **MapModel** * **mapModel**
- **NetworkModel** * **networkModel**

8.24.1 Detailed Description

The base for all commands.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.24.2 Constructor & Destructor Documentation

8.24.2.1 **Command::Command** (QObject * *parent* = 0)

8.24.3 Member Function Documentation

8.24.3.1 virtual bool **Command::execute** () [pure virtual]

Executes the command.

Returns

True, if successful.

Implements **BaseCommand** (p. 64).

Implemented in **CreateCharacter** (p. 109), **SetupNewGame** (p. 325), **ShowGameOptions** (p. 330), **ConnectTo** (p. 100), **LoadGame** (p. 150), **FirstTurn** (p. 130), **RemoveAllObjects** (p. 273), **BroadcastCommand** (p. 69), **SaveEMailGame** (p. 297), **ApplyMapGeometryChanges** (p. 50), **CopyFolder** (p. 105), **AddToChosenList** (p. 48), **NewMapFile** (p. 210), **BroadcastUpdateCharacter** (p. 73), **RemoveFolder** (p. 275), **FindPath** (p. 129), **LoadMap** (p. 152), **LoadMapByIndex** (p. 153), **PlaceEditorObject** (p. 244), **PlaceWall** (p. 249), **LoadWorld** (p. 157), **MoveFile** (p. 201), **RemoveFromChosenList** (p. 277), **MoveCharacter** (p. 199), **ObjectCommand** (p. 227), **CopyWall** (p. 106), **PlaceSpecial** (p. 247), **RemoveWall** (p. 280), **DuplicateWorldEntry** (p. 112), **RemoveWorldEntry** (p. 281), **SpawnGameObject** (p. 333), **PreloadWorld** (p. 257), **ResumeEMailGame** (p. 293), **AskForSaveMap** (p. 54), **ApplyWorldPropertiesChanges** (p. 52), **AskForSaveWorld** (p. 56), **NewFolder** (p. 208), **NewWorld** (p. 211), **RenameWorldEntry** (p. 283), **SaveMap** (p. 302), **SaveMapAs** (p. 303),

SaveWorld (p. 304), **SaveWorldAs** (p. 306), **PreloadAllMaps** (p. 255), **SaveGame** (p. 299), **SpawnPlayerOnCurrentMap** (p. 335), **CutWall** (p. 111), **PasteObject** (p. 242), **ChooseMainTool** (p. 77), **SendUpdateCharacterToServer** (p. 313), **ServerCommand** (p. 316), **UseRedo** (p. 359), **Quit** (p. 262), **ChangeZoomFactor** (p. 76), **LoadStartMapFromIndex** (p. 155), **ChangePlayerIDOnServer** (p. 74), **RequestMapFromServer** (p. 288), **RequestPlayerFromServer** (p. 290), **SendSpawnCharacterToServer** (p. 308), **ReceiveSpawnCharacterFrom** (p. 270), **BroadcastSpawnCharacter** (p. 71), **NotUndoable** (p. 215), **CommonCommand** (p. 98), **GameCommand** (p. 133), **ClientCommand** (p. 80), **NetworkCommand** (p. 202), **ReceiveUpdateCharacterFrom** (p. 271), **Undoable** (p. 354), **UseUndo** (p. 361), **AskForSave** (p. 53), **EndTurn** (p. 125), **NextTurn** (p. 213), **YourTurnInformation** (p. 374), **ClearPath** (p. 78), **ReceiveInitialPlayerInformation** (p. 264), **ReceiveMap** (p. 266), **ReceivePlayer** (p. 267), **ReceivePlayerModels** (p. 268), **RequestPlayerModelsFromServer** (p. 291), **SendToServerEndTurn** (p. 311), **ClientDisconnected** (p. 81), **ReceiveChangePlayerID** (p. 263), **ReplyToRequestForMap** (p. 284), **ReplyToRequestForPlayer** (p. 286), **ReplyToRequestForPlayerModels** (p. 287), **SendInitialPlayerInformation** (p. 307), **SendToClientNextTurn** (p. 310), **ShowAbout** (p. 327), and **ShowDescription** (p. 328).

8.24.3.2 `virtual bool Command::isClearingUndoRedo() [pure virtual]`

Determines, if undo and redo should be cleared.

Returns

True, if undo and redo should be cleared.

Implements **BaseCommand** (p. 65).

Implemented in **NotUndoable** (p. 215), and **Undoable** (p. 355).

8.24.3.3 `virtual bool Command::isExecutable() const [pure virtual]`

Check if the command is executable.

Returns

If the command is executable.

Implements **BaseCommand** (p. 65).

Implemented in **ObjectCommand** (p. 227), **ServerCommand** (p. 316), **NotUndoable** (p. 216), **CommonCommand** (p. 99), **GameCommand** (p. 133), **ClientCommand** (p. 80), **NetworkCommand** (p. 203), and **Undoable** (p. 355).

8.24.3.4 `virtual bool Command::isUndoable() [pure virtual]`

Determines undoability.

Returns

True, if this command can be undone.

Implements **BaseCommand** (p. 65).

Implemented in **NotUndoable** (p. 216), and **Undoable** (p. 355).

8.24.3.5 `virtual void Command::redo() [pure virtual]`

Redo changes made by the undoing of the command.

Implements **BaseCommand** (p. 65).

Implemented in **RemoveAllObjects** (p. 273), **PlaceWall** (p. 249), **PlaceSpecial** (p. 247), **RemoveWall** (p. 280), **NotUndoable** (p. 216), and **Undoable** (p. 355).

8.24.3.6 `void Command::setModels (QObject * processor, QObject * undoRedo, QObject * editorModel, QObject * worldModel, QObject * gameModel, QObject * mapModel, QObject * networkModel)` [virtual]

Implements **BaseCommand** (p. 65).

8.24.3.7 `virtual void Command::undo ()` [pure virtual]

Undo changes made by the command.

Implements **BaseCommand** (p. 66).

Implemented in **RemoveAllObjects** (p. 273), **PlaceWall** (p. 249), **PlaceSpecial** (p. 247), **RemoveWall** (p. 280), **NotUndoable** (p. 216), and **Undoable** (p. 355).

8.24.4 Member Data Documentation

8.24.4.1 `EditorModel* Command::editorModel` [protected]

8.24.4.2 `GameModel* Command::gameModel` [protected]

8.24.4.3 `MapModel* Command::mapModel` [protected]

8.24.4.4 `NetworkModel* Command::networkModel` [protected]

8.24.4.5 `Processor* Command::processor` [protected]

8.24.4.6 `UndoRedo* Command::undoRedo` [protected]

8.24.4.7 `WorldModel* Command::worldModel` [protected]

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

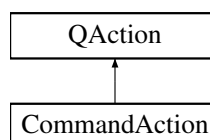
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/**Command.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/**Command.cpp**

8.25 CommandAction Class Reference

An action that will trigger a command.

```
#include <CommandAction.h>
```

Inheritance diagram for CommandAction:



Signals

- void **triggerWithName** (const QString &)

Public Member Functions

- **CommandAction** (const QIcon &icon, const QString &text, QObject *parent)

Private Slots

- void **commandActionTriggered** ()

8.25.1 Detailed Description

An action that will trigger a command.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.25.2 Constructor & Destructor Documentation

8.25.2.1 **CommandAction::CommandAction** (const QIcon & *icon*, const QString & *text*, QObject * *parent*)

8.25.3 Member Function Documentation

8.25.3.1 void **CommandAction::commandActionTriggered** () [private],[slot]

8.25.3.2 void **CommandAction::triggerWithName** (const QString &) [signal]

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

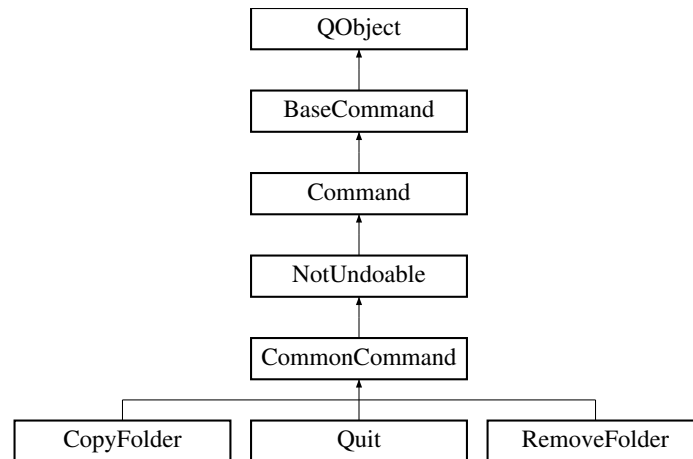
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/**CommandAction.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/**CommandAction.cpp**

8.26 CommonCommand Class Reference

Interface for commonly used commands.

```
#include <CommonCommand.h>
```

Inheritance diagram for CommonCommand:



Public Member Functions

- **CommonCommand** (QObject *parent=0)
- virtual bool **isExecutable** () const
Check if the command is executable.
- virtual bool **execute** ()=0
Executes the command.

Additional Inherited Members

8.26.1 Detailed Description

Interface for commonly used commands.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.26.2 Constructor & Destructor Documentation

8.26.2.1 **CommonCommand::CommonCommand** (QObject * *parent* = 0)

8.26.3 Member Function Documentation

8.26.3.1 virtual bool **CommonCommand::execute** () [pure virtual]

Executes the command.

Returns

True, if successful.

Implements **NotUndoable** (p. 215).

Implemented in **CopyFolder** (p. 105), **RemoveFolder** (p. 275), and **Quit** (p. 262).

8.26.3.2 bool CommonCommand::isExecutable () const [virtual]

Check if the command is executable.

Returns

If the command is executable.

Reimplemented from **NotUndoable** (p. 216).

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

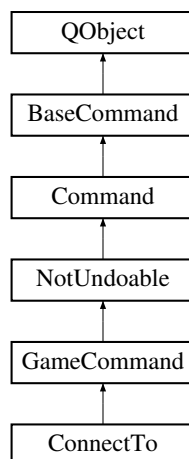
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/**CommonCommand.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/**CommonCommand.cpp**

8.27 ConnectTo Class Reference

Connect to a running **PIDGIRL** (p. 37) Server.

```
#include <ConnectTo.h>
```

Inheritance diagram for ConnectTo:



Public Member Functions

- **ConnectTo** (QObject *parent=0)
- virtual bool **execute** ()
Executes the command.
- bool **initToConnectDialog** ()
Initialize the Connect Dialog.

Private Slots

- void **connectToServer** ()
Connect a server, close dialog with accept, if succesful and otherwise close dialog with reject.

Private Attributes

- **ClientModel** * **clientModel**
- QDialog **connectToDialog**
- QGridLayout * **connectToLayout**
- QLabel * **IPAddressLabel**
- QLineEdit * **IPAddressEdit**
- QRegExpValidator * **IPAddressValidator**
- QLabel * **portLabel**
- QLineEdit * **portEdit**
- QIntValidator * **portValidator**
- QPushButton * **startNetworkButton**

Additional Inherited Members

8.27.1 Detailed Description

Connect to a running **PIDGIRL** (p. 37) Server.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.27.2 Constructor & Destructor Documentation

8.27.2.1 **ConnectTo::ConnectTo** (**QObject** * *parent* = 0)

8.27.3 Member Function Documentation

8.27.3.1 **void ConnectTo::connectToServer** () [private],[slot]

Connect a server, close dialog with accept, if succesful and otherwise close dialog with reject.

8.27.3.2 **bool ConnectTo::execute** () [virtual]

Executes the command.

Returns

True, if successful.

Implements **GameCommand** (p. 133).

8.27.3.3 **bool ConnectTo::initToConnectDialog** ()

Initialize the Connect Dialog.

Returns

True, if initialization is successful. Otherwise, false.

8.27.4 Member Data Documentation

- 8.27.4.1 `ClientModel* ConnectTo::clientModel` [private]
- 8.27.4.2 `QDialog ConnectTo::connectToDialog` [private]
- 8.27.4.3 `QGridLayout* ConnectTo::connectToLayout` [private]
- 8.27.4.4 `QLineEdit* ConnectTo::IPAddressEdit` [private]
- 8.27.4.5 `QLabel* ConnectTo::IPAddressLabel` [private]
- 8.27.4.6 `QRegExpValidator* ConnectTo::IPAddressValidator` [private]
- 8.27.4.7 `QLineEdit* ConnectTo::portEdit` [private]
- 8.27.4.8 `QLabel* ConnectTo::portLabel` [private]
- 8.27.4.9 `QIntValidator* ConnectTo::portValidator` [private]
- 8.27.4.10 `QPushButton* ConnectTo::startNetworkButton` [private]

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

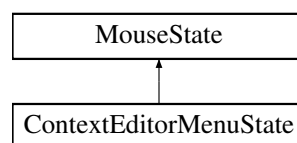
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/**ConnectTo.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/**ConnectTo.cpp**

8.28 ContextEditorMenuState Class Reference

The state in which to show the editor context menu.

```
#include <ContextEditorMenuState.h>
```

Inheritance diagram for ContextEditorMenuState:



Public Member Functions

- **ContextEditorMenuState (Processor *processor, StackModel *target)**
- virtual **MouseState * nextState (MouseState::MouseEventType type, QMouseEvent *event, StackModel *targetStack, StackModel *fromStack=0)**
Determines which state is next, depending on the input. This method might destroy this state and changes occur.
- virtual bool **showContextMenu ()**

Additional Inherited Members

8.28.1 Detailed Description

The state in which to show the editor context menu.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.28.2 Constructor & Destructor Documentation

8.28.2.1 `ContextEditorMenuState::ContextEditorMenuState (Processor * processor, StackModel * target)`

8.28.3 Member Function Documentation

8.28.3.1 `MouseState * ContextEditorMenuState::nextState (MouseState::MouseEventType type, QMouseEvent * event, StackModel * targetStack, StackModel * fromStack = 0) [virtual]`

Determines which state is next, depending on the input. This method might destroy this state and changes occur.

Parameters

<i>event</i>	Holds the event that has been triggered.
<i>stack</i>	The stack where this method has been invoked from.

Returns

a pointer to the next state.

Implements **MouseState** (p. 197).8.28.3.2 `bool ContextEditorMenuState::showContextMenu () [virtual]`Reimplemented from **MouseState** (p. 198).

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

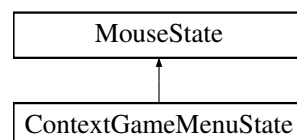
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/**ContextEditorMenuState.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/**ContextEditorMenuState.cpp**

8.29 ContextGameMenuState Class Reference

The state in which to show the game menu.

#include <ContextGameMenuState.h>

Inheritance diagram for ContextGameMenuState:



Public Member Functions

- **ContextGameMenuState** (**Processor** *processor, **StackModel** *target)
- virtual **MouseState** * **nextState** (**MouseState::MouseEventType** type, **QMouseEvent** *event, **StackModel** *targetStack, **StackModel** *fromStack=0)

Determines which state is next, depending on the input. This method might destroy this state and changes occur.
- virtual bool **showContextMenu** ()

Additional Inherited Members

8.29.1 Detailed Description

The state in which to show the game menu.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.29.2 Constructor & Destructor Documentation

8.29.2.1 **ContextGameMenuState::ContextGameMenuState** (**Processor** * processor, **StackModel** * target)

8.29.3 Member Function Documentation

8.29.3.1 **MouseState** * **ContextGameMenuState::nextState** (**MouseState::MouseEventType** type, **QMouseEvent** * event, **StackModel** * targetStack, **StackModel** * fromStack = 0) [virtual]

Determines which state is next, depending on the input. This method might destroy this state and changes occur.

Parameters

<i>event</i>	Holds the event that has been triggered.
<i>stack</i>	The stack where this method has been invoked from.

Returns

a pointer to the next state.

Implements **MouseState** (p. 197).

8.29.3.2 **bool** **ContextGameMenuState::showContextMenu** () [virtual]

Reimplemented from **MouseState** (p. 198).

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

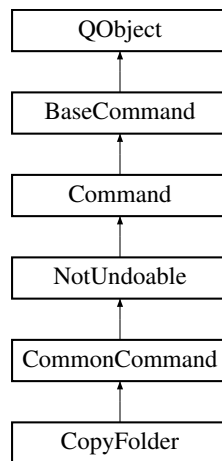
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/**ContextGameMenuState.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/**ContextGameMenuState.cpp**

8.30 CopyFolder Class Reference

Copy a folder recursively to another location.

```
#include <CopyFolder.h>
```

Inheritance diagram for CopyFolder:



Public Member Functions

- **CopyFolder** (const QDir &**from**, const QDir &**to**, QObject *parent=0)
- virtual bool **execute** ()

Executes the command.

Private Member Functions

- QList< QPair< QFileInfoList, QDir > > **copy** (QPair< QFileInfoList, QDir > next)

The actual copy function.

Private Attributes

- QDir **from**
The directory from which to copy the folder.
- QDir **to**
The directory to which the folder will be copied to.

Additional Inherited Members

8.30.1 Detailed Description

Copy a folder recursively to another location.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.30.2 Constructor & Destructor Documentation

8.30.2.1 `CopyFolder::CopyFolder (const QDir & from, const QDir & to, QObject * parent = 0)`

8.30.3 Member Function Documentation

8.30.3.1 `QList< QPair< QFileInfoList, QDir > > CopyFolder::copy (QPair< QFileInfoList, QDir > next)` [private]

The actual copy function.

Parameters

<i>next</i>	Represents the current entries in a directory over which to find new files and folders.
-------------	---

Returns

A list of folders found in the current entries which need to be looked over as well.

8.30.3.2 `bool CopyFolder::execute ()` [virtual]

Executes the command.

Returns

True, if successful.

Implements **CommonCommand** (p. 98).

8.30.4 Member Data Documentation

8.30.4.1 `QDir CopyFolder::from` [private]

The directory from which to copy the folder.

8.30.4.2 `QDir CopyFolder::to` [private]

The directory to which the folder will be copied to.

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

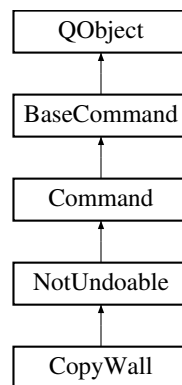
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/**CopyFolder.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/**CopyFolder.cpp**

8.31 CopyWall Class Reference

Copy the wall to the clipboard.

#include <CopyWall.h>

Inheritance diagram for CopyWall:



Public Member Functions

- **CopyWall** (**StackModel** *stackModel, QObject *parent=0)
- virtual bool **execute** ()
Executes the command.

Private Attributes

- **ObjectTile** * wall
The wall to be copied.

Additional Inherited Members

8.31.1 Detailed Description

Copy the wall to the clipboard.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.31.2 Constructor & Destructor Documentation

8.31.2.1 CopyWall::CopyWall (StackModel * stackModel, QObject * parent = 0)

8.31.3 Member Function Documentation

8.31.3.1 bool CopyWall::execute () [virtual]

Executes the command.

Returns

True, if successful.

Implements **NotUndoable** (p. 215).

8.31.4 Member Data Documentation**8.31.4.1 ObjectTile* CopyWall::wall** [private]

The wall to be copied.

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

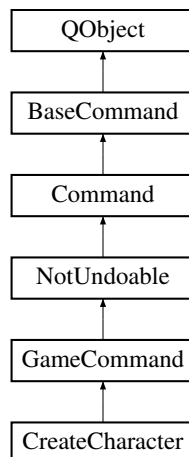
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/**CopyWall.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/**CopyWall.cpp**

8.32 CreateCharacter Class Reference

Create a character and put his starting information into the game model.

```
#include <CreateCharacter.h>
```

Inheritance diagram for CreateCharacter:

**Public Member Functions**

- **CreateCharacter** (QObject *parent=0)
- **~CreateCharacter** ()
- virtual bool **execute** ()

Executes the command.

Private Slots

- void **characterDone** ()
Check for character being done, and triggering accept if so.
- void **connectionsChanged** ()
Connection has changed.

Private Member Functions

- bool **init** ()
Initialize Create Character.
- void **deleteConnectionListItems** ()
Delete the connection list items.

Private Attributes

- QDialog **characterCreationDialog**
- OptionsLayout * **characterCreationLayout**
- QWidget * **informationWidget**
- QGridLayout * **informationWidgetLayout**
- QRegExpValidator **playerNameValidator**
- QLineEdit * **playerNameEdit**
- QWidget * **chooseStartMap**
- QGridLayout * **chooseStartMapLayout**
- QListView * **chooseListView**
- QVector< QStandardItem * > **chooseListItems**
- QStandardItemModel * **chooseListModel**
- QWidget * **serverOverview**
- QGridLayout * **serverOverviewLayout**
- QLabel * **connectionLabel**
- QListView * **connectionListView**
- QVector< QStandardItem * > **connectionListItems**
- QStandardItemModel * **connectionListModel**

Additional Inherited Members

8.32.1 Detailed Description

Create a character and put his starting information into the game model.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.32.2 Constructor & Destructor Documentation

8.32.2.1 **CreateCharacter::CreateCharacter** (QObject * *parent* = 0)

8.32.2.2 **CreateCharacter::~~CreateCharacter** ()

8.32.3 Member Function Documentation

8.32.3.1 **void CreateCharacter::characterDone** () [private],[slot]

Check for character being done, and triggering accept if so.

8.32.3.2 void CreateCharacter::connectionsChanged () [private],[slot]

Connection has changed.

8.32.3.3 void CreateCharacter::deleteConnectionListItems () [private]

Delete the connection list items.

8.32.3.4 bool CreateCharacter::execute () [virtual]

Executes the command.

Returns

True, if successful.

Implements **GameCommand** (p. 133).

8.32.3.5 bool CreateCharacter::init () [private]

Initialize Create Character.

Returns

True, if initialization was succsesful. Otherwise, false.

8.32.4 Member Data Documentation

8.32.4.1 QDialog CreateCharacter::characterCreationDialog [private]

8.32.4.2 OptionsLayout* CreateCharacter::characterCreationLayout [private]

8.32.4.3 QVector< QStandardItem * > CreateCharacter::chooseListItems [private]

8.32.4.4 QStandardItemModel* CreateCharacter::chooseListModel [private]

8.32.4.5 QListView* CreateCharacter::chooseListView [private]

8.32.4.6 QWidget* CreateCharacter::chooseStartMap [private]

8.32.4.7 QGridLayout* CreateCharacter::chooseStartMapLayout [private]

8.32.4.8 QLabel* CreateCharacter::connectionLabel [private]

8.32.4.9 QVector< QStandardItem * > CreateCharacter::connectionListItems [private]

8.32.4.10 QStandardItemModel* CreateCharacter::connectionListModel [private]

8.32.4.11 QListView* CreateCharacter::connectionListView [private]

8.32.4.12 QWidget* CreateCharacter::informationWidget [private]

8.32.4.13 QGridLayout* CreateCharacter::informationWidgetLayout [private]

8.32.4.14 QLineEdit* CreateCharacter::playerNameEdit [private]

8.32.4.15 `QRegExpValidator CreateCharacter::playerNameValidator` [private]

8.32.4.16 `QWidget* CreateCharacter::serverOverview` [private]

8.32.4.17 `QGridLayout* CreateCharacter::serverOverviewLayout` [private]

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

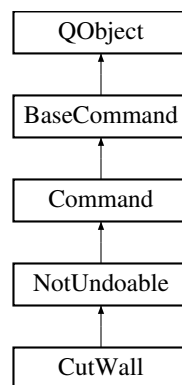
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/CreateCharacter.h`
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/CreateCharacter.cpp`

8.33 CutWall Class Reference

Copy and delete (Cut) the wall to the clipboard.

```
#include <CutWall.h>
```

Inheritance diagram for CutWall:



Public Member Functions

- **CutWall** (**StackModel** ***stackModel**, **QObject** ***parent**=0)
- virtual bool **execute** ()
Executes the command.

Private Attributes

- **StackModel** * **stackModel**
The stack from which to cut the wall.

Additional Inherited Members

8.33.1 Detailed Description

Copy and delete (Cut) the wall to the clipboard.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.33.2 Constructor & Destructor Documentation

8.33.2.1 `CutWall::CutWall (StackModel * stackModel, QObject * parent = 0)`

8.33.3 Member Function Documentation

8.33.3.1 `bool CutWall::execute ()` [virtual]

Executes the command.

Returns

True, if successful.

Implements **NotUndoable** (p. 215).

8.33.4 Member Data Documentation

8.33.4.1 `StackModel* CutWall::stackModel` [private]

The stack from which to cut the wall.

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

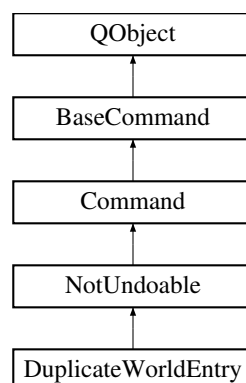
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/**CutWall.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/**CutWall.cpp**

8.34 DuplicateWorldEntry Class Reference

Make a copy of an entry in the world.

#include <DuplicateWorldEntry.h>

Inheritance diagram for DuplicateWorldEntry:



Public Member Functions

- **DuplicateWorldEntry** (const QModelIndex &**index**, QObject *parent=0)
- virtual bool **execute** ()

Executes the command.

Private Attributes

- QModelIndex **index**

The index to be duplicated entry.

Additional Inherited Members

8.34.1 Detailed Description

Make a copy of an entry in the world.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.34.2 Constructor & Destructor Documentation

8.34.2.1 DuplicateWorldEntry::DuplicateWorldEntry (const QModelIndex & *index*, QObject * *parent* = 0)

8.34.3 Member Function Documentation

8.34.3.1 bool DuplicateWorldEntry::execute () [virtual]

Executes the command.

Returns

True, if successful.

Implements **NotUndoable** (p. 215).

8.34.4 Member Data Documentation

8.34.4.1 QModelIndex DuplicateWorldEntry::index [private]

The index to be duplicated entry.

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

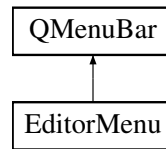
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/**DuplicateWorldEntry.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/**DuplicateWorldEntry.cpp**

8.35 EditorMenu Class Reference

Represents the menubar of the editor.

```
#include <EditorMenu.h>
```

Inheritance diagram for EditorMenu:



Public Slots

- void **createNewWorld** ()
Create and save a new empty World, load it into the world model and open a new empty map.
- void **saveMap** ()
Save map as is. If unsafed map, use Save As.
- void **saveMapAs** ()
Save map as a new map.
- void **saveWorld** ()
Save world as is.
- void **saveWorldAs** ()
Save a copy of the world and its maps under a new path.
- void **loadWorld** ()
Load an existing world and its designated first map.
- void **useUndo** ()
Undo last command.
- void **useRedo** ()
Redo last undone command.
- void **showMapProperties** ()
Shows the map properties dialog to edit them.
- void **showWorldProperties** ()
Shows the world properties dialog to edit the world.
- void **zoom** (QAction *zoomAction)
Zoom the map.
- void **quit** ()
***Quit** (p. 261) editor.*
- void **about** ()
Show an about dialog.

Public Member Functions

- **EditorMenu** (Processor *processor, EditorModel *editorModel, WorldModel *worldModel, QWidget *parent=0)
- void **updateSize** (int horizontal, int vertical, int maxHeight)

Private Attributes

- **Processor** * **processor**
- **EditorModel** * **editorModel**
- **WorldModel** * **worldModel**
- **MapPropertiesDialog** **mapPropertiesDialog**
- **WorldPropertiesDialog** **worldPropertiesDialog**
- QMenu * **fileMenu**
- QAction * **newWorldAction**
- QAction * **saveMapAction**
- QAction * **saveMapAsAction**
- QAction * **saveWorldAction**
- QAction * **saveWorldAsAction**
- QAction * **loadWorldAction**
- QAction * **quitAction**
- QMenu * **editMenu**
- QAction * **undoAction**
- QAction * **redoAction**
- QAction * **mapPropertiesAction**
- QAction * **worldPropertiesAction**
- QMenu * **viewMenu**
- QMenu * **zoomMenu**
- QActionGroup * **zoomActionGroup**
- QMenu * **helpMenu**
- QAction * **aboutAction**

8.35.1 Detailed Description

Represents the menubar of the editor.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.35.2 Constructor & Destructor Documentation

8.35.2.1 `EditorMenu::EditorMenu (Processor * processor, EditorModel * editorModel, WorldModel * worldModel, QWidget * parent = 0)` `[explicit]`

8.35.3 Member Function Documentation

8.35.3.1 `void EditorMenu::about ()` `[slot]`

Show an about dialog.

8.35.3.2 `void EditorMenu::createNewWorld ()` `[slot]`

Create and save a new empty World, load it into the world model and open a new empty map.

8.35.3.3 `void EditorMenu::loadWorld () [slot]`

Load an existing world and its designated first map.

8.35.3.4 `void EditorMenu::quit () [slot]`

Quit (p. 261) editor.

8.35.3.5 `void EditorMenu::saveMap () [slot]`

Save map as is. If unsafed map, use Save As.

8.35.3.6 `void EditorMenu::saveMapAs () [slot]`

Save map as a new map.

8.35.3.7 `void EditorMenu::saveWorld () [slot]`

Save world as is.

8.35.3.8 `void EditorMenu::saveWorldAs () [slot]`

Save a copy of the world and its maps under a new path.

8.35.3.9 `void EditorMenu::showMapProperties () [slot]`

Shows the map properties dialog to edit them.

8.35.3.10 `void EditorMenu::showWorldProperties () [slot]`

Shows the world properties dialog to edit the world.

8.35.3.11 `void EditorMenu::updateSize (int horizontal, int vertical, int maxHeight)`

8.35.3.12 `void EditorMenu::useRedo () [slot]`

Redo last undone command.

8.35.3.13 `void EditorMenu::useUndo () [slot]`

Undo last command.

8.35.3.14 `void EditorMenu::zoom (QAction * zoomAction) [slot]`

Zoom the map.

8.35.4 Member Data Documentation

- 8.35.4.1 `QAction* EditorMenu::aboutAction` [private]
- 8.35.4.2 `QMenu* EditorMenu::editMenu` [private]
- 8.35.4.3 `EditorModel* EditorMenu::editorModel` [private]
- 8.35.4.4 `QMenu* EditorMenu::fileMenu` [private]
- 8.35.4.5 `QMenu* EditorMenu::helpMenu` [private]
- 8.35.4.6 `QAction* EditorMenu::loadWorldAction` [private]
- 8.35.4.7 `QAction* EditorMenu::mapPropertiesAction` [private]
- 8.35.4.8 `MapPropertiesDialog EditorMenu::mapPropertiesDialog` [private]
- 8.35.4.9 `QAction* EditorMenu::newWorldAction` [private]
- 8.35.4.10 `Processor* EditorMenu::processor` [private]
- 8.35.4.11 `QAction* EditorMenu::quitAction` [private]
- 8.35.4.12 `QAction* EditorMenu::redoAction` [private]
- 8.35.4.13 `QAction* EditorMenu::saveMapAction` [private]
- 8.35.4.14 `QAction* EditorMenu::saveMapAsAction` [private]
- 8.35.4.15 `QAction* EditorMenu::saveWorldAction` [private]
- 8.35.4.16 `QAction* EditorMenu::saveWorldAsAction` [private]
- 8.35.4.17 `QAction* EditorMenu::undoAction` [private]
- 8.35.4.18 `QMenu* EditorMenu::viewMenu` [private]
- 8.35.4.19 `WorldModel* EditorMenu::worldModel` [private]
- 8.35.4.20 `QAction* EditorMenu::worldPropertiesAction` [private]
- 8.35.4.21 `WorldPropertiesDialog EditorMenu::worldPropertiesDialog` [private]
- 8.35.4.22 `QActionGroup* EditorMenu::zoomActionGroup` [private]
- 8.35.4.23 `QMenu* EditorMenu::zoomMenu` [private]

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

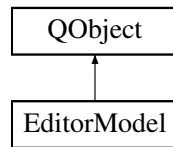
- `/home/ryoga/git/pidgirl/pidgirl-engine/Editor/EditorMenu.h`
- `/home/ryoga/git/pidgirl/pidgirl-engine/Editor/EditorMenu.cpp`

8.36 EditorModel Class Reference

Represents the editor state.

```
#include <EditorModel.h>
```

Inheritance diagram for EditorModel:



Public Member Functions

- **EditorModel** (bool **isEditor**, QObject *parent=0)
- bool **getIsEditor** () const
- void **setMainToolObject** (ObjectBase *object)
- ObjectBase * **refMainObject** ()

Private Attributes

- bool **isEditor**
Is true, if the program is used as a editor.
- ObjectBase * **mainToolObject**
The object that is to be used, when setting the object.

8.36.1 Detailed Description

Represents the editor state.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.36.2 Constructor & Destructor Documentation

8.36.2.1 **EditorModel::EditorModel** (bool *isEditor*, QObject * *parent* = 0) [explicit]

8.36.3 Member Function Documentation

8.36.3.1 bool **EditorModel::getIsEditor** () const

8.36.3.2 ObjectBase * **EditorModel::refMainObject** ()

8.36.3.3 void **EditorModel::setMainToolObject** (ObjectBase * *object*)

8.36.4 Member Data Documentation

8.36.4.1 `bool EditorModel::isEditor` [private]

Is true, if the programm ist used as a editor.

8.36.4.2 `ObjectBase* EditorModel::mainToolObject` [private]

The object that is to be used, when setting the object.

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

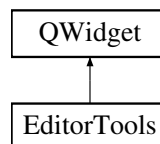
- /home/ryoga/git/pidgirl/pidgirl-engine/Editor/Model/**EditorModel.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Editor/Model/**EditorModel.cpp**

8.37 EditorTools Class Reference

The tools to be used in the editor.

```
#include <EditorTools.h>
```

Inheritance diagram for EditorTools:



Public Member Functions

- **EditorTools** (**Processor** *processor, **EditorModel** *editorModel, **QWidget** *parent=0)

Private Slots

- void **setMainTool** (int button)

Private Member Functions

- void **initializeToolButtons** (**PIDGIRL::MainToolButton** buttonID, int x, int y)
Puts a new button in the grid layout.
- void **addButton** (const **QString** &iconPath)
Append a new Button showing a given icon.

Private Attributes

- **Processor** * processor
- **EditorModel** * editorModel
- **QButtonGroup** toolButtons
All the possible objects to be chosen.
- **QVector**< **QToolButton** * > buttonVector
The list of buttons to click on to choose tools.

- `QGridLayout * gridLayout`

Grid Layout for the buttons to choose a tool.

8.37.1 Detailed Description

The tools to be used in the editor.

Note

To add more Buttons, just add one to the Editor Model MainToolButton enum and the adding of ButtonVector. Showing a different amount of Buttons per row, you'll have to modify the MAX_TOOL_BUTTONS_IN_ROW in **Common.h** (p. 418). Should be changed later to just all the possible objects.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.37.2 Constructor & Destructor Documentation

8.37.2.1 `EditorTools::EditorTools (Processor * processor, EditorModel * editorModel, QWidget * parent = 0)`
[explicit]

8.37.3 Member Function Documentation

8.37.3.1 `void EditorTools::addButton (const QString & iconPath)` [private]

Append a new Button showing a given icon.

Parameters

<i>iconPath</i>	The path to the icon to be used as button.
-----------------	--

8.37.3.2 `void EditorTools::initializeToolButtons (PIDGIRL::MainToolButton buttonID, int x, int y)` [private]

Puts a new button in the grid layout.

Parameters

<i>buttonID</i>	Enum that represents the button used when selecting this button.
<i>x</i>	X coordinate inside the grid layout.
<i>y</i>	Y coordinate inside the grid layout.

8.37.3.3 `void EditorTools::setMainTool (int button)` [private], [slot]

8.37.4 Member Data Documentation

8.37.4.1 `QVector<QToolButton *> EditorTools::buttonVector` [private]

The list of buttons to click on to choose tools.

8.37.4.2 **EditorModel*** EditorTools::editorModel [private]

8.37.4.3 **QGridLayout*** EditorTools::gridLayout [private]

Grid Layout for the buttons to choose a tool.

8.37.4.4 **Processor*** EditorTools::processor [private]

8.37.4.5 **QButtonGroup** EditorTools::toolButtons [private]

All the possible objects to be chosen.

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

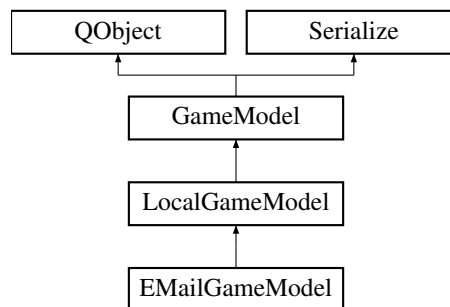
- /home/ryoga/git/pidgirl/pidgirl-engine/Editor/**EditorTools.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Editor/**EditorTools.cpp**

8.38 EMailGameModel Class Reference

The model representing an E-Mail game.

```
#include <EMailGameModel.h>
```

Inheritance diagram for EMailGameModel:



Public Member Functions

- **EMailGameModel (Processor *processor, QObject *parent)**
- virtual bool **nextTurn ()**
Prepare for the next turn.
- virtual bool **beginTurn ()**
Begin the turn.
- virtual bool **firstTurn ()**
Prepare for the first turn.
- virtual bool **endTurn ()**
End your turn.
- virtual void **convertFromDifferentGameModel (GameModel *gameModel)**
Update this game model with a game model of different type.
- virtual **PIDGIRL::GameType** **getGameType ()** const
- unsigned int **getTurnCount ()** const
- void **setTurnCount** (unsigned int **turnCount**)
- **MailGameInfo *** **refMailGameInfo** (unsigned int **playerID**)
- virtual void **serialize** (QIODevice *device) const

- void **serialize** (QDataStream &stream) const
- virtual void **deserialize** (QIODevice *device)
- void **deserialize** (QDataStream &stream)

Private Member Functions

- void **nextPlayerID** ()

Private Attributes

- QMap< unsigned int, **MailGameInfo** * > **mailGameInfos**
- unsigned int **turnCount**

Additional Inherited Members

8.38.1 Detailed Description

The model representing an E-Mail game.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.38.2 Constructor & Destructor Documentation

8.38.2.1 **EMailGameModel::EMailGameModel** (**Processor** * *processor*, **QObject** * *parent*)

8.38.3 Member Function Documentation

8.38.3.1 **bool EMailGameModel::beginTurn** () [virtual]

Begin the turn.

Implements **GameModel** (p. 137).

8.38.3.2 **void EMailGameModel::convertFromDifferentGameModel** (**GameModel** * *gameModel*) [virtual]

Update this game model with a game model of different type.

Parameters

<i>gameModel</i>	Game Model with a different type than the one being updated.
------------------	--

Reimplemented from **LocalGameModel** (p. 160).

8.38.3.3 **void EMailGameModel::deserialize** (**QIODevice** * *device*) [virtual]

Implements **Serialize** (p. 314).

8.38.3.4 `void EMailGameModel::deserialize (QDataStream & stream) [virtual]`

Reimplemented from **LocalGameModel** (p. 160).

8.38.3.5 `bool EMailGameModel::endTurn () [virtual]`

End your turn.

Implements **GameModel** (p. 138).

8.38.3.6 `bool EMailGameModel::firstTurn () [virtual]`

Prepare for the first turn.

Implements **GameModel** (p. 138).

8.38.3.7 `PIDGIRL::GameType EMailGameModel::getGameType () const [virtual]`

Implements **GameModel** (p. 138).

8.38.3.8 `unsigned int EMailGameModel::getTurnCount () const`

8.38.3.9 `void EMailGameModel::nextPlayerID () [private]`

8.38.3.10 `bool EMailGameModel::nextTurn () [virtual]`

Prepare for the next turn.

Implements **GameModel** (p. 140).

8.38.3.11 `MailGameInfo * EMailGameModel::refMailGameInfo (unsigned int playerID)`

8.38.3.12 `void EMailGameModel::serialize (QIODevice * device) const [virtual]`

Implements **Serialize** (p. 314).

8.38.3.13 `void EMailGameModel::serialize (QDataStream & stream) const [virtual]`

Reimplemented from **LocalGameModel** (p. 161).

8.38.3.14 `void EMailGameModel::setTurnCount (unsigned int turnCount)`

8.38.4 Member Data Documentation

8.38.4.1 `QMap<unsigned int, MailGameInfo *> EMailGameModel::mailGameInfos [private]`

8.38.4.2 `unsigned int EMailGameModel::turnCount [private]`

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

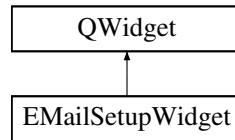
- `/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/EMailGameModel.h`
- `/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/EMailGameModel.cpp`

8.39 EMailSetupWidget Class Reference

The widget to set up an E-Mail Game.

```
#include <EMailSetupWidget.h>
```

Inheritance diagram for EMailSetupWidget:



Public Member Functions

- **EMailSetupWidget** (QWidget *parent=0)

Public Attributes

- QGridLayout * **emailWidgetLayout**
- QLabel * **playerCountLabel**
- QSpinBox * **playerCountSpinBox**
- QPushButton * **gameStartButton**

8.39.1 Detailed Description

The widget to set up an E-Mail Game.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.39.2 Constructor & Destructor Documentation

8.39.2.1 **EMailSetupWidget::EMailSetupWidget** (QWidget * *parent* = 0) [explicit]

8.39.3 Member Data Documentation

8.39.3.1 QGridLayout* EMailSetupWidget::emailWidgetLayout

8.39.3.2 QPushButton* EMailSetupWidget::gameStartButton

8.39.3.3 QLabel* EMailSetupWidget::playerCountLabel

8.39.3.4 QSpinBox* EMailSetupWidget::playerCountSpinBox

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

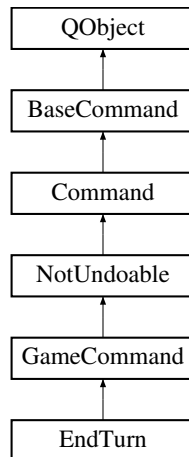
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/**EMailSetupWidget.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/**EMailSetupWidget.cpp**

8.40 EndTurn Class Reference

Things to do at the end of the turn and prepare to start the next turn.

```
#include <EndTurn.h>
```

Inheritance diagram for EndTurn:



Public Member Functions

- **EndTurn** (QObject *parent=0)
- virtual bool **execute** ()
Executes the command.

Additional Inherited Members

8.40.1 Detailed Description

Things to do at the end of the turn and prepare to start the next turn.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.40.2 Constructor & Destructor Documentation

8.40.2.1 EndTurn::EndTurn (QObject * parent = 0)

8.40.3 Member Function Documentation

8.40.3.1 bool EndTurn::execute () [virtual]

Executes the command.

Returns

True, if successful.

Implements **GameCommand** (p. 133).

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

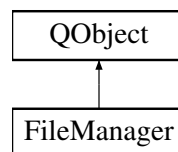
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/**EndTurn.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/**EndTurn.cpp**

8.41 FileManager Class Reference

Manages files to open and close accordingly and other things.

```
#include <FileManager.h>
```

Inheritance diagram for FileManager:



Public Member Functions

- **FileManager** (QObject *parent=0)
- bool **fileExists** ()
Check if file exists.
- bool **fileRemove** ()
Remove file if it exists.
- bool **fileOpen** (QIODevice::OpenModeFlag openFlag)
Open a handle to the file.
- bool **fileClose** ()
Close a handle to the file.
- void **setFilePath** (const QString &path)
Set a new file path as a target.
- bool **serialize** (**Serialize** *serializable, const QString &path)
***Serialize** (p. 313) datatype to path.*
- bool **deserialize** (**Serialize** *serializable, const QString &path)
***Serialize** (p. 313) datatype to path.*
- QFile * **refFile** ()

Private Attributes

- QFile **file**
The file which is managed.

8.41.1 Detailed Description

Manages files to open and close accordingly and other things.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.41.2 Constructor & Destructor Documentation

8.41.2.1 `FileManager::FileManager (QObject * parent = 0)`

8.41.3 Member Function Documentation

8.41.3.1 `bool FileManager::deserialize (Serialize * serializable, const QString & path)`

Serialize (p. 313) datatype to path.

Parameters

<i>serializable</i>	The datatype to be deserialized.
<i>path</i>	The path from where to deserialize.

8.41.3.2 `bool FileManager::fileClose ()`

Close a handle to the file.

Returns

True, if successful.

8.41.3.3 `bool FileManager::fileExists ()`

Check if file exists.

Returns

True, if exists.

8.41.3.4 `bool FileManager::fileOpen (QIODevice::OpenModeFlag openFlag)`

Open a handle to the file.

Returns

True, if successful.

8.41.3.5 bool FileManager::fileRemove ()

Remove file if it exists.

Returns

True, if successful.

8.41.3.6 QFile * FileManager::refFile ()

8.41.3.7 bool FileManager::serialize (Serialize * *serializable*, const QString & *path*)

Serialize (p. 313) datatype to path.

Parameters

<i>serializable</i>	The datatype to be serialized.
<i>path</i>	The path where to serialize.

8.41.3.8 void FileManager::setFilePath (const QString & *path*)

Set a new file path as a target.

Parameters

<i>path</i>	The path of the target.
-------------	-------------------------

8.41.4 Member Data Documentation

8.41.4.1 QFile FileManager::file [private]

The file which is managed.

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

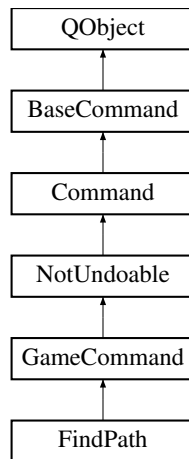
- /home/ryoga/git/pidgirl/pidgirl-engine/Common/**FileManager.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Common/**FileManager.cpp**

8.42 FindPath Class Reference

Creates a path that shows how to get from one stack to another.

```
#include <FindPath.h>
```

Inheritance diagram for FindPath:



Public Member Functions

- **FindPath** (`StackModel *from`, `StackModel *to`, `QObject *parent=0`)
- virtual bool **execute** ()

Executes the command.

Private Attributes

- **StackModel * from**
The stack from where to start the path from.
- **StackModel * to**
The stack to where the path leads to.
- **AStarMap aStarMap**
The A Map where the path will be calculated with.*

Additional Inherited Members

8.42.1 Detailed Description

Creates a path that shows how to get from one stack to another.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.42.2 Constructor & Destructor Documentation

8.42.2.1 `FindPath::FindPath (StackModel * from, StackModel * to, QObject * parent = 0)`

8.42.3 Member Function Documentation

8.42.3.1 `bool FindPath::execute () [virtual]`

Executes the command.

Returns

True, if successful.

Implements **GameCommand** (p. 133).

8.42.4 Member Data Documentation

8.42.4.1 `AStarMap FindPath::aStarMap [private]`

The A* Map where the path will be calculated with.

8.42.4.2 `StackModel* FindPath::from [private]`

The stack from where to start the path from.

8.42.4.3 `StackModel* FindPath::to [private]`

The stack to where the path leads to.

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

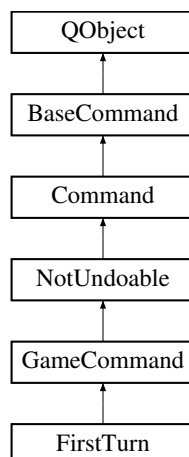
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Map/**FindPath.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Map/**FindPath.cpp**

8.43 FirstTurn Class Reference

Things to do for the very first turn.

```
#include <FirstTurn.h>
```

Inheritance diagram for FirstTurn:



Public Member Functions

- **FirstTurn** (QObject *parent=0)
- virtual bool **execute** ()
Executes the command.
- bool **init** ()

Private Slots

- void **indexChanged** (int index)
- void **done** ()

Private Attributes

- int **inactivePlayerChosen**
Is -1 if a new player should be created. Otherwise it's an index corresponding to the inactivePlayers vector.
- QMap< unsigned int, **PlayerModel** * > **players**
Copy of the player models currently available.
- QList< **Player** * > **inactivePlayers**
- QDialog **chooseCharacterDialog**
- **OptionsLayout** * **chooseCharacterLayout**
The amount of widgets in this layout is always choosableCharacterWidgets.size() + 1.
- QWidget * **newCharacterWidget**
- QGridLayout * **newCharacterLayout**
- QVector< QWidget * > **choosableCharacterWidgets**
Choosable Characters. Characters of players that are inactive are choosable.

Additional Inherited Members

8.43.1 Detailed Description

Things to do for the very first turn.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.43.2 Constructor & Destructor Documentation

8.43.2.1 FirstTurn::FirstTurn (QObject * parent = 0)

8.43.3 Member Function Documentation

8.43.3.1 void FirstTurn::done () [private],[slot]

8.43.3.2 bool FirstTurn::execute () [virtual]

Executes the command.

Returns

True, if successful.

Implements **GameCommand** (p. 133).

8.43.3.3 void FirstTurn::indexChanged (int *index*) [private], [slot]

8.43.3.4 bool FirstTurn::init ()

8.43.4 Member Data Documentation

8.43.4.1 QVector<QWidget*> FirstTurn::choosableCharacterWidgets [private]

Choosable Characters. Characters of players that are inactive are choosable.

8.43.4.2 QDialog FirstTurn::chooseCharacterDialog [private]

8.43.4.3 OptionsLayout* FirstTurn::chooseCharacterLayout [private]

The amount of widgets in this layout is always choosableCharacterWidgets.size() + 1.

8.43.4.4 int FirstTurn::inactivePlayerChosen [private]

Is -1 if a new player should be created. Otherwise it's an index corresponding to the inactivePlayers vector.

8.43.4.5 QList<Player*> FirstTurn::inactivePlayers [private]

8.43.4.6 QGridLayout* FirstTurn::newCharacterLayout [private]

8.43.4.7 QWidget* FirstTurn::newCharacterWidget [private]

8.43.4.8 QMap<unsigned int, PlayerModel*> FirstTurn::players [private]

Copy of the player models currently available.

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

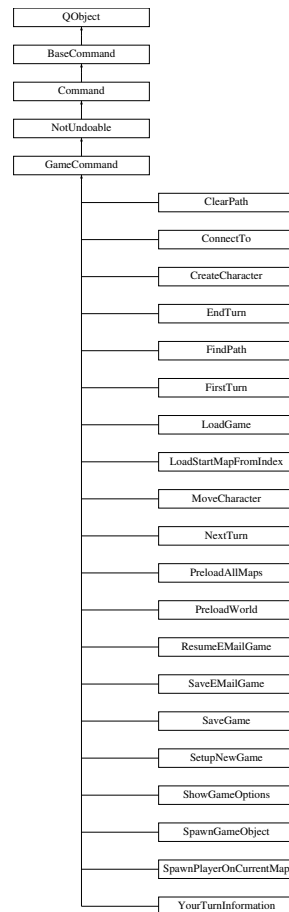
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/**FirstTurn.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/**FirstTurn.cpp**

8.44 GameCommand Class Reference

Interface for a command done in games.

```
#include <GameCommand.h>
```

Inheritance diagram for GameCommand:



Public Member Functions

- **GameCommand** (QObject *parent=0)
- virtual bool **isExecutable** () const
Check if the command is executable.
- virtual bool **execute** ()=0
Executes the command.

Protected Slots

- bool **quit** ()
***Quit** (p. 261) the game.*

Additional Inherited Members

8.44.1 Detailed Description

Interface for a command done in games.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.44.2 Constructor & Destructor Documentation

8.44.2.1 `GameCommand::GameCommand (QObject * parent = 0)`

8.44.3 Member Function Documentation

8.44.3.1 `virtual bool GameCommand::execute () [pure virtual]`

Executes the command.

Returns

True, if successful.

Implements **NotUndoable** (p. 215).

Implemented in **CreateCharacter** (p. 109), **SetupNewGame** (p. 325), **ShowGameOptions** (p. 330), **ConnectTo** (p. 100), **LoadGame** (p. 150), **FirstTurn** (p. 130), **SaveEmailGame** (p. 297), **FindPath** (p. 129), **MoveCharacter** (p. 199), **SpawnGameObject** (p. 333), **PreloadWorld** (p. 257), **ResumeEmailGame** (p. 293), **PreloadAllMaps** (p. 255), **SaveGame** (p. 299), **SpawnPlayerOnCurrentMap** (p. 335), **LoadStartMapFromIndex** (p. 155), **EndTurn** (p. 125), **NextTurn** (p. 213), **YourTurnInformation** (p. 374), and **ClearPath** (p. 78).

8.44.3.2 `bool GameCommand::isExecutable () const [virtual]`

Check if the command is executable.

Returns

If the command is executable.

Reimplemented from **NotUndoable** (p. 216).8.44.3.3 `bool GameCommand::quit () [protected], [slot]`**Quit** (p. 261) the game.

Returns

True, if the game was quit. Otherwise, false.

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

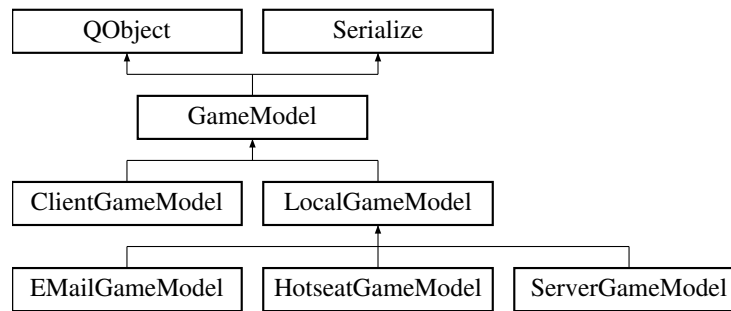
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/**GameCommand.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/**GameCommand.cpp**

8.45 GameModel Class Reference

The model representing the game preferences and players.

#include <GameModel.h>

Inheritance diagram for GameModel:



Signals

- void **setNewMapSignal** (**MapModel** *)
Send a **MapModel** (p. 173) out to be set.

Public Member Functions

- **GameModel** (QObject *parent)
- virtual bool **addNewPlayer** (unsigned int **playerID**, **PlayerModel** *playerModel)=0
Add a new player to the game.
- virtual void **addtoPreloadedMaps** (const QString &localMapPath, **MapModel** *map)=0
Add a preloaded map to the preloaded maps.
- virtual bool **loadMap** (const QString &localMapPath)=0
Load a map from a local map path.
- virtual bool **loadMap** (**MapModel** *map)
Load a map directly from a map model.
- virtual bool **spawnCharacter** (unsigned int **playerID**, **MapModel** *map, const **ObjectCharacter** &new-Character)
Spawn a given character on a given map.
- virtual bool **updateCharacter** (unsigned int **playerID**, **MapModel** *map, const **ObjectCharacter** &old-Character, const **ObjectCharacter** &newCharacter)
Update a given character with the information of a new character on a given map.
- virtual void **deactivatePlayer** (unsigned int **playerID**)=0
Deactivate a player.
- virtual bool **firstTurn** ()=0
Prepare for the first turn.
- virtual bool **nextTurn** ()=0
Prepare for the next turn.
- virtual bool **beginTurn** ()=0
Begin the turn.
- virtual bool **endTurn** ()=0
End your turn.
- void **deletePlayerModels** ()
Delete the current player models.
- virtual void **changePlayerID** (unsigned int newID)
Change the player ID.
- virtual void **convertFromDifferentGameModel** (**GameModel** *gameModel)
Update this game model with a game model of different type.
- virtual **PIDGIRL::GameType** **getGameType** () const =0

- virtual bool **isYourTurn** () const =0
- bool **getPlayerChoosesStartLocation** () const
- unsigned int **getPlayerID** () const
- virtual int **getCurrentPlayerID** () const
- virtual QMap< unsigned int, **PlayerModel** * > **getPlayerModels** () const
- virtual QMap< QString, **MapModel** * > **getPreloadedMaps** () const =0
- void **setPlayerChoosesStartLocation** (bool **playerChoosesStartLocation**)
- void **setPlayerID** (unsigned int **playerID**)
- virtual void **setCurrentMap** (**MapModel** *map)=0
- void **setPlayerModels** (const QMap< unsigned int, **PlayerModel** * > **playerModels**)
- void **setRequestedData** (QVariant data)
- void **setRequestedObject** (QObject *object)
- virtual **MapModel** * **refPreloadedMap** (const QString &localMapPath) const =0
- virtual **MapModel** * **refCurrentMap** () const =0
- virtual **PlayerModel** * **refPlayerModel** (unsigned int **playerID**) const =0
- virtual **Player** * **refPlayer** (unsigned int **playerID**) const =0
- virtual void **serialize** (QDataStream &dataStream) const
- virtual void **deserialize** (QDataStream &dataStream)

Protected Attributes

- QVariant **requestedData**
Requested Data QVariant. Everything that is not an object, has to be saved here.
- QObject * **requestedObject**
For all the objects which inherit from QObject.
- QMap< unsigned int, **PlayerModel** * > **playerModels**
The players registered in the game. The key is the player's ID.
- bool **playerChoosesStartLocation**
True, if players can choose their own start location.
- unsigned int **playerID**
Your player ID.

8.45.1 Detailed Description

The model representing the game preferences and players.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.45.2 Constructor & Destructor Documentation

8.45.2.1 `GameModel::GameModel (QObject * parent)` `[explicit]`

8.45.3 Member Function Documentation

8.45.3.1 `virtual bool GameModel::addNewPlayer (unsigned int playerID, PlayerModel * playerModel)` `[pure virtual]`

Add a new player to the game.

Parameters

<i>playerID</i>	The playerID to be added.
<i>playerModel</i>	The player model representing the player to be added.

Returns

True, if the player has been added. Otherwise, false.

Implemented in **ClientGameModel** (p. 83), and **LocalGameModel** (p. 160).

8.45.3.2 `virtual void GameModel::addtoPreloadedMaps (const QString & localMapPath, MapModel * map) [pure virtual]`

Add a preloaded map to the preloaded maps.

Parameters

<i>localMapPath</i>	Local file path.
<i>map</i>	The preloaded map.

Implemented in **ClientGameModel** (p. 84), and **LocalGameModel** (p. 160).

8.45.3.3 `virtual bool GameModel::beginTurn () [pure virtual]`

Begin the turn.

Implemented in **ClientGameModel** (p. 84), **EmailGameModel** (p. 121), **ServerGameModel** (p. 318), and **HotseatGameModel** (p. 144).

8.45.3.4 `void GameModel::changePlayerID (unsigned int newID) [virtual]`

Change the player ID.

Parameters

<i>newID</i>	The new ID of the player.
--------------	---------------------------

Reimplemented in **ClientGameModel** (p. 84), and **HotseatGameModel** (p. 144).

8.45.3.5 `void GameModel::convertFromDifferentGameModel (GameModel * gameModel) [virtual]`

Update this game model with a game model of different type.

Parameters

<i>gameModel</i>	Game Model with a different type than the one being updated.
------------------	--

Reimplemented in **ClientGameModel** (p. 84), **ServerGameModel** (p. 318), **EmailGameModel** (p. 121), **LocalGameModel** (p. 160), and **HotseatGameModel** (p. 144).

8.45.3.6 `virtual void GameModel::deavtivatePlayer (unsigned int playerID) [pure virtual]`

Deactivate a player.

Parameters

<i>playerID</i>	The playerID of which to deactivate the character of.
-----------------	---

Implemented in **ClientGameModel** (p. 84), **ServerGameModel** (p. 318), and **LocalGameModel** (p. 160).

8.45.3.7 void GameModel::deletePlayerModels ()

Delete the current player models.

8.45.3.8 void GameModel::deserialize (QDataStream & *dataStream*) [virtual]

Reimplemented in **LocalGameModel** (p. 160), and **EmailGameModel** (p. 122).

8.45.3.9 virtual bool GameModel::endTurn () [pure virtual]

End your turn.

Implemented in **ClientGameModel** (p. 85), **EmailGameModel** (p. 122), **ServerGameModel** (p. 319), and **HotseatGameModel** (p. 145).

8.45.3.10 virtual bool GameModel::firstTurn () [pure virtual]

Prepare for the first turn.

Implemented in **ClientGameModel** (p. 85), **EmailGameModel** (p. 122), **ServerGameModel** (p. 319), and **HotseatGameModel** (p. 145).

8.45.3.11 int GameModel::getCurrentPlayerID () const [virtual]

Reimplemented in **ServerGameModel** (p. 319).

8.45.3.12 virtual PIDGIRL::GameType GameModel::getGameType () const [pure virtual]

Implemented in **ClientGameModel** (p. 85), **ServerGameModel** (p. 319), **EmailGameModel** (p. 122), and **HotseatGameModel** (p. 145).

8.45.3.13 bool GameModel::getPlayerChoosesStartLocation () const

8.45.3.14 unsigned int GameModel::getPlayerID () const

8.45.3.15 QMap< unsigned int, PlayerModel * > GameModel::getPlayerModels () const [virtual]

Reimplemented in **ClientGameModel** (p. 85).

8.45.3.16 virtual QMap<QString, MapModel *> GameModel::getPreloadedMaps () const [pure virtual]

Implemented in **ClientGameModel** (p. 85), and **LocalGameModel** (p. 161).

8.45.3.17 virtual bool GameModel::isYourTurn () const [pure virtual]

Implemented in **ClientGameModel** (p. 85), **ServerGameModel** (p. 319), and **LocalGameModel** (p. 161).

8.45.3.18 virtual bool GameModel::loadMap (const QString & *localMapPath*) [pure virtual]

Load a map from a local map path.

Parameters

<i>localMapPath</i>	The local map path from which to load a map.
---------------------	--

Returns

True, if the map has been loaded. Otherwise. false.

Implemented in **ClientGameModel** (p. 85), and **LocalGameModel** (p. 161).

8.45.3.19 `bool GameModel::loadMap (MapModel * map) [virtual]`

Load a map directly from a map model.

Parameters

<i>map</i>	The map model to be loaded.
------------	-----------------------------

Returns

True, if the map model has been loaded. Otherwise, false.

Reimplemented in **ClientGameModel** (p. 85).

8.45.3.20 `virtual bool GameModel::nextTurn () [pure virtual]`

Prepare for the next turn.

Implemented in **ClientGameModel** (p. 86), **EmailGameModel** (p. 122), **ServerGameModel** (p. 319), and **HotseatGameModel** (p. 145).

8.45.3.21 `virtual MapModel* GameModel::refCurrentMap () const [pure virtual]`

Implemented in **ClientGameModel** (p. 86), **LocalGameModel** (p. 161), and **ServerGameModel** (p. 319).

8.45.3.22 `virtual Player* GameModel::refPlayer (unsigned int playerId) const [pure virtual]`

Implemented in **ClientGameModel** (p. 86), and **LocalGameModel** (p. 161).

8.45.3.23 `virtual PlayerModel* GameModel::refPlayerModel (unsigned int playerId) const [pure virtual]`

Implemented in **ClientGameModel** (p. 86), and **LocalGameModel** (p. 161).

8.45.3.24 `virtual MapModel* GameModel::refPreloadedMap (const QString & localMapPath) const [pure virtual]`

Implemented in **ClientGameModel** (p. 86), and **LocalGameModel** (p. 161).

8.45.3.25 `void GameModel::serialize (QDataStream & dataStream) const [virtual]`

Reimplemented in **LocalGameModel** (p. 161), and **EmailGameModel** (p. 122).

8.45.3.26 `virtual void GameModel::setCurrentMap (MapModel * map) [pure virtual]`

Implemented in **ClientGameModel** (p. 86), and **LocalGameModel** (p. 161).

8.45.3.27 void GameModel::setNewMapSignal (**MapModel** *) [signal]

Send a **MapModel** (p. 173) out to be set.

8.45.3.28 void GameModel::setPlayerChoosesStartLocation (bool *playerChoosesStartLocation*)

8.45.3.29 void GameModel::setPlayerID (unsigned int *playerID*)

8.45.3.30 void GameModel::setPlayerModels (const QMap< unsigned int, **PlayerModel** * > *playerModels*)

8.45.3.31 void GameModel::setRequestedData (QVariant *data*)

8.45.3.32 void GameModel::setRequestedObject (QObject * *object*)

8.45.3.33 bool GameModel::spawnCharacter (unsigned int *playerID*, **MapModel** * *map*, const **ObjectCharacter** & *newCharacter*) [virtual]

Spawn a given character on a given map.

Parameters

<i>playerID</i>	The playerID of the player that wants to spawn a character.
<i>map</i>	The map where to spawn the character on.
<i>newCharacter</i>	The character to be spawned with given coordinates already set.

Returns

True, if the character has been spawned. Otherwise, false.

Reimplemented in **ClientGameModel** (p. 86), and **ServerGameModel** (p. 319).

8.45.3.34 bool GameModel::updateCharacter (unsigned int *playerID*, **MapModel** * *map*, const **ObjectCharacter** & *oldCharacter*, const **ObjectCharacter** & *newCharacter*) [virtual]

Update a given character with the information of a new character on a given map.

Parameters

<i>playerID</i>	The playerID of the player that wants to update a character.
<i>map</i>	The map where to update the character.
<i>oldCharacter</i>	The old character that will be updated.
<i>newCharacter</i>	The new character with the updated information.

Returns

True, if the character has been updated. Otherwise, false.

Reimplemented in **ClientGameModel** (p. 87), and **ServerGameModel** (p. 320).

8.45.4 Member Data Documentation

8.45.4.1 bool GameModel::playerChoosesStartLocation [protected]

True, if players can choose their own start location.

8.45.4.2 `unsigned int GameModel::playerID` [protected]

Your player ID.

8.45.4.3 `QMap< unsigned int, PlayerModel * > GameModel::playerModels` [protected]

The players registered in the game. The key is the player's ID.

8.45.4.4 `QVariant GameModel::requestedData` [protected]

Requested Data QVariant. Everything that is not an object, has to be saved here.

8.45.4.5 `QObject* GameModel::requestedObject` [protected]

For all the objects which inherit from QObject.

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

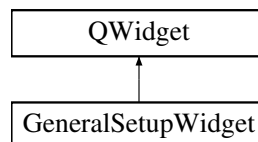
- `/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/GameModel.h`
- `/home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/GameModel.cpp`

8.46 GeneralSetupWidget Class Reference

The widget to set up general options for every game before it begins.

```
#include <GeneralSetupWidget.h>
```

Inheritance diagram for GeneralSetupWidget:



Public Member Functions

- **GeneralSetupWidget** (QWidget *parent=0)

Public Attributes

- QGridLayout * **generalSetupWidgetLayout**
- QCheckBox * **playerChoosesStartLocation**

8.46.1 Detailed Description

The widget to set up general options for every game before it begins.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.46.2 Constructor & Destructor Documentation

8.46.2.1 GeneralSetupWidget::GeneralSetupWidget (QWidget * *parent* = 0) [explicit]

8.46.3 Member Data Documentation

8.46.3.1 QGridLayout* GeneralSetupWidget::generalSetupWidgetLayout

8.46.3.2 QCheckBox* GeneralSetupWidget::playerChoosesStartLocation

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

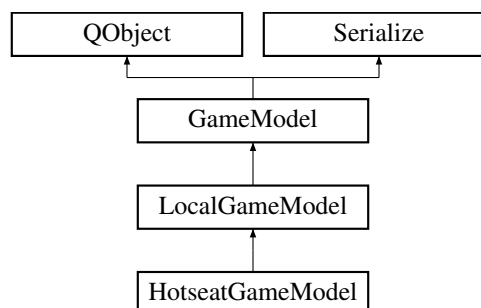
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/**GeneralSetupWidget.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/**GeneralSetupWidget.cpp**

8.47 HotseatGameModel Class Reference

The model representing a hotseat game with one or more players.

#include <HotseatGameModel.h>

Inheritance diagram for HotseatGameModel:



Public Member Functions

- **HotseatGameModel** (Processor ***processor**, QObject ***parent**)
- virtual bool **nextTurn** ()
Prepare for the next turn.
- virtual bool **beginTurn** ()
Begin the turn.
- virtual bool **firstTurn** ()
Prepare for the first turn.
- virtual bool **endTurn** ()
End your turn.
- virtual void **changePlayerID** (unsigned int **newID**)
Change the player ID.
- virtual void **convertFromDifferentGameModel** (GameModel ***gameModel**)

Update this game model with a game model of different type.

- virtual **PIDGIRL::GameType** **getGameType** () const
- virtual void **serialize** (QIODevice *device) const
- virtual void **deserialize** (QIODevice *device)

Private Member Functions

- bool **nextPlayerID** ()

Additional Inherited Members

8.47.1 Detailed Description

The model representing a hotseat game with one or more players.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.47.2 Constructor & Destructor Documentation

8.47.2.1 **HotseatGameModel::HotseatGameModel** (**Processor** * *processor*, **QObject** * *parent*)

8.47.3 Member Function Documentation

8.47.3.1 **bool** **HotseatGameModel::beginTurn** () [virtual]

Begin the turn.

Implements **GameModel** (p. 137).

8.47.3.2 **void** **HotseatGameModel::changePlayerID** (**unsigned int** *newID*) [virtual]

Change the player ID.

Parameters

<i>newID</i>	The new ID of the player.
--------------	---------------------------

Reimplemented from **GameModel** (p. 137).

8.47.3.3 **void** **HotseatGameModel::convertFromDifferentGameModel** (**GameModel** * *gameModel*) [virtual]

Update this game model with a game model of different type.

Parameters

<i>gameModel</i>	Game Model with a different type than the one being updated.
------------------	--

Reimplemented from **LocalGameModel** (p. 160).

8.47.3.4 `void HotseatGameModel::deserialize (QIODevice * device) [virtual]`

Implements **Serialize** (p. 314).

8.47.3.5 `bool HotseatGameModel::endTurn () [virtual]`

End your turn.

Implements **GameModel** (p. 138).

8.47.3.6 `bool HotseatGameModel::firstTurn () [virtual]`

Prepare for the first turn.

Implements **GameModel** (p. 138).

8.47.3.7 `PIDGIRL::GameType HotseatGameModel::getGameType () const [virtual]`

Implements **GameModel** (p. 138).

8.47.3.8 `bool HotseatGameModel::nextPlayerID () [private]`

8.47.3.9 `bool HotseatGameModel::nextTurn () [virtual]`

Prepare for the next turn.

Implements **GameModel** (p. 140).

8.47.3.10 `void HotseatGameModel::serialize (QIODevice * device) const [virtual]`

Implements **Serialize** (p. 314).

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

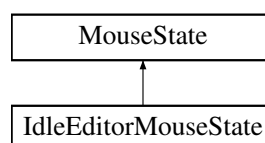
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/**HotseatGameModel.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/**HotseatGameModel.cpp**

8.48 IdleEditorMouseState Class Reference

The editor mouse state when nothing is being done.

```
#include <IdleEditorMouseState.h>
```

Inheritance diagram for IdleEditorMouseState:



Public Member Functions

- **IdleEditorMouseState** (**Processor** *processor, **StackModel** *targetStack)
- virtual **MouseState** * **nextState** (**MouseState::MouseEventType** type, **QMouseEvent** *event, **StackModel** *targetStack, **StackModel** *fromStack=0)

Determines which state is next, depending on the input. This method might destroy this state and changes occur.

Additional Inherited Members

8.48.1 Detailed Description

The editor mouse state when nothing is being done.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.48.2 Constructor & Destructor Documentation

8.48.2.1 **IdleEditorMouseState::IdleEditorMouseState** (**Processor** * processor, **StackModel** * targetStack)

8.48.3 Member Function Documentation

8.48.3.1 **MouseState** * **IdleEditorMouseState::nextState** (**MouseState::MouseEventType** type, **QMouseEvent** * event, **StackModel** * targetStack, **StackModel** * fromStack = 0) [virtual]

Determines which state is next, depending on the input. This method might destroy this state and changes occur.

Parameters

<i>event</i>	Holds the event that has been triggered.
<i>stack</i>	The stack where this method has been invoked from.

Returns

a pointer to the next state.

Implements **MouseState** (p. 197).

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

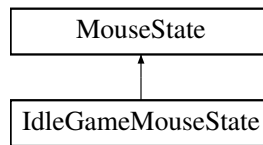
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/**IdleEditorMouseState.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/**IdleEditorMouseState.cpp**

8.49 IdleGameMouseState Class Reference

The game mouse state when nothing is being done.

```
#include <IdleGameMouseState.h>
```

Inheritance diagram for IdleGameMouseState:



Public Member Functions

- **IdleGameMouseState** (**Processor** *processor, **StackModel** *targetStack)
- virtual **MouseState** * **nextState** (**MouseState::MouseEventType** type, **QMouseEvent** *event, **StackModel** *targetStack, **StackModel** *fromStack=0)

Determines which state is next, depending on the input. This method might destroy this state and changes occur.

Additional Inherited Members

8.49.1 Detailed Description

The game mouse state when nothing is being done.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.49.2 Constructor & Destructor Documentation

8.49.2.1 **IdleGameMouseState::IdleGameMouseState** (**Processor** * processor, **StackModel** * targetStack)

8.49.3 Member Function Documentation

8.49.3.1 **MouseState** * **IdleGameMouseState::nextState** (**MouseState::MouseEventType** type, **QMouseEvent** * event, **StackModel** * targetStack, **StackModel** * fromStack = 0) [virtual]

Determines which state is next, depending on the input. This method might destroy this state and changes occur.

Parameters

<i>event</i>	Holds the event that has been triggered.
<i>stack</i>	The stack where this method has been invoked from.

Returns

a pointer to the next state.

Implements **MouseState** (p. 197).

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

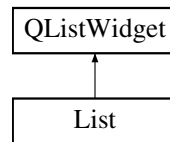
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/**IdleGameMouseState.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/**IdleGameMouseState.cpp**

8.50 List Class Reference

The list for widgets with reimplemented mouse events.

```
#include <OptionsLayout.h>
```

Inheritance diagram for List:



Public Member Functions

- **List** (QWidget *parent=0)

Protected Member Functions

- virtual void **mousePressEvent** (QMouseEvent *event)
- virtual void **mouseMoveEvent** (QMouseEvent *event)

8.50.1 Detailed Description

The list for widgets with reimplemented mouse events.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.50.2 Constructor & Destructor Documentation

8.50.2.1 **List::List** (QWidget * *parent* = 0) [inline]

8.50.3 Member Function Documentation

8.50.3.1 virtual void **List::mouseMoveEvent** (QMouseEvent * *event*) [inline],[protected],[virtual]

8.50.3.2 virtual void **List::mousePressEvent** (QMouseEvent * *event*) [inline],[protected],[virtual]

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

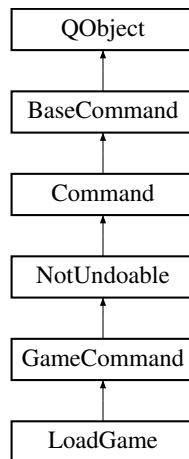
- /home/ryoga/git/pidgirl/pidgirl-engine/Common/**OptionsLayout.h**

8.51 LoadGame Class Reference

Load a game for **PIDGIRL** (p. 37) Engine.

```
#include <LoadGame.h>
```

Inheritance diagram for LoadGame:



Public Member Functions

- **LoadGame** (**GameModel** **gameModelPointer, **QObject** *parent=0)
- virtual bool **execute** ()
Executes the command.

Private Slots

- void **loadSoloHotseatGame** ()
- void **loadNetworkGame** ()

Private Member Functions

- bool **init** ()

Private Attributes

- **GameModel** ** gameModelPointer
The Game Model pointer from the Main Window to be set in this.
- **FileManager** fileManager
- **QString** saveFilePath
- **QDialog** loadSetupDialog
- **OptionsLayout** * loadSetupLayout
- **SoloHotseatSetupWidget** * soloHotseatSetupWidget
- **NetworkSetupWidget** * networkSetupWidget

Additional Inherited Members

8.51.1 Detailed Description

Load a game for **PIDGIRL** (p. 37) Engine.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.51.2 Constructor & Destructor Documentation

8.51.2.1 **LoadGame::LoadGame** (**GameModel** ** *gameModelPointer*, **QObject** * *parent* = 0)

8.51.3 Member Function Documentation

8.51.3.1 **bool** **LoadGame::execute** () [virtual]

Executes the command.

Returns

True, if successful.

Implements **GameCommand** (p. 133).

8.51.3.2 **bool** **LoadGame::init** () [private]

8.51.3.3 **void** **LoadGame::loadNetworkGame** () [private],[slot]

8.51.3.4 **void** **LoadGame::loadSoloHotseatGame** () [private],[slot]

8.51.4 Member Data Documentation

8.51.4.1 **FileManager** **LoadGame::fileManager** [private]

8.51.4.2 **GameModel**** **LoadGame::gameModelPointer** [private]

The Game Model pointer from the Main Window to be set in this.

8.51.4.3 **QDialog** **LoadGame::loadSetupDialog** [private]

8.51.4.4 **OptionsLayout*** **LoadGame::loadSetupLayout** [private]

8.51.4.5 **NetworkSetupWidget*** **LoadGame::networkSetupWidget** [private]

8.51.4.6 **QString** **LoadGame::saveFilePath** [private]

8.51.4.7 SoloHotseatSetupWidget* LoadGame::soloHotseatSetupWidget [private]

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

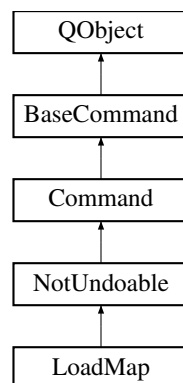
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/**LoadGame.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/**LoadGame.cpp**

8.52 LoadMap Class Reference

Load a map from a given path.

```
#include <LoadMap.h>
```

Inheritance diagram for LoadMap:



Public Member Functions

- **LoadMap** (const QString &**path**, bool **ask**=true, QObject *parent=0)
- virtual bool **execute** ()
Executes the command.

Private Attributes

- QString **path**
The path from which to load the map from.
- **FileManager** **fileManager**
The file manager to load the map with.
- bool **ask**
True, if before loading a new map it should ask for changes to be saved.

Additional Inherited Members

8.52.1 Detailed Description

Load a map from a given path.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.52.2 Constructor & Destructor Documentation

8.52.2.1 `LoadMap::LoadMap (const QString & path, bool ask = true, QObject * parent = 0)`

8.52.3 Member Function Documentation

8.52.3.1 `bool LoadMap::execute () [virtual]`

Executes the command.

Returns

True, if successful.

Implements **NotUndoable** (p. 215).

8.52.4 Member Data Documentation

8.52.4.1 `bool LoadMap::ask [private]`

True, if before loading a new map it should ask for changes to be saved.

8.52.4.2 `FileManager LoadMap::fileManager [private]`

The file manager to load the map with.

8.52.4.3 `QString LoadMap::path [private]`

The path from which to load the map from.

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

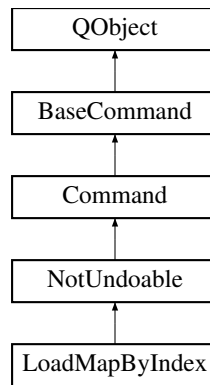
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/**LoadMap.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/**LoadMap.cpp**

8.53 LoadMapByIndex Class Reference

Load a map according to the given path in a model index.

```
#include <LoadMapByIndex.h>
```

Inheritance diagram for LoadMapByIndex:



Public Member Functions

- **LoadMapByIndex** (const QModelIndex &**modelIndex**, bool **ask**=true, QObject *parent=0)
- virtual bool **execute** ()
Executes the command.

Private Attributes

- QModelIndex **modelIndex**
The model index from which to load the map from.
- bool **ask**
True, if before loading a new map it should ask for changes to be saved.

Additional Inherited Members

8.53.1 Detailed Description

Load a map according to the given path in a model index.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.53.2 Constructor & Destructor Documentation

8.53.2.1 LoadMapByIndex::LoadMapByIndex (const QModelIndex & *modelIndex*, bool *ask* = true, QObject * *parent* = 0)

8.53.3 Member Function Documentation

8.53.3.1 bool LoadMapByIndex::execute () [virtual]

Executes the command.

Returns

True, if successful.

Implements **NotUndoable** (p. 215).

8.53.4 Member Data Documentation**8.53.4.1 bool LoadMapByIndex::ask** [private]

True, if before loading a new map it should ask for changes to be saved.

8.53.4.2 QModelIndex LoadMapByIndex::modelIndex [private]

The model index from which to load the map from.

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

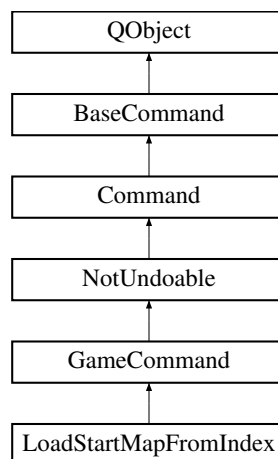
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/**LoadMapByIndex.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/**LoadMapByIndex.cpp**

8.54 LoadStartMapFromIndex Class Reference

Load the map depending on the index given. Can be random as well.

```
#include <LoadStartMapFromIndex.h>
```

Inheritance diagram for LoadStartMapFromIndex:

**Public Member Functions**

- **LoadStartMapFromIndex** (int **startMapIndex**, QObject *parent=0)
- virtual bool **execute** ()
Executes the command.

Private Attributes

- int **startMapIndex**
The index from which to load the start map from looked up in the world model. If -1 or anything invalid, choose a random map from the starter maps.

Additional Inherited Members

8.54.1 Detailed Description

Load the map depending on the index given. Can be random as well.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.54.2 Constructor & Destructor Documentation

8.54.2.1 LoadStartMapFromIndex::LoadStartMapFromIndex (int *startMapIndex*, QObject * *parent* = 0)

8.54.3 Member Function Documentation

8.54.3.1 bool LoadStartMapFromIndex::execute () [virtual]

Executes the command.

Returns

True, if successful.

Implements **GameCommand** (p. 133).

8.54.4 Member Data Documentation

8.54.4.1 int LoadStartMapFromIndex::startMapIndex [private]

The index from which to load the start map from looked up in the world model. If -1 or anything invalid, choose a random map from the starter maps.

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

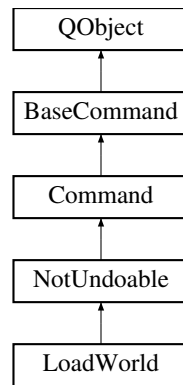
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/**LoadStartMapFromIndex.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/**LoadStartMapFromIndex.cpp**

8.55 LoadWorld Class Reference

Choose the name of a file and load the world from this file.

```
#include <LoadWorld.h>
```

Inheritance diagram for LoadWorld:



Public Member Functions

- **LoadWorld** (bool **resetToStandardMap**=true, bool **mayAsk**=true, `QObject *parent=0`)
- bool **execute** ()

Executes the command.

Private Member Functions

- bool **loadWorld** ()

Load a new world from a file loaded with the filemanager.

Private Attributes

- **FileManager** **fileManager**

The file manager to load the world with.

- bool **mayAsk**

True, if it should be asked for unsaved changes to be saved.

- bool **resetToStandardMap**

True, if a standard map is to be loaded to the current map model when loading a world.

Additional Inherited Members

8.55.1 Detailed Description

Choose the name of a file and load the world from this file.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.55.2 Constructor & Destructor Documentation

8.55.2.1 `LoadWorld::LoadWorld (bool resetToStandardMap = true, bool mayAsk = true, QObject * parent = 0)`

8.55.3 Member Function Documentation

8.55.3.1 `bool LoadWorld::execute () [virtual]`

Executes the command.

Returns

True, if successful.

Implements **NotUndoable** (p. 215).

8.55.3.2 `bool LoadWorld::loadWorld () [private]`

Load a new world from a file loaded with the filemanager.

Returns

True, if successful. Otherwise, false.

8.55.4 Member Data Documentation

8.55.4.1 `FileManager LoadWorld::fileManager [private]`

The file manager to load the world with.

8.55.4.2 `bool LoadWorld::mayAsk [private]`

True, if it should be asked for unsaved changes to be saved.

8.55.4.3 `bool LoadWorld::resetToStandardMap [private]`

True, if a standard map is to be loaded to the current map model when loading a world.

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

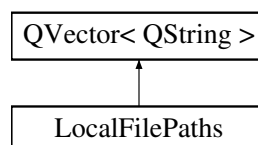
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/**LoadWorld.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/**LoadWorld.cpp**

8.56 LocalFilePaths Class Reference

Locally in world folder represented file paths.

```
#include <LocalFilePaths.h>
```

Inheritance diagram for LocalFilePaths:



Public Member Functions

- **LocalFilePaths** ()
- **LocalFilePaths** (int size)
- **LocalFilePaths** (const QString &localFilePath)
- **LocalFilePaths** (const QVector< QString > &v)
- QVector< QString > **convertToFileNameVector** ()
- QString **convertToFileName** (const QString &localFilePath)

8.56.1 Detailed Description

Locally in world folder represented file paths.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.56.2 Constructor & Destructor Documentation

8.56.2.1 **LocalFilePaths::LocalFilePaths** ()

8.56.2.2 **LocalFilePaths::LocalFilePaths** (int *size*)

8.56.2.3 **LocalFilePaths::LocalFilePaths** (const QString & *localFilePath*)

8.56.2.4 **LocalFilePaths::LocalFilePaths** (const QVector< QString > & *v*)

8.56.3 Member Function Documentation

8.56.3.1 **QString LocalFilePaths::convertToFileName** (const QString & *localFilePath*)

8.56.3.2 **QVector< QString > LocalFilePaths::convertToFileNameVector** ()

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

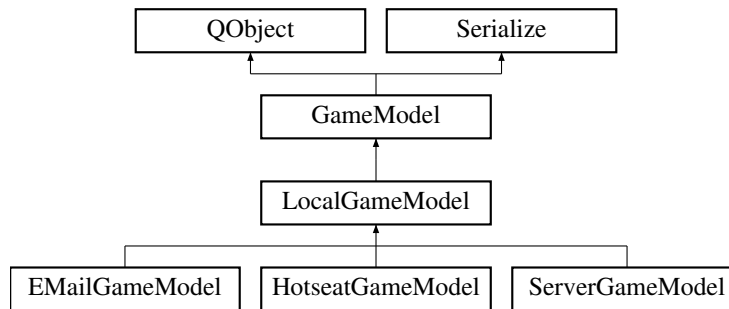
- /home/ryoga/git/pidgirl/pidgirl-engine/Editor/Model/**LocalFilePaths.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Editor/Model/**LocalFilePaths.cpp**

8.57 LocalGameModel Class Reference

The model representing a local game where all the information is held locally.

```
#include <LocalGameModel.h>
```

Inheritance diagram for LocalGameModel:



Public Member Functions

- **LocalGameModel** (**Processor** *processor, QObject *parent)
- virtual bool **addNewPlayer** (unsigned int **playerID**, **PlayerModel** *playerModel)
Add a new player to the game.
- virtual void **addtoPreloadedMaps** (const QString &localMapPath, **MapModel** *map)
Add a preloaded map to the preloaded maps.
- virtual bool **loadMap** (const QString &localMapPath)
Load a map from a local map path.
- virtual void **deactivatePlayer** (unsigned int **playerID**)
Deactivate a player.
- virtual void **convertFromDifferentGameModel** (**GameModel** *gameModel)
Update this game model with a game model of different type.
- virtual bool **isYourTurn** () const
- virtual QMap< QString, **MapModel** * > **getPreloadedMaps** () const
- virtual void **setCurrentMap** (**MapModel** *map)
- virtual **PlayerModel** * **refPlayerModel** (unsigned int **playerID**) const
- virtual **MapModel** * **refCurrentMap** () const
- virtual **MapModel** * **refPreloadedMap** (const QString &localMapPath) const
- virtual **Player** * **refPlayer** (unsigned int **playerID**) const
- virtual void **serialize** (QDataStream &dataStream) const
- virtual void **deserialize** (QDataStream &dataStream)

Protected Attributes

- **Processor** * **processor**
- QMap< QString, **MapModel** * > **preloadedMaps**
The map of preloaded maps. The key is the local map path for the map.

Additional Inherited Members

8.57.1 Detailed Description

The model representing a local game where all the information is held locally.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.57.2 Constructor & Destructor Documentation

8.57.2.1 `LocalGameModel::LocalGameModel (Processor * processor, QObject * parent)`

8.57.3 Member Function Documentation

8.57.3.1 `bool LocalGameModel::addNewPlayer (unsigned int playerID, PlayerModel * playerModel) [virtual]`

Add a new player to the game.

Parameters

<i>playerID</i>	The playerID to be added.
<i>playerModel</i>	The player model representing the player to be added.

Returns

True, if the player has been added. Otherwise, false.

Implements **GameModel** (p. 136).

8.57.3.2 `void LocalGameModel::addtoPreloadedMaps (const QString & localMapPath, MapModel * map) [virtual]`

Add a preloaded map to the preloaded maps.

Parameters

<i>localMapPath</i>	Local file path.
<i>map</i>	The preloaded map.

Implements **GameModel** (p. 137).

8.57.3.3 `void LocalGameModel::convertFromDifferentGameModel (GameModel * gameModel) [virtual]`

Update this game model with a game model of different type.

Parameters

<i>gameModel</i>	Game Model with a different type than the one being updated.
------------------	--

Reimplemented from **GameModel** (p. 137).

Reimplemented in **ServerGameModel** (p. 318), **EmailGameModel** (p. 121), and **HotseatGameModel** (p. 144).

8.57.3.4 `void LocalGameModel::deactivatePlayer (unsigned int playerID) [virtual]`

Deactivate a player.

Parameters

<i>playerID</i>	The playerID of which to deactivate the character of.
-----------------	---

Implements **GameModel** (p. 137).

Reimplemented in **ServerGameModel** (p. 318).

8.57.3.5 `void LocalGameModel::deserialize (QDataStream & dataStream) [virtual]`

Reimplemented from **GameModel** (p. 138).

Reimplemented in **EmailGameModel** (p. 122).

8.57.3.6 `QMap< QString, MapModel * > LocalGameModel::getPreloadedMaps () const` [virtual]

Implements **GameModel** (p. 138).

8.57.3.7 `bool LocalGameModel::isYourTurn () const` [virtual]

Implements **GameModel** (p. 138).

Reimplemented in **ServerGameModel** (p. 319).

8.57.3.8 `bool LocalGameModel::loadMap (const QString & localMapPath)` [virtual]

Load a map from a local map path.

Parameters

<i>localMapPath</i>	The local map path from which to load a map.
---------------------	--

Returns

True, if the map has been loaded. Otherwise. false.

Implements **GameModel** (p. 139).

8.57.3.9 `MapModel * LocalGameModel::refCurrentMap () const` [virtual]

Implements **GameModel** (p. 140).

Reimplemented in **ServerGameModel** (p. 319).

8.57.3.10 `Player * LocalGameModel::refPlayer (unsigned int playerId) const` [virtual]

Implements **GameModel** (p. 140).

8.57.3.11 `PlayerModel * LocalGameModel::refPlayerModel (unsigned int playerId) const` [virtual]

Implements **GameModel** (p. 140).

8.57.3.12 `MapModel * LocalGameModel::refPreloadedMap (const QString & localMapPath) const` [virtual]

Implements **GameModel** (p. 140).

8.57.3.13 `void LocalGameModel::serialize (QDataStream & dataStream) const` [virtual]

Reimplemented from **GameModel** (p. 140).

Reimplemented in **EmailGameModel** (p. 122).

8.57.3.14 `void LocalGameModel::setCurrentMap (MapModel * map)` [virtual]

Implements **GameModel** (p. 140).

8.57.4 Member Data Documentation

8.57.4.1 QMap< QString, MapModel *> LocalGameModel::preloadedMaps [protected]

The map of preloaded maps. The key is the local map path for the map.

8.57.4.2 Processor* LocalGameModel::processor [protected]

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

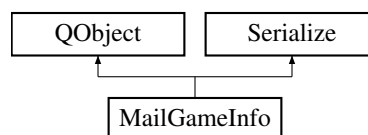
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/**LocalGameModel.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/**LocalGameModel.cpp**

8.58 MailGameInfo Class Reference

Holds all the informations about your E-Mail Game.

```
#include <MailGameInfo.h>
```

Inheritance diagram for MailGameInfo:



Public Slots

- void **setSaveDir** (const QString &saveDirAbsolutePath)

Public Member Functions

- **MailGameInfo** (QObject *parent=0)
- bool **initializeMailGameInfo** ()
- bool **authenticatePlayer** ()
 - *Check if the player requesting the authentication decrypts the cipherText correctly.*
- bool **hasEMailGameDirSet** () const
- QDir **getEMailGameDir** () const
- QString **getLocalEMailGamePath** () const
- std::string **getEncryptedText** () const
- bool **hasSaveDirSet** () const
- QDir **getSaveDir** () const
- QString **getSaveDirAbsolutePath** () const
- void **setLocalEMailGamePath** (const QString &localEMailGamePath)
- virtual void **serialize** (QIODevice *device) const
- void **serialize** (QDataStream &stream) const
- virtual void **deserialize** (QIODevice *device)
- void **deserialize** (QDataStream &stream)

Private Attributes

- bool **emailGameDirSet**
- QDir **emailGameDir**
- std::string **cipherText**
- bool **saveDirSet**
- QDir **saveDir**

8.58.1 Detailed Description

Holds all the informations about your E-Mail Game.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.58.2 Constructor & Destructor Documentation

8.58.2.1 MailGameInfo::MailGameInfo (QObject * *parent* = 0)

8.58.3 Member Function Documentation

8.58.3.1 bool MailGameInfo::authenticatePlayer ()

Check if the player requesting the authentication decrypts the cipherText correctly.

Returns

True, if you are the player authenticating. Otherwise, false.

8.58.3.2 void MailGameInfo::deserialize (QIODevice * *device*) [virtual]

Implements **Serialize** (p.314).

8.58.3.3 void MailGameInfo::deserialize (QDataStream & *stream*)

8.58.3.4 QDir MailGameInfo::getEmailGameDir () const

8.58.3.5 std::string MailGameInfo::getEncryptedText () const

8.58.3.6 QString MailGameInfo::getLocalEmailGamePath () const

8.58.3.7 QDir MailGameInfo::getSaveDir () const

8.58.3.8 QString MailGameInfo::getSaveDirAbsolutePath () const

8.58.3.9 bool MailGameInfo::hasEmailGameDirSet () const

8.58.3.10 `bool MailGameInfo::hasSaveDirSet () const`

8.58.3.11 `bool MailGameInfo::initializeMailGameInfo ()`

8.58.3.12 `void MailGameInfo::serialize (QIODevice * device) const` [virtual]

Implements **Serialize** (p.314).

8.58.3.13 `void MailGameInfo::serialize (QDataStream & stream) const`

8.58.3.14 `void MailGameInfo::setLocalEmailGamePath (const QString & localEmailGamePath)`

8.58.3.15 `void MailGameInfo::setSaveDir (const QString & saveDirAbsolutePath)` [slot]

8.58.4 Member Data Documentation

8.58.4.1 `std::string MailGameInfo::cipherText` [private]

8.58.4.2 `QDir MailGameInfo::emailGameDir` [private]

8.58.4.3 `bool MailGameInfo::emailGameDirSet` [private]

8.58.4.4 `QDir MailGameInfo::saveDir` [private]

8.58.4.5 `bool MailGameInfo::saveDirSet` [private]

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

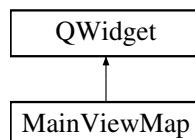
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/**MailGameInfo.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/**MailGameInfo.cpp**

8.59 MainViewMap Class Reference

The main view showing every stack on an grid.

```
#include <MainViewMap.h>
```

Inheritance diagram for MainViewMap:



Signals

- void **mapSizeUpdated** ()

Public Member Functions

- **MainViewMap** (**Processor** *processor, **MapModel** *mapModel, **EditorModel** *editorModel, **GameModel** *gameModel, QWidget *parent)

Protected Member Functions

- virtual void **paintEvent** (QPaintEvent *)
Overload paint event to make the mainview view draw itself everytime it needs to be updated.

Private Slots

- void **paintGrid** ()
Paints the grid.
- void **updateMap** ()
Update the complete stack grid.
- void **showContextMenu** (StackModel *stackModel)

Private Member Functions

- void **createStackGrid** (int x, int y)
Create an mainViewStack.
- void **deleteStackGrid** (int x, int y)
Delete and set to 0 one of the mainViewStack.

Private Attributes

- Processor * **processor**
- EditorModel * **editorModel**
- MapModel * **mapModel**
- GameModel * **gameModel**
- MousetInput * **mousetInput**
- QVector< QVector
 < **MainViewStack** * > > **stackGrid**

8.59.1 Detailed Description

The main view showing every stack on an grid.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.59.2 Constructor & Destructor Documentation

- 8.59.2.1 **MainViewMap::MainViewMap** (Processor * *processor*, MapModel * *mapModel*, EditorModel * *editorModel*, GameModel * *gameModel*, QWidget * *parent*) [explicit]

8.59.3 Member Function Documentation

- 8.59.3.1 void **MainViewMap::createStackGrid** (int x, int y) [private]

Create an mainViewStack.

Parameters

<i>x</i>	X Coordinate for the MainViewStack (p. 166).
<i>y</i>	Y Coordinate for the MainViewStack (p. 166).

8.59.3.2 `void MainViewMap::deleteStackGrid (int x, int y)` [private]

Delete and set to 0 one of the mainViewStack.

Parameters

<i>x</i>	X Coordinate for the MainViewStack (p. 166).
<i>y</i>	Y Coordinate for the MainViewStack (p. 166).

8.59.3.3 `void MainViewMap::mapSizeUpdated ()` [signal]

8.59.3.4 `void MainViewMap::paintEvent (QPaintEvent *)` [protected],[virtual]

Overload paint event to make the mainview view draw itself everytime it needs to be updated.

8.59.3.5 `void MainViewMap::paintGrid ()` [private],[slot]

Paints the grid.

8.59.3.6 `void MainViewMap::showContextMenu (StackModel * stackModel)` [private],[slot]

8.59.3.7 `void MainViewMap::updateMap ()` [private],[slot]

Update the complete stack grid.

8.59.4 Member Data Documentation

8.59.4.1 `EditorModel* MainViewMap::editorModel` [private]

8.59.4.2 `GameModel* MainViewMap::gameModel` [private]

8.59.4.3 `MapModel* MainViewMap::mapModel` [private]

8.59.4.4 `MouseInput* MainViewMap::mouseInput` [private]

8.59.4.5 `Processor* MainViewMap::processor` [private]

8.59.4.6 `QVector<QVector <MainViewStack *> > MainViewMap::stackGrid` [private]

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

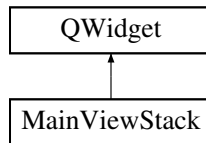
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/**MainViewMap.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/**MainViewMap.cpp**

8.60 MainViewStack Class Reference

Graphically represents a stack of a square in the game.

```
#include <MainViewStack.h>
```

Inheritance diagram for MainViewStack:



Public Slots

- void **updateAll** ()

Public Member Functions

- **MainViewStack** (**Processor** *processor, **StackModel** *stackModel, **MapModel** *mapModel, **EditorModel** *editorModel, **GameModel** *gameModel, **MouseInput** *mouseInput, **QWidget** *parent=0)
- void **drawObject** (**QPainter** *painter, **ObjectBase** *objectBase)
Draw a given object.
- void **drawDirectionIndicator** (**QPainter** *painter)
Draw a direction indicator if there is one.
- virtual void **paintEvent** (**QPaintEvent** *)
Override paint event to make the mainView Stack draw itself everytime it needs to be updated.
- void **setStackModel** (**StackModel** *stackModel)

Protected Member Functions

- virtual void **mousePressEvent** (**QMouseEvent** *event)
The events that occur when the mousebutton is pressed.
- virtual void **mouseMoveEvent** (**QMouseEvent** *event)
- virtual void **mouseReleaseEvent** (**QMouseEvent** *event)

Private Slots

- void **updateSize** ()

Private Attributes

- **Processor** * processor
- **StackModel** * stackModel
- **MapModel** * mapModel
- **EditorModel** * editorModel
- **GameModel** * gameModel
- **MouseInput** * mouseInput

8.60.1 Detailed Description

Graphically represents a stack of a square in the game.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.60.2 Constructor & Destructor Documentation

8.60.2.1 `MainViewStack::MainViewStack (Processor * processor, StackModel * stackModel, MapModel * mapModel, EditorModel * editorModel, GameModel * gameModel, MouseInput * mouseInput, QWidget * parent = 0)` `[explicit]`

8.60.3 Member Function Documentation

8.60.3.1 `void MainViewStack::drawDirectionIndicator (QPainter * painter)`

Draw a direction indicator if there is one.

8.60.3.2 `void MainViewStack::drawObject (QPainter * painter, ObjectBase * objectBase)`

Draw a given object.

Parameters

<i>painter</i>	The painter with which to paint the given object.
<i>objectBase</i>	The object to be painted.

8.60.3.3 `void MainViewStack::mouseMoveEvent (QMouseEvent * event)` `[protected]`, `[virtual]`

8.60.3.4 `void MainViewStack::mousePressEvent (QMouseEvent * event)` `[protected]`, `[virtual]`

The events that occur when the mousebutton is pressed.

8.60.3.5 `void MainViewStack::mouseReleaseEvent (QMouseEvent * event)` `[protected]`, `[virtual]`

8.60.3.6 `void MainViewStack::paintEvent (QPaintEvent *)` `[virtual]`

Override paint event to make the mainView Stack draw itself everytime it needs to be updated.

8.60.3.7 `void MainViewStack::setStackModel (StackModel * stackModel)`

8.60.3.8 `void MainViewStack::updateAll ()` `[slot]`

Parameters

<i>Update</i>	all buttons.
---------------	--------------

8.60.3.9 void `MainViewStack::updateSize ()` [private],[slot]

Parameters

<i>Update</i>	size of the stack.
---------------	--------------------

8.60.4 Member Data Documentation

8.60.4.1 `EditorModel*` `MainViewStack::editorModel` [private]

8.60.4.2 `GameModel*` `MainViewStack::gameModel` [private]

8.60.4.3 `MapModel*` `MainViewStack::mapModel` [private]

8.60.4.4 `MouseInput*` `MainViewStack::mouseInput` [private]

8.60.4.5 `Processor*` `MainViewStack::processor` [private]

8.60.4.6 `StackModel*` `MainViewStack::stackModel` [private]

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

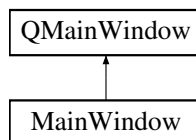
- /home/ryoga/git/pidgirl/pidgirl-engine/MainWindow/**MainViewStack.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/MainWindow/**MainViewStack.cpp**

8.61 MainWindow Class Reference

Holds all the models and GUIs and initializes them.

```
#include <MainWindow.h>
```

Inheritance diagram for MainWindow:



Signals

- void **windowSizeChanged** ()

Public Member Functions

- **MainWindow** (QWidget *parent=0)

Protected Member Functions

- virtual void **closeEvent** (QCloseEvent *event)

Private Slots

- void **updateSize** ()
Updates the size of the main window depending on the mapsize.
- void **updateEditorSize** ()
Updates the size of the main window depending on the mapsize and the dockwidgets.
- void **updateWindowTitle** ()
Updates the title of the window to whatever the current mapfilename is.
- void **createMainViewMap** ()
*Creates a new **MainViewMap** (p. 164).*
- void **setNewMap** (MapModel *mapModel)
Sets a new map.

Private Member Functions

- void **init** ()
Initialize the main window completely.
- bool **initializeEditor** ()
Initialize the editor correctly. Make user choose wether to load an existing editor or create a new one.
- bool **initializeGame** ()
Initialize Game, so the current map can be loaded. Make the user choose wether to start a new game, load an existing game (I.e.: E-Mail Game), set some options (if any) or exit the game.

Private Attributes

- **EditorModel** * **editorModel**
- **WorldModel** * **worldModel**
- **MapModel** * **mapModel**
- **MainViewMap** * **mainViewMap**
The MainView Map that represents the world, the game takes place in.
- **EditorTools** * **editorTools**
The tools dialog for the editor.
- **EditorMenu** * **editorMenu**
The menu dialog, having all sorts of functionallity to change editor and game related things.
- **WorldView** * **worldView**
The side pane, showing the world structure and maps.
- **GameModel** * **gameModel**
- QDockWidget * **editorToolsDock**
- QDockWidget * **worldViewDock**
- **Processor** * **processor**
- bool **done**

8.61.1 Detailed Description

Holds all the models and GUIs and initializes them.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.61.2 Constructor & Destructor Documentation

8.61.2.1 `MainWindow::MainWindow (QWidget * parent = 0) [explicit]`

8.61.3 Member Function Documentation

8.61.3.1 `void MainWindow::closeEvent (QCloseEvent * event) [protected],[virtual]`

8.61.3.2 `void MainWindow::createMainViewMap () [private],[slot]`

Creates a new **MainViewMap** (p. 164).

8.61.3.3 `void MainWindow::init () [private]`

Initialize the main window completely.

Returns

If the initialization was a success.

8.61.3.4 `bool MainWindow::initializeEditor () [private]`

Initialize the editor correctly. Make user choose whether to load an existing editor or create a new one.

Returns

If the initialization was a success.

8.61.3.5 `bool MainWindow::initializeGame () [private]`

Initialize Game, so the current map can be loaded. Make the user choose whether to start a new game, load an existing game (i.e.: E-Mail Game), set some options (if any) or exit the game.

Returns

If the initialization was a success.

8.61.3.6 `void MainWindow::setNewMap (MapModel * mapModel) [private],[slot]`

Sets a new map.

Parameters

<i>mapModel</i>	The given map to be set.
-----------------	--------------------------

8.61.3.7 void MainWindow::updateEditorSize () [private],[slot]

Updates the size of the main window depending on the mapsize and the dockwidgets.

8.61.3.8 void MainWindow::updateSize () [private],[slot]

Updates the size of the main window depending on the mapsize.

8.61.3.9 void MainWindow::updateWindowTitle () [private],[slot]

Updates the title of the window to whatever the current mapfilename is.

8.61.3.10 void MainWindow::windowSizeChanged () [signal]

8.61.4 Member Data Documentation

8.61.4.1 bool MainWindow::done [private]

8.61.4.2 EditorMenu* MainWindow::editorMenu [private]

The menu dialog, having all sorts of functionality to change editor and game related things.

8.61.4.3 EditorModel* MainWindow::editorModel [private]

8.61.4.4 EditorTools* MainWindow::editorTools [private]

The tools dialog for the editor.

8.61.4.5 QDockWidget* MainWindow::editorToolsDock [private]

8.61.4.6 GameModel* MainWindow::gameModel [private]

8.61.4.7 MainViewMap* MainWindow::mainViewMap [private]

The MainView Map that represents the world, the game takes place in.

8.61.4.8 MapModel* MainWindow::mapModel [private]

8.61.4.9 Processor* MainWindow::processor [private]

8.61.4.10 WorldModel* MainWindow::worldModel [private]

8.61.4.11 WorldView* MainWindow::worldView [private]

The side pane, showing the world structure and maps.

8.61.4.12 QDockWidget* MainWindow::worldViewDock [private]

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

- /home/ryoga/git/pidgirl/pidgirl-engine/**MainWindow.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/**MainWindow.cpp**

8.62 MapInfos Class Reference

Information about maps.

```
#include <MapPropertiesDialog.h>
```

8.62.1 Detailed Description

Information about maps.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

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

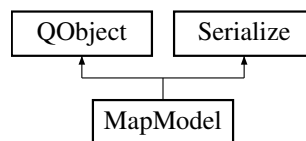
- /home/ryoga/git/pidgirl/pidgirl-engine/Editor/**MapPropertiesDialog.h**

8.63 MapModel Class Reference

Represents the map.

```
#include <MapModel.h>
```

Inheritance diagram for MapModel:



Public Types

- enum **Side** { **Left** =0, **Right** }

Public Slots

- void **addStartPositionStack** (**StackModel** *startPositionStack)
- void **removeStartPositionStack** (**StackModel** *startPositionStack)
- void **addPlayerStack** (**StackModel** *playerStacks)
- void **removePlayerStack** (**StackModel** *playerStack)

Signals

- void **gridChanged** ()
- void **horizontalSizeChanged** ()
- void **verticalSizeChanged** ()
- void **heightSizeChanged** ()
- void **squareSizeChanged** ()
- void **gridLineSizeChanged** ()
- void **clipboardObjectSet** ()
- void **mapPathChanged** ()

Public Member Functions

- **MapModel** (QObject *parent=0)
- **MapModel** (int hSquareCount, int vSquareCount, int maxHeight, int **gridLineSize**, QObject *parent=0)
- void **initMapModel** (int hSquareCount, int vSquareCount, int maxHeight, int **gridLineSize**)
Initialize the map model.
- void **resizeStackModelGrid** (int newHSquareCount, **MapModel::Side** hDirection, int newVSquareCount, **MapModel::Side** vDirection, int newMaxHeight, **MapModel::Side** heightDirection)
Resize the stack model grid.
- void **clearStackModelGrid** ()
Clear everything from the stack model grid.
- void **setNewStandardMap** ()
Clears the map model and loads the standard map.
- void **correctPositionEntries** ()
Correct the position entries of every single stack model to represent their actual position.
- void **clearPath** ()
Clear the current path.
- QString **getLocalMapPath** () const
- int **getHSquareCount** () const
- int **getVSquareCount** () const
- int **getSquareSizeZoomFactor** () const
- int **getSquareSize** () const
- int **getGridLineSize** () const
- int **getMaxHeight** () const
- int **getCurrentHeight** () const
- QSize **getMapSize** () const
- QVector< **StackModel** * > **getCurrentPath** () const
- bool **hasStartPositions** () const
- void **setLocalMapPath** (const QString &localMapPath)
- void **setSquareSizeZoomFactor** (int zoomFactor)
- void **updateSquareSize** ()
- void **setGridLineSize** (int gridLineSize, bool willEmit=true)
- void **setCurrentHeight** (int currentHeight)
- void **setClipboardObject** (ObjectBase *object)
- void **setConvertedPath** (QVector< **AStarMapEntry** > path)
- **StackModel** * **refStartPositionStack** () const
- **StackModel** * **refPlayerStack** (unsigned int playerId) const
- **StackModel** * **refDestinationStackModel** () const
- **StackModel** * **refStackModel** (int x, int y) const
- **StackModel** * **refStackModel** (int x, int y, int z) const
- **StackModel** * **refStackModel** (const QPoint &pos) const
- **ObjectBase** * **refClipboardObject** () const
- virtual void **serialize** (QIODevice *device) const
- void **serialize** (QDataStream &dataStream) const
- virtual void **deserialize** (QIODevice *device)
- void **deserialize** (QDataStream &dataStream)

Private Member Functions

- void **firstTimeInit** ()
First and only initialization of the map model for the first time.
- void **createStackModelAt** (int x, int y, int z)
*Create a new **StackModel** (p. 335) at a given position.*
- void **removeStackModelAt** (int x, int y, int z)
*Remove a **StackModel** (p. 335) at a given position.*
- void **deleteStackModelAt** (int x, int y, int z)
*Delete a **StackModel** (p. 335) at a given position.*
- void **removeHorizontalLeft** (int amount)
Remove elements of the vector so the horizontal square count changes while still maintaining the other vectors from left.
- void **removeHorizontalRight** (int amount)
Remove elements of the vector so the horizontal square count changes while still maintaining the other vectors from right.
- void **appendHorizontalLeft** (int amount)
Append elements of the vector so the horizontal square count changes while still maintaining the other vectors from left.
- void **appendHorizontalRight** (int amount)
Append elements of the vector so the horizontal square count changes while still maintaining the other vectors from right.
- void **setHSquareCount** (int hSquareCount, **MapModel::Side** from=**MapModel::Right**)
Resize the vector so the horizontal square count changes while still maintaining the other vectors.
- void **removeVerticalLeft** (int amount)
Remove elements of the vector so the vertical square count changes while still maintaining the other vectors from left.
- void **removeVerticalRight** (int amount)
Remove elements of the vector so the vertical square count changes while still maintaining the other vectors from right.
- void **appendVerticalLeft** (int amount)
Append elements of the vector so the vertical square count changes while still maintaining the other vectors from left.
- void **appendVerticalRight** (int amount)
Append elements of the vector so the vertical square count changes while still maintaining the other vectors from right.
- void **setVSquareCount** (int vSquareCount, **MapModel::Side** from=**MapModel::Right**)
Resize the vector so the vertical square count changes while still maintaining the other vectors.
- void **removeHeightLeft** (int amount)
Remove elements of the vector so the height square count changes while still maintaining the other vectors.
- void **removeHeightRight** (int amount)
Remove elements of the vector so the height square count changes while still maintaining the other vectors.
- void **appendHeightLeft** (int amount)
Append elements of the vector so the height square count changes while still maintaining the other vectors.
- void **appendHeightRight** (int amount)
Append elements of the vector so the height square count changes while still maintaining the other vectors.
- void **setMaxHeight** (int maxHeight, **MapModel::Side** from=**MapModel::Right**)
Resize the vector so the height square count changes while still maintaining the other vectors.

Private Attributes

- QVector< QVector< QVector< **StackModel** * > > > **stackModelGrid**
The actual representation of the mainView map as a grid.
- QString **localMapPath**
Is the unique ID and path for this map.
- int **squareSize**
The size of a single square on the mainView grid.
- int **gridLineSize**
- int **squareSizeZoomFactor**
The size of the grid lines. It works that 0 means there will be only a line with a width of 1px. Therefore 1 means that the width will be increased by one on both sides of the line, resulting in a line with a width of 3px. 2 would mean 5px and so on.
- int **currentHeight**
- QVector< **StackModel** * > **startPositionStacks**
The stack of start positions.
- QVector< **StackModel** * > **playerStacks**
The stacks of the player characters.
- QVector< **StackModel** * > **currentPath**
The path that, if confirmed, will be taken by the current player.
- **ObjectBase** * **clipboardObject**
The object that has been copied and will be pasted if issued.

8.63.1 Detailed Description

Represents the map.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.63.2 Member Enumeration Documentation

8.63.2.1 enum MapModel::Side

Enumerator

Left

Right

8.63.3 Constructor & Destructor Documentation

8.63.3.1 MapModel::MapModel (QObject * *parent* = 0) [explicit]

8.63.3.2 MapModel::MapModel (int *hSquareCount*, int *vSquareCount*, int *maxHeight*, int *gridLineSize*, QObject * *parent* = 0) [explicit]

8.63.4 Member Function Documentation

8.63.4.1 void MapModel::addPlayerStack (StackModel * *playerStacks*) [slot]

8.63.4.2 void MapModel::addStartPositionStack (StackModel * *startPositionStack*) [slot]

8.63.4.3 void MapModel::appendHeightLeft (int *amount*) [private]

Append elements of the vector so the height square count changes while still maintaining the other vectors.

Parameters

<i>amount</i>	The count to which the height squares should be appended.
<i>from</i>	The side from which to append from.

8.63.4.4 void MapModel::appendHeightRight (int *amount*) [private]

Append elements of the vector so the height square count changes while still maintaining the other vectors.

Parameters

<i>amount</i>	The count to which the height squares should be appended.
<i>from</i>	The side from which to append from.

8.63.4.5 void MapModel::appendHorizontalLeft (int *amount*) [private]

Append elements of the vector so the horizontal square count changes while still maintaining the other vectors from left.

Parameters

<i>amount</i>	The count to which the horizontal squares should be appended.
<i>from</i>	The side from which to append from.

8.63.4.6 void MapModel::appendHorizontalRight (int *amount*) [private]

Append elements of the vector so the horizontal square count changes while still maintaining the other vectors from right.

Parameters

<i>amount</i>	The count to which the horizontal squares should be appended.
<i>from</i>	The side from which to append from.

8.63.4.7 void MapModel::appendVerticalLeft (int *amount*) [private]

Append elements of the vector so the vertical square count changes while still maintaining the other vectors from left.

Parameters

<i>amount</i>	The count to which the vertical squares should be appended.
---------------	---

<i>from</i>	The side from which to append from.
-------------	-------------------------------------

8.63.4.8 void MapModel::appendVerticalRight (int *amount*) [private]

Append elements of the vector so the vertical square count changes while still maintaining the other vectors from right.

Parameters

<i>amount</i>	The count to which the vertical squares should be appended.
<i>from</i>	The side from which to append from.

8.63.4.9 void MapModel::clearPath ()

Clear the current path.

8.63.4.10 void MapModel::clearStackModelGrid ()

Clear everything from the stack model grid.

8.63.4.11 void MapModel::clipboardObjectSet () [signal]

8.63.4.12 void MapModel::correctPositionEntries ()

Correct the position entries of every single stack model to represent their actual position.

8.63.4.13 void MapModel::createStackModelAt (int *x*, int *y*, int *z*) [private]

Create a new **StackModel** (p. 335) at a given position.

Parameters

<i>x</i>	The x position of the stack.
<i>y</i>	The y position of the stack.
<i>z</i>	The z position of the stack.

8.63.4.14 void MapModel::deleteStackModelAt (int *x*, int *y*, int *z*) [private]

Delete a **StackModel** (p. 335) at a given position.

Parameters

<i>x</i>	The x position of the stack.
<i>y</i>	The y position of the stack.
<i>z</i>	The z position of the stack.

8.63.4.15 void MapModel::deserialize (QIODevice * *device*) [virtual]

Implements **Serialize** (p. 314).

8.63.4.16 void MapModel::deserialize (QDataStream & *dataStream*)

8.63.4.17 void MapModel::firstTimeInit () [private]

First and only initialization of the map model for the first time.

8.63.4.18 int MapModel::getCurrentHeight () const

8.63.4.19 QVector< StackModel * > MapModel::getCurrentPath () const

8.63.4.20 int MapModel::getGridLineSize () const

8.63.4.21 int MapModel::getHSquareCount () const

8.63.4.22 QString MapModel::getLocalMapPath () const

8.63.4.23 QSize MapModel::getMapSize () const

8.63.4.24 int MapModel::getMaxHeight () const

8.63.4.25 int MapModel::getSquareSize () const

8.63.4.26 int MapModel::getSquareSizeZoomFactor () const

8.63.4.27 int MapModel::getVSquareCount () const

8.63.4.28 void MapModel::gridChanged () [signal]

8.63.4.29 void MapModel::gridLineSizeChanged () [signal]

8.63.4.30 bool MapModel::hasStartPositions () const

8.63.4.31 void MapModel::heightSizeChanged () [signal]

8.63.4.32 void MapModel::horizontalSizeChanged () [signal]

8.63.4.33 void MapModel::initMapModel (int *hSquareCount*, int *vSquareCount*, int *maxHeight*, int *gridLineSize*)

Initialize the map model.

8.63.4.34 void MapModel::mapPathChanged () [signal]

8.63.4.35 ObjectBase * MapModel::refClipboardObject () const

8.63.4.36 StackModel * MapModel::refDestinationStackModel () const

8.63.4.37 StackModel * MapModel::refPlayerStack (unsigned int *playerID*) const

8.63.4.38 StackModel * MapModel::refStackModel (int *x*, int *y*) const

8.63.4.39 StackModel * MapModel::refStackModel (int *x*, int *y*, int *z*) const

8.63.4.40 StackModel * MapModel::refStackModel (const QPoint & *pos*) const

8.63.4.41 StackModel * MapModel::refStartPositionStack () const

8.63.4.42 `void MapModel::removeHeightLeft (int amount)` [private]

Remove elements of the vector so the height square count changes while still maintaining the other vectors.

Parameters

<i>amount</i>	The count to which the height squares should be removed.
<i>from</i>	The side from which to remove from.

8.63.4.43 void MapModel::removeHeightRight (int *amount*) [private]

Remove elements of the vector so the height square count changes while still maintaining the other vectors.

Parameters

<i>amount</i>	The count to which the height squares should be removed.
<i>from</i>	The side from which to remove from.

8.63.4.44 void MapModel::removeHorizontalLeft (int *amount*) [private]

Remove elements of the vector so the horizontal square count changes while still maintaining the other vectors from left.

Parameters

<i>amount</i>	The count to which the horizontal squares should be removed.
<i>from</i>	The side from which to remove from.

8.63.4.45 void MapModel::removeHorizontalRight (int *amount*) [private]

Remove elements of the vector so the horizontal square count changes while still maintaining the other vectors from right.

Parameters

<i>amount</i>	The count to which the horizontal squares should be removed.
<i>from</i>	The side from which to remove from.

8.63.4.46 void MapModel::removePlayerStack (StackModel * *playerStack*) [slot]8.63.4.47 void MapModel::removeStackModelAt (int *x*, int *y*, int *z*) [private]

Remove a **StackModel** (p. 335) at a given position.

Parameters

<i>x</i>	The x position of the stack.
<i>y</i>	The y position of the stack.
<i>z</i>	The z position of the stack.

8.63.4.48 void MapModel::removeStartPositionStack (StackModel * *startPositionStack*) [slot]8.63.4.49 void MapModel::removeVerticalLeft (int *amount*) [private]

Remove elements of the vector so the vertical square count changes while still maintaining the other vectors from left.

Parameters

<i>amount</i>	The count to which the vertical squares should be removed.
---------------	--

8.63.4.50 `void MapModel::removeVerticalRight (int amount) [private]`

Remove elements of the vector so the vertical square count changes while still maintaining the other vectors from right.

Parameters

<i>amount</i>	The count to which the vertical squares should be removed.
---------------	--

8.63.4.51 `void MapModel::resizeStackModelGrid (int newHSquareCount, MapModel::Side hDirection, int newVSquareCount, MapModel::Side vDirection, int newMaxHeight, MapModel::Side heightDirection)`

Resize the stack model grid.

Parameters

<i>hSquareCount</i>	The new horizontal square count.
<i>hDirection</i>	Vertical resize direction.
<i>vSquareCount</i>	The new vertical square count.
<i>vDirection</i>	Horizontal resize direction.
<i>maxHeight</i>	The new max height.
<i>heightDirection</i>	Height resize direction.

8.63.4.52 `void MapModel::serialize (QIODevice * device) const [virtual]`

Implements **Serialize** (p.314).

8.63.4.53 `void MapModel::serialize (QDataStream & dataStream) const`

8.63.4.54 `void MapModel::setClipboardObject (ObjectBase * object)`

8.63.4.55 `void MapModel::setConvertedPath (QVector< AStarMapEntry > path)`

8.63.4.56 `void MapModel::setCurrentHeight (int currentHeight)`

8.63.4.57 `void MapModel::setGridLineSize (int gridLineSize, bool willEmit = true)`

8.63.4.58 `void MapModel::setHSquareCount (int hSquareCount, MapModel::Side from = MapModel::Right) [private]`

Resize the vector so the horizontal square count changes while still maintaining the other vectors.

Parameters

<i>hSquareCount</i>	The count to which the horizontal squares should be appended or removed.
<i>from</i>	The side from which to append or remove from.

8.63.4.59 `void MapModel::setLocalMapPath (const QString & localMapPath)`

8.63.4.60 `void MapModel::setMaxHeight (int maxHeight, MapModel::Side from = MapModel::Right)` [private]

Resize the vector so the height square count changes while still maintaining the other vectors.

Parameters

<i>vSquareCount</i>	The count to which the height squares should be appended or removed.
<i>from</i>	The side from which to append or remove from.

8.63.4.61 `void MapModel::setNewStandardMap ()`

Clears the map model and loads the standard map.

8.63.4.62 `void MapModel::setSquareSizeZoomFactor (int zoomFactor)`

8.63.4.63 `void MapModel::setVSquareCount (int vSquareCount, MapModel::Side from = MapModel::Right)`
[private]

Resize the vector so the vertical square count changes while still maintaining the other vectors.

Parameters

<i>vSquareCount</i>	The count to which the vertical squares should be appended or removed.
<i>from</i>	The side from which to append or remove from.

8.63.4.64 `void MapModel::squareSizeChanged ()` [signal]

8.63.4.65 `void MapModel::updateSquareSize ()`

8.63.4.66 `void MapModel::verticalSizeChanged ()` [signal]

8.63.5 Member Data Documentation

8.63.5.1 `ObjectBase* MapModel::clipboardObject` [private]

The object that has been copied and will be pasted if issued.

8.63.5.2 `int MapModel::currentHeight` [private]

8.63.5.3 `QVector< StackModel *> MapModel::currentPath` [private]

The path that, if confirmed, will be taken by the current player.

8.63.5.4 `int MapModel::gridLineSize` [private]

8.63.5.5 `QString MapModel::localMapPath` [private]

Is the unique ID and path for this map.

8.63.5.6 `QVector< StackModel *> MapModel::playerStacks` [private]

The stacks of the player characters.

8.63.5.7 `int MapModel::squareSize` [private]

The size of a single square on the mainView grid.

8.63.5.8 `int MapModel::squareSizeZoomFactor` [private]

The size of the grid lines. It works that 0 means there will be only a line with a width of 1px. Therefore 1 means that the width will be increased by one on both sides of the line, resulting in a line with a width of 3px. 2 would mean 5px and so on.

8.63.5.9 `QVector<QVector<QVector<StackModel*>>> MapModel::stackModelGrid` [private]

The actual representation of the mainView map as a grid.

8.63.5.10 `QVector<StackModel*> MapModel::startPositionStacks` [private]

The stack of start positions.

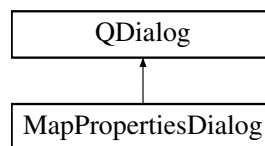
The documentation for this class was generated from the following files:

- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/**MapModel.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/**MapModel.cpp**

8.64 MapPropertiesDialog Class Reference

```
#include <MapPropertiesDialog.h>
```

Inheritance diagram for MapPropertiesDialog:



Public Member Functions

- **MapPropertiesDialog** (**Processor** *processor, QWidget *parent=0)
- void **updateHorizontal** (int horizontalSize)
- void **updateVertical** (int verticalSize)
- void **updateMaxHeight** (int maxHeight)

Private Slots

- void **applyChanges** ()

Private Attributes

- **Processor** * processor
- **OptionsLayout** * optionsLayout
- QWidget * geometryPropeties
- QGridLayout * geometryPropetiesLayout
- QSpinBox * horizontalSquareSpinBox
- QComboBox * horizontalSquareSideComboBox
- QSpinBox * verticalSquareSpinBox
- QComboBox * verticalSquareSideComboBox

- QSpinBox * **maxHeightSpinBox**
- QComboBox * **maxHeightComboBox**

8.64.1 Constructor & Destructor Documentation

8.64.1.1 MapPropertiesDialog::MapPropertiesDialog (Processor * *processor*, QWidget * *parent* = 0) [explicit]

8.64.2 Member Function Documentation

8.64.2.1 void MapPropertiesDialog::applyChanges () [private],[slot]

8.64.2.2 void MapPropertiesDialog::updateHorizontal (int *horizontalSize*)

8.64.2.3 void MapPropertiesDialog::updateMaxHeight (int *maxHeight*)

8.64.2.4 void MapPropertiesDialog::updateVertical (int *verticalSize*)

8.64.3 Member Data Documentation

8.64.3.1 QWidget* MapPropertiesDialog::geometryPropeties [private]

8.64.3.2 QGridLayout* MapPropertiesDialog::geometryPropetiesLayout [private]

8.64.3.3 QComboBox* MapPropertiesDialog::horizontalSquareSideComboBox [private]

8.64.3.4 QSpinBox* MapPropertiesDialog::horizontalSquareSpinBox [private]

8.64.3.5 QComboBox* MapPropertiesDialog::maxHeightComboBox [private]

8.64.3.6 QSpinBox* MapPropertiesDialog::maxHeightSpinBox [private]

8.64.3.7 OptionsLayout* MapPropertiesDialog::optionsLayout [private]

8.64.3.8 Processor* MapPropertiesDialog::processor [private]

8.64.3.9 QComboBox* MapPropertiesDialog::verticalSquareSideComboBox [private]

8.64.3.10 QSpinBox* MapPropertiesDialog::verticalSquareSpinBox [private]

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

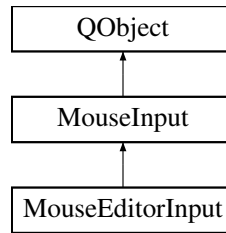
- /home/ryoga/git/pidgirl/pidgirl-engine/Editor/**MapPropertiesDialog.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Editor/**MapPropertiesDialog.cpp**

8.65 MouseEditorInput Class Reference

The mouse input manager for the editor.

```
#include <MouseEditorInput.h>
```

Inheritance diagram for MouseEditorInput:



Public Member Functions

- **MouseEditorInput** (**GameModel** *gameModel, **MapModel** *mapModel, **Processor** *processor, **QObject** *parent=0)
- virtual void **nextState** (**MouseState::MouseEventType** type, **QMouseEvent** *event, **StackModel** *target-Stack)
- void **showEditorContextMenu** (**StackModel** *contextMenuStack)

Private Slots

- void **checkRemoveAllAction** ()
Enables the remove all action, if there is anything on the stack.
- void **copyCharacter** ()
Create a copy of the character to maybe paste it later.
- void **cutCharacter** ()
Cut the character to maybe paste it later.
- void **removeCharacter** ()
Set the character to 0. Should it have been already set to 0, do nothing.
- void **editCharacter** ()
Edit the character.
- void **checkCharacterActions** ()
Change the state of the actions associated with the character.
- void **copyItems** ()
Create a copy of items to maybe paste it later.
- void **cutItems** ()
Cut items to maybe paste it later.
- void **removeItems** ()
Remove items from the stack.
- void **editItems** ()
Edit an item.
- void **checkItemsActions** ()
Change the state of the actions associated with the items.
- void **copySpecials** ()
Create a copy of specials to maybe paste it later.
- void **cutSpecials** ()
Cut specials to maybe paste them later.
- void **removeSpecials** ()
Remove specials from the stack.
- void **editSpecials** ()
Edit a special.
- void **checkSpecialsActions** ()
Change the state of the actions associated with the specials.

- void **copyWall** ()
Create a copy of the wall to maybe paste it later.
- void **cutWall** ()
Cut the wall to maybe paste it later.
- void **removeWall** ()
Set the wall to 0. Should it have been already set to 0, do nothing.
- void **editWall** ()
Edit the wall.
- void **checkWallActions** ()
Change the state of the actions associated with the wall.
- void **copyFloor** ()
Create a copy of the wall to maybe paste it later.
- void **cutFloor** ()
Cut the wall to maybe paste it later.
- void **removeFloor** ()
Set the wall to 0. Should it have been already set to 0, do nothing.
- void **editFloor** ()
Edit the wall.
- void **checkFloorActions** ()
Change the state of the actions associated with the floor.
- void **checkPasteAction** ()
Change the state of the paste action according what object was given.
- void **pasteObject** ()
Paste the content of the clipboard if it's not 0.
- void **removeAll** ()
Action represents the functionality to remove everything on a stack.

Private Attributes

- **StackModel** * **contextMenuStack**
- **QMenu** * **characterMenu**
- **QMenu** * **wallMenu**
- **QMenu** * **floorMenu**
- **QMenu** * **itemsMenu**
- **QMenu** * **specialsMenu**
- **QAction** * **copyCharacterAction**
- **QAction** * **cutCharacterAction**
- **QAction** * **removeCharacterAction**
- **QAction** * **editCharacterAction**
- **QAction** * **copyWallAction**
- **QAction** * **cutWallAction**
- **QAction** * **removeWallAction**
- **QAction** * **editWallAction**
- **QAction** * **copyFloorAction**
- **QAction** * **cutFloorAction**
- **QAction** * **removeFloorAction**
- **QAction** * **editFloorAction**
- **QAction** * **copyItemsAction**
- **QAction** * **cutItemsAction**
- **QAction** * **removeItemsAction**
- **QAction** * **editItemsAction**
- **QAction** * **copySpecialsAction**

- QAction * **cutSpecialsAction**
- QAction * **removeSpecialsAction**
- QAction * **editSpecialsAction**
- QAction * **pasteObjectAction**
- QAction * **removeAllAction**

Additional Inherited Members

8.65.1 Detailed Description

The mouse input manager for the editor.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.65.2 Constructor & Destructor Documentation

8.65.2.1 `MouseEditorInput::MouseEditorInput (GameModel * gameModel, MapModel * mapModel, Processor * processor, QObject * parent = 0)`

8.65.3 Member Function Documentation

8.65.3.1 `void MouseEditorInput::checkCharacterActions () [private],[slot]`

Change the state of the actions associated with the character.

8.65.3.2 `void MouseEditorInput::checkFloorActions () [private],[slot]`

Change the state of the actions associated with the floor.

8.65.3.3 `void MouseEditorInput::checkItemsActions () [private],[slot]`

Change the state of the actions associated with the items.

8.65.3.4 `void MouseEditorInput::checkPasteAction () [private],[slot]`

Change the state of the paste action according what object was given.

8.65.3.5 `void MouseEditorInput::checkRemoveAllAction () [private],[slot]`

Enables the remove all action, if there is anything on the stack.

8.65.3.6 `void MouseEditorInput::checkSpecialsActions () [private],[slot]`

Change the state of the actions associated with the specials.

8.65.3.7 `void MouseEditorInput::checkWallActions () [private],[slot]`

Change the state of the actions associated with the wall.

8.65.3.8 `void MouseEditorInput::copyCharacter () [private],[slot]`

Create a copy of the character to maybe paste it later.

8.65.3.9 `void MouseEditorInput::copyFloor () [private],[slot]`

Create a copy of the wall to maybe paste it later.

8.65.3.10 `void MouseEditorInput::copyItems () [private],[slot]`

Create a copy of items to maybe paste it later.

8.65.3.11 `void MouseEditorInput::copySpecials () [private],[slot]`

Create a copy of specials to maybe paste it later.

8.65.3.12 `void MouseEditorInput::copyWall () [private],[slot]`

Create a copy of the wall to maybe paste it later.

8.65.3.13 `void MouseEditorInput::cutCharacter () [private],[slot]`

Cut the character to maybe paste it later.

8.65.3.14 `void MouseEditorInput::cutFloor () [private],[slot]`

Cut the wall to maybe paste it later.

8.65.3.15 `void MouseEditorInput::cutItems () [private],[slot]`

Cut items to maybe paste it later.

8.65.3.16 `void MouseEditorInput::cutSpecials () [private],[slot]`

Cut specials to maybe paste them later.

8.65.3.17 `void MouseEditorInput::cutWall () [private],[slot]`

Cut the wall to maybe paste it later.

8.65.3.18 `void MouseEditorInput::editCharacter () [private],[slot]`

Edit the character.

8.65.3.19 void MouseEditorInput::editFloor () [private],[slot]

Edit the wall.

8.65.3.20 void MouseEditorInput::editItems () [private],[slot]

Edit an item.

8.65.3.21 void MouseEditorInput::editSpecials () [private],[slot]

Edit a special.

8.65.3.22 void MouseEditorInput::editWall () [private],[slot]

Edit the wall.

8.65.3.23 void MouseEditorInput::nextState (MouseState::MouseEventType type, QMouseEvent * event, StackModel * targetStack) [virtual]

Implements **MouseInput** (p. 195).

8.65.3.24 void MouseEditorInput::pasteObject () [private],[slot]

Paste the content of the clipboard if it's not 0.

8.65.3.25 void MouseEditorInput::removeAll () [private],[slot]

Action represents the functionality to remove everything on a stack.

8.65.3.26 void MouseEditorInput::removeCharacter () [private],[slot]

Set the character to 0. Should it have been already set to 0, do nothing.

8.65.3.27 void MouseEditorInput::removeFloor () [private],[slot]

Set the wall to 0. Should it have been already set to 0, do nothing.

8.65.3.28 void MouseEditorInput::removeItems () [private],[slot]

Remove items from the stack.

8.65.3.29 void MouseEditorInput::removeSpecials () [private],[slot]

Remove specials from the stack.

8.65.3.30 void MouseEditorInput::removeWall () [private],[slot]

Set the wall to 0. Should it have been already set to 0, do nothing.

8.65.3.31 void MouseEditorInput::showEditorContextMenu (StackModel * *contextMenuStack*)

8.65.4 Member Data Documentation

8.65.4.1 QMenu* MouseEditorInput::characterMenu [private]

8.65.4.2 StackModel* MouseEditorInput::contextMenuStack [private]

8.65.4.3 QAction* MouseEditorInput::copyCharacterAction [private]

8.65.4.4 QAction* MouseEditorInput::copyFloorAction [private]

8.65.4.5 QAction* MouseEditorInput::copyItemsAction [private]

8.65.4.6 QAction* MouseEditorInput::copySpecialsAction [private]

8.65.4.7 QAction* MouseEditorInput::copyWallAction [private]

8.65.4.8 QAction* MouseEditorInput::cutCharacterAction [private]

8.65.4.9 QAction* MouseEditorInput::cutFloorAction [private]

8.65.4.10 QAction* MouseEditorInput::cutItemsAction [private]

8.65.4.11 QAction* MouseEditorInput::cutSpecialsAction [private]

8.65.4.12 QAction* MouseEditorInput::cutWallAction [private]

8.65.4.13 QAction* MouseEditorInput::editCharacterAction [private]

8.65.4.14 QAction* MouseEditorInput::editFloorAction [private]

8.65.4.15 QAction* MouseEditorInput::editItemsAction [private]

8.65.4.16 QAction* MouseEditorInput::editSpecialsAction [private]

8.65.4.17 QAction* MouseEditorInput::editWallAction [private]

8.65.4.18 QMenu* MouseEditorInput::floorMenu [private]

8.65.4.19 QMenu* MouseEditorInput::itemsMenu [private]

8.65.4.20 QAction* MouseEditorInput::pasteObjectAction [private]

8.65.4.21 QAction* MouseEditorInput::removeAllAction [private]

8.65.4.22 QAction* MouseEditorInput::removeCharacterAction [private]

8.65.4.23 QAction* MouseEditorInput::removeFloorAction [private]

8.65.4.24 QAction* MouseEditorInput::removeItemsAction [private]

8.65.4.25 QAction* MouseEditorInput::removeSpecialsAction [private]

8.65.4.26 QAction* MouseEditorInput::removeWallAction [private]

8.65.4.27 `QMenu* MouseEditorInput::specialsMenu` [private]

8.65.4.28 `QMenu* MouseEditorInput::wallMenu` [private]

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

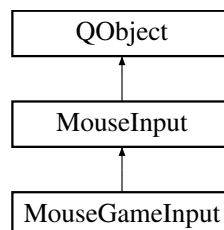
- `/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/MouseEditorInput.h`
- `/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/MouseEditorInput.cpp`

8.66 MouseGameInput Class Reference

The mouse input manager for the game.

```
#include <MouseGameInput.h>
```

Inheritance diagram for MouseGameInput:



Public Member Functions

- **MouseGameInput** (**GameModel** *gameModel, **MapModel** *mapModel, **Processor** *processor, **QObject** *parent=0)
- virtual void **nextState** (**MouseState::MouseEventType** type, **QMouseEvent** *event, **StackModel** *target-Stack)
- void **showGameContextMenu** (**StackModel** *stackModel)

Private Slots

- void **endTurn** ()
- void **showGameOptions** ()

Additional Inherited Members

8.66.1 Detailed Description

The mouse input manager for the game.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.66.2 Constructor & Destructor Documentation

8.66.2.1 **MouseGameInput::MouseGameInput** (**GameModel** * *gameModel*, **MapModel** * *mapModel*, **Processor** * *processor*, **QObject** * *parent* = 0)

8.66.3 Member Function Documentation

8.66.3.1 **void MouseGameInput::endTurn** () [private],[slot]

8.66.3.2 **void MouseGameInput::nextState** (**MouseState::MouseEventType** *type*, **QMouseEvent** * *event*, **StackModel** * *targetStack*) [virtual]

Implements **MouseInput** (p. 195).

8.66.3.3 **void MouseGameInput::showGameContextMenu** (**StackModel** * *stackModel*)

8.66.3.4 **void MouseGameInput::showGameOptions** () [private],[slot]

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

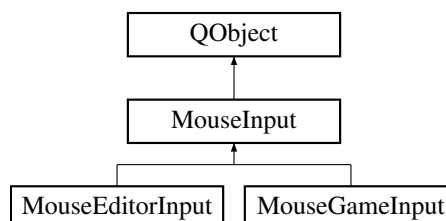
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/**MouseGameInput.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/**MouseGameInput.cpp**

8.67 MouseInput Class Reference

The base class that manages the flow of the events to be processed next.

```
#include <MouseInput.h>
```

Inheritance diagram for MouseInput:



Signals

- void **showContextMenu** (**StackModel** *)

Public Member Functions

- **MouseInput** (**MouseState** **mouseState*, **GameModel** **gameModel*, **MapModel** **mapModel*, **Processor** **processor*, **QObject** **parent*=0)
- **~MouseInput** ()
- virtual void **nextState** (**MouseState::MouseEventType** *type*, **QMouseEvent** **event*, **StackModel** **targetStack*)=0
- virtual void **mousePressed** (**QMouseEvent** **event*, **StackModel** **target*)
- virtual void **mouseReleased** (**QMouseEvent** **event*, **StackModel** **target*)
- void **showContextMenu** (const **QPoint** &*pos*)

Protected Slots

- void **deleteContextMenu** ()

Protected Attributes

- **MouseState** * **currentMouseState**
- **Processor** * **processor**
- **GameModel** * **gameModel**
- **MapModel** * **mapModel**
- QMenu * **contextMenu**

8.67.1 Detailed Description

The base class that manages the flow of the events to be processed next.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.67.2 Constructor & Destructor Documentation

8.67.2.1 **MouseInput::MouseInput** (**MouseState** * *mouseState*, **GameModel** * *gameModel*, **MapModel** * *mapModel*, **Processor** * *processor*, **QObject** * *parent* = 0) [explicit]

8.67.2.2 **MouseInput::~~MouseInput** ()

8.67.3 Member Function Documentation

8.67.3.1 void **MouseInput::deleteContextMenu** () [protected],[slot]

8.67.3.2 void **MouseInput::mousePressed** (**QMouseEvent** * *event*, **StackModel** * *target*) [virtual]

8.67.3.3 void **MouseInput::mouseReleased** (**QMouseEvent** * *event*, **StackModel** * *target*) [virtual]

8.67.3.4 virtual void **MouseInput::nextState** (**MouseState::MouseEventType** *type*, **QMouseEvent** * *event*, **StackModel** * *targetStack*) [pure virtual]

Implemented in **MouseEditorInput** (p. 191), and **MouseGameInput** (p. 194).

8.67.3.5 void **MouseInput::showContextMenu** (**StackModel** *) [signal]

8.67.3.6 void **MouseInput::showContextMenu** (const **QPoint** & *pos*)

8.67.4 Member Data Documentation

8.67.4.1 **QMenu*** **MouseInput::contextMenu** [protected]

8.67.4.2 **MouseState*** **MouseInput::currentMouseState** [protected]

8.67.4.3 **GameModel*** **MouseInput::gameModel** [protected]

8.67.4.4 **MapModel*** **MouseInput::mapModel** [protected]

8.67.4.5 **Processor*** **MouseInput::processor** [protected]

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

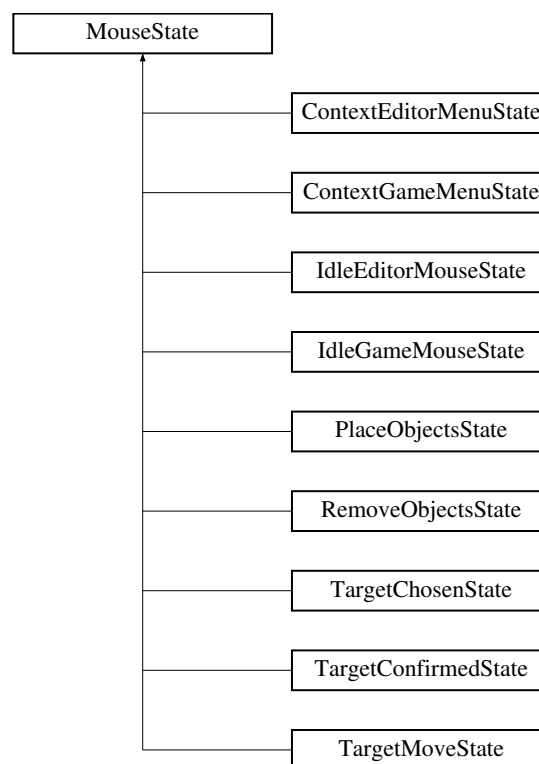
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/**MouseInput.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/**MouseInput.cpp**

8.68 MouseState Class Reference

The interface used for any state concerning mouse inputs.

```
#include <MouseState.h>
```

Inheritance diagram for MouseState:



Public Types

- enum **MouseEventType** { **PRESSED** = 0, **RELEASED** }

Public Member Functions

- **MouseState** (**Processor** *processor, **StackModel** *target=0)
- virtual ~**MouseState** ()

- virtual **MouseState** * **nextState** (**MouseState::MouseEventType** type, QMouseEvent *event, **StackModel** *targetStack, **StackModel** *fromStack=0)=0
Determines which state is next, depending on the input. This method might destroy this state and changes occur.
- virtual bool **showContextMenu** ()
- void **setTarget** (**StackModel** *targetStack)
- **StackModel** * **refTarget** ()

Protected Attributes

- **Processor** * **processor**
- **StackModel** * **target**

8.68.1 Detailed Description

The interface used for any state concerning mouse inputs.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.68.2 Member Enumeration Documentation

8.68.2.1 enum MouseState::MouseEventType

Enumerator

PRESSED

RELEASED

8.68.3 Constructor & Destructor Documentation

8.68.3.1 **MouseState::MouseState** (**Processor** * *processor*, **StackModel** * *target* = 0)

8.68.3.2 **MouseState::~~MouseState** () [virtual]

8.68.4 Member Function Documentation

8.68.4.1 virtual **MouseState*** **MouseState::nextState** (**MouseState::MouseEventType** type, QMouseEvent * event, **StackModel** * targetStack, **StackModel** * fromStack = 0) [pure virtual]

Determines which state is next, depending on the input. This method might destroy this state and changes occur.

Parameters

<i>event</i>	Holds the event that has been triggered.
<i>stack</i>	The stack where this method has been invoked from.

Returns

a pointer to the next state.

Implemented in **PlaceObjectsState** (p. 245), **RemoveObjectsState** (p. 278), **IdleGameMouseState** (p. 147), **ContextEditorMenuState** (p. 102), **ContextGameMenuState** (p. 103), **IdleEditorMouseState** (p. 146), **TargetChosenState** (p. 350), **TargetConfirmedState** (p. 352), and **TargetMoveState** (p. 353).

8.68.4.2 **StackModel * MouseState::refTarget ()**

8.68.4.3 **void MouseState::setTarget (StackModel * targetStack)**

8.68.4.4 **bool MouseState::showContextMenu ()** [virtual]

Reimplemented in **ContextEditorMenuState** (p. 102), and **ContextGameMenuState** (p. 103).

8.68.5 Member Data Documentation

8.68.5.1 **Processor* MouseState::processor** [protected]

8.68.5.2 **StackModel* MouseState::target** [protected]

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

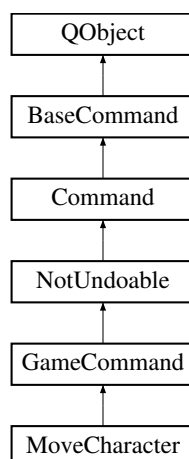
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/**MouseState.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/**MouseState.cpp**

8.69 MoveCharacter Class Reference

Move a character from stack to stack.

```
#include <MoveCharacter.h>
```

Inheritance diagram for MoveCharacter:



Public Member Functions

- **MoveCharacter** (**StackModel** ***from**, **StackModel** ***to**, **QObject** ***parent**=0)
- virtual bool **execute** ()
Executes the command.

Private Attributes

- **StackModel** * **from**
The stack from where to move the character.
- **StackModel** * **to**
The stack to where the character to.

Additional Inherited Members

8.69.1 Detailed Description

Move a character from stack to stack.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.69.2 Constructor & Destructor Documentation

8.69.2.1 **MoveCharacter::MoveCharacter** (**StackModel** * *from*, **StackModel** * *to*, **QObject** * *parent* = 0)

8.69.3 Member Function Documentation

8.69.3.1 bool **MoveCharacter::execute** () [virtual]

Executes the command.

Returns

True, if successful.

Implements **GameCommand** (p. 133).

8.69.4 Member Data Documentation

8.69.4.1 **StackModel*** **MoveCharacter::from** [private]

The stack from where to move the character.

8.69.4.2 StackModel* MoveCharacter::to [private]

The stack to where the character to.

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

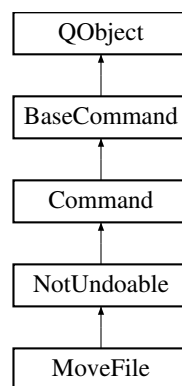
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Map/**MoveCharacter.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Map/**MoveCharacter.cpp**

8.70 MoveFile Class Reference

Move a file from an index to another one.

```
#include <MoveFile.h>
```

Inheritance diagram for MoveFile:



Public Member Functions

- **MoveFile** (const QModelIndex &**fromIndex**, const QModelIndex &**toIndex**, QObject *parent=0)
- virtual bool **execute** ()
Executes the command.

Private Attributes

- QModelIndex **fromIndex**
The index pointing to the file that will be moved.
- QModelIndex **toIndex**
The index to which the file will be moved to.

Additional Inherited Members

8.70.1 Detailed Description

Move a file from an index to another one.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.70.2 Constructor & Destructor Documentation

8.70.2.1 `MoveFile::MoveFile (const QModelIndex & fromIndex, const QModelIndex & toIndex, QObject * parent = 0)`

8.70.3 Member Function Documentation

8.70.3.1 `bool MoveFile::execute () [virtual]`

Executes the command.

Returns

True, if successful.

Implements **NotUndoable** (p. 215).

8.70.4 Member Data Documentation

8.70.4.1 `QModelIndex MoveFile::fromIndex [private]`

The index pointing to the file that will be moved.

8.70.4.2 `QModelIndex MoveFile::toIndex [private]`

The index to which the file will be moved to.

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

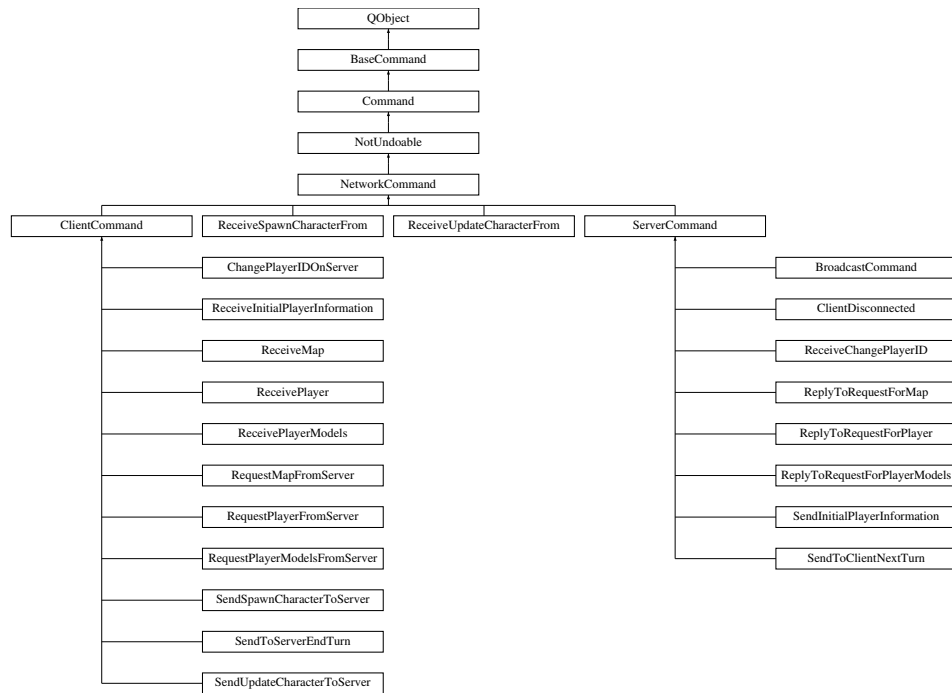
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/**MoveFile.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/**MoveFile.cpp**

8.71 NetworkCommand Class Reference

Command (p. 93) sent by Server or Client alike.

```
#include <NetworkCommand.h>
```

Inheritance diagram for NetworkCommand:



Public Member Functions

- **NetworkCommand** (QObject *parent=0)
- virtual bool **isExecutable** () const
Check if the command is executable.
- virtual bool **execute** ()=0
Executes the command.

Additional Inherited Members

8.71.1 Detailed Description

Command (p. 93) sent by Server or Client alike.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.71.2 Constructor & Destructor Documentation

8.71.2.1 **NetworkCommand::NetworkCommand** (QObject * *parent* = 0)

8.71.3 Member Function Documentation

8.71.3.1 virtual bool **NetworkCommand::execute** () [pure virtual]

Executes the command.

Returns

True, if successful.

Implements **NotUndoable** (p. 215).

Implemented in **BroadcastCommand** (p. 69), **BroadcastUpdateCharacter** (p. 73), **SendUpdateCharacterToServer** (p. 313), **ServerCommand** (p. 316), **ChangePlayerIDOnServer** (p. 74), **RequestMapFromServer** (p. 288), **RequestPlayerFromServer** (p. 290), **SendSpawnCharacterToServer** (p. 308), **ReceiveSpawnCharacterFrom** (p. 270), **BroadcastSpawnCharacter** (p. 71), **ClientCommand** (p. 80), **ReceiveUpdateCharacterFrom** (p. 271), **ReceiveInitialPlayerInformation** (p. 264), **ReceiveMap** (p. 266), **ReceivePlayer** (p. 267), **ReceivePlayerModels** (p. 268), **RequestPlayerModelsFromServer** (p. 291), **SendToServerEndTurn** (p. 311), **ClientDisconnected** (p. 81), **ReceiveChangePlayerID** (p. 263), **ReplyToRequestForMap** (p. 284), **ReplyToRequestForPlayer** (p. 286), **ReplyToRequestForPlayerModels** (p. 287), **SendInitialPlayerInformation** (p. 307), and **SendToClientNextTurn** (p. 310).

8.71.3.2 bool NetworkCommand::isExecutable () const [virtual]

Check if the command is executable.

Returns

If the command is executable.

Reimplemented from **NotUndoable** (p. 216).

Reimplemented in **ServerCommand** (p. 316), and **ClientCommand** (p. 80).

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

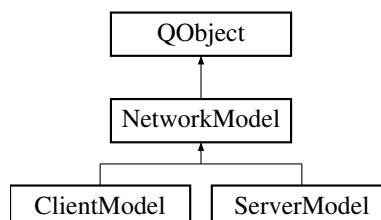
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/**NetworkCommand.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/**NetworkCommand.cpp**

8.72 NetworkModel Class Reference

The network model managing connections and interpretations of incoming messages to be converted to commands.

```
#include <NetworkModel.h>
```

Inheritance diagram for NetworkModel:

**Signals**

- void **connectionEstablished** ()
- void **clientDisconnect** ()
- void **allDataReceivedFromServer** (FromServer)
- void **allDataReceivedFromClient** (FromClient)

Public Member Functions

- **NetworkModel** (int **port**, QObject *parent=0)
- virtual ~**NetworkModel** ()
- virtual bool **changeClientID** (unsigned int fromID, unsigned int toID)=0
Change a client ID.
- virtual void **sendData** (unsigned int data, unsigned int playerId=0)=0
Sends the given data to a playerId.
- virtual void **sendData** (int data, unsigned int playerId=0)=0
- virtual void **sendData** (bool data, unsigned int playerId=0)=0
- virtual void **sendData** (const **Serialize** &data, unsigned int playerId=0)=0
- virtual void **sendData** (const QString &data, unsigned int playerId=0)=0
- virtual void **receiveData** (unsigned int &data, unsigned int playerId=0)=0
Receive data from a client with the given playerId.
- virtual void **receiveData** (int &data, unsigned int playerId=0)=0
- virtual void **receiveData** (bool &data, unsigned int playerId=0)=0
- virtual void **receiveData** (**Serialize** &data, unsigned int playerId=0)=0
- virtual void **receiveData** (QString &data, unsigned int playerId=0)=0
- virtual bool **isClient** ()=0
- virtual bool **isServer** ()=0
- virtual QVector< QString > **getConnectionList** ()=0
- virtual QVector< unsigned int > **getConnectedPlayerIDs** ()=0

Protected Attributes

- int **port**

8.72.1 Detailed Description

The network model managing connections and interpretations of incoming messages to be converted to commands.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.72.2 Constructor & Destructor Documentation

8.72.2.1 **NetworkModel::NetworkModel** (int *port*, QObject * *parent* = 0)

8.72.2.2 **NetworkModel::~~NetworkModel** () [virtual]

8.72.3 Member Function Documentation

8.72.3.1 void **NetworkModel::allDataReceivedFromClient** (**FromClient**) [signal]

8.72.3.2 void **NetworkModel::allDataReceivedFromServer** (**FromServer**) [signal]

8.72.3.3 virtual bool **NetworkModel::changeClientID** (unsigned int *fromID*, unsigned int *toID*) [pure virtual]

Change a client ID.

Parameters

<i>fromID</i>	The client connection ID from which to change.
<i>toID</i>	The client ID to which to change to.

Returns

True, if successfully changed. Otherwise, false.

Implemented in **ServerModel** (p. 321), and **ClientModel** (p. 89).

8.72.3.4 `void NetworkModel::clientDisconnect () [signal]`

8.72.3.5 `void NetworkModel::connectionEstablished () [signal]`

8.72.3.6 `virtual QVector< unsigned int > NetworkModel::getConnectedPlayerIDs () [pure virtual]`

Implemented in **ClientModel** (p. 89), and **ServerModel** (p. 322).

8.72.3.7 `virtual QVector< QString > NetworkModel::getConnectionList () [pure virtual]`

Implemented in **ClientModel** (p. 90), and **ServerModel** (p. 322).

8.72.3.8 `virtual bool NetworkModel::isClient () [pure virtual]`

Implemented in **ServerModel** (p. 322), and **ClientModel** (p. 90).

8.72.3.9 `virtual bool NetworkModel::isServer () [pure virtual]`

Implemented in **ServerModel** (p. 322), and **ClientModel** (p. 90).

8.72.3.10 `virtual void NetworkModel::receiveData (unsigned int & data, unsigned int playerId = 0) [pure virtual]`

Receive data from a client with the given playerId.

Parameters

<i>data</i>	The data to hold the data received.
<i>playerID</i>	The ID of the client data is received from.

Implemented in **ServerModel** (p. 322), and **ClientModel** (p. 90).

8.72.3.11 `virtual void NetworkModel::receiveData (int & data, unsigned int playerId = 0) [pure virtual]`

Implemented in **ServerModel** (p. 322), and **ClientModel** (p. 90).

8.72.3.12 `virtual void NetworkModel::receiveData (bool & data, unsigned int playerId = 0) [pure virtual]`

Implemented in **ServerModel** (p. 322), and **ClientModel** (p. 90).

8.72.3.13 `virtual void NetworkModel::receiveData (Serialize & data, unsigned int playerId = 0) [pure virtual]`

Implemented in **ServerModel** (p. 323), and **ClientModel** (p. 90).

8.72.3.14 `virtual void NetworkModel::receiveData (QString & data, unsigned int playerId = 0)` [pure virtual]

Implemented in **ServerModel** (p. 323), and **ClientModel** (p. 90).

8.72.3.15 `virtual void NetworkModel::sendData (unsigned int data, unsigned int playerId = 0)` [pure virtual]

Sends the given data to a playerId.

Parameters

<i>data</i>	The data to be sent.
<i>playerID</i>	The ID the data needs to be sent to.

Implemented in **ServerModel** (p. 323), and **ClientModel** (p. 90).

8.72.3.16 `virtual void NetworkModel::sendData (int data, unsigned int playerId = 0)` [pure virtual]

Implemented in **ServerModel** (p. 323), and **ClientModel** (p. 91).

8.72.3.17 `virtual void NetworkModel::sendData (bool data, unsigned int playerId = 0)` [pure virtual]

Implemented in **ServerModel** (p. 323), and **ClientModel** (p. 91).

8.72.3.18 `virtual void NetworkModel::sendData (const Serialize & data, unsigned int playerId = 0)` [pure virtual]

Implemented in **ServerModel** (p. 323), and **ClientModel** (p. 91).

8.72.3.19 `virtual void NetworkModel::sendData (const QString & data, unsigned int playerId = 0)` [pure virtual]

Implemented in **ServerModel** (p. 323), and **ClientModel** (p. 91).

8.72.4 Member Data Documentation

8.72.4.1 `int NetworkModel::port` [protected]

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

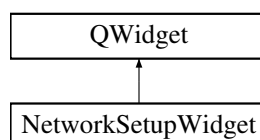
- /home/ryoga/git/pidgirl/pidgirl-engine/Network/**NetworkModel.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Network/**NetworkModel.cpp**

8.73 NetworkSetupWidget Class Reference

The widget to set up an Network Game.

```
#include <NetworkSetupWidget.h>
```

Inheritance diagram for NetworkSetupWidget:



Public Member Functions

- **NetworkSetupWidget** (QWidget *parent=0)

Public Attributes

- QGridLayout * **networkWidgetLayout**
- QLabel * **portLabel**
- QLineEdit * **portEdit**
- QIntValidator * **portValidator**
- QPushButton * **gameStartButton**

8.73.1 Detailed Description

The widget to set up an Network Game.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.73.2 Constructor & Destructor Documentation

8.73.2.1 **NetworkSetupWidget::NetworkSetupWidget** (QWidget * *parent* = 0) [explicit]

8.73.3 Member Data Documentation

8.73.3.1 **QPushButton*** NetworkSetupWidget::gameStartButton

8.73.3.2 **QGridLayout*** NetworkSetupWidget::networkWidgetLayout

8.73.3.3 **QLineEdit*** NetworkSetupWidget::portEdit

8.73.3.4 **QLabel*** NetworkSetupWidget::portLabel

8.73.3.5 **QIntValidator*** NetworkSetupWidget::portValidator

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

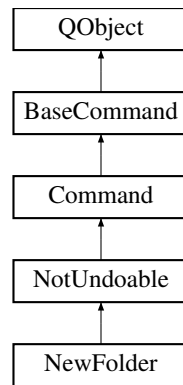
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/**NetworkSetupWidget.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/**NetworkSetupWidget.cpp**

8.74 NewFolder Class Reference

Create a new folder in the world.

```
#include <NewFolder.h>
```

Inheritance diagram for NewFolder:



Public Member Functions

- **NewFolder** (const QModelIndex &**index**, QObject *parent=0)
- virtual bool **execute** ()
Executes the command.

Private Attributes

- QModelIndex **index**
Create a new folder in the place this index is pointing to.

Additional Inherited Members

8.74.1 Detailed Description

Create a new folder in the world.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.74.2 Constructor & Destructor Documentation

8.74.2.1 `NewFolder::NewFolder (const QModelIndex & index, QObject * parent = 0)`

8.74.3 Member Function Documentation

8.74.3.1 `bool NewFolder::execute () [virtual]`

Executes the command.

Returns

True, if successful.

Implements **NotUndoable** (p. 215).

8.74.4 Member Data Documentation

8.74.4.1 QModelIndex NewFolder::index [private]

Create a new folder in the place this index is pointing to.

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

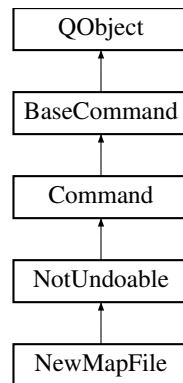
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/**NewFolder.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/**NewFolder.cpp**

8.75 NewMapFile Class Reference

Create a new map file.

```
#include <NewMapFile.h>
```

Inheritance diagram for NewMapFile:



Public Member Functions

- **NewMapFile** (const QModelIndex &**index**, QObject *parent=0)
- virtual bool **execute** ()
Executes the command.

Private Attributes

- QModelIndex **index**
The index where to create a new map.
- FileManager **fileManager**
The file manager to create the map with.
- QDir **dir**
The directory in which to create the new map in.
- QString **file**
The file name which to create.

Additional Inherited Members

8.75.1 Detailed Description

Create a new map file.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.75.2 Constructor & Destructor Documentation

8.75.2.1 `NewMapFile::NewMapFile (const QModelIndex & index, QObject * parent = 0)`

8.75.3 Member Function Documentation

8.75.3.1 `bool NewMapFile::execute ()` [virtual]

Executes the command.

Returns

True, if successful.

Implements **NotUndoable** (p. 215).

8.75.4 Member Data Documentation

8.75.4.1 `QDir NewMapFile::dir` [private]

The directory in which to create the new map in.

8.75.4.2 `QString NewMapFile::file` [private]

The file name which to create.

8.75.4.3 `FileManager NewMapFile::fileManager` [private]

The file manager to create the map with.

8.75.4.4 `QModelIndex NewMapFile::index` [private]

The index where to create a new map.

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

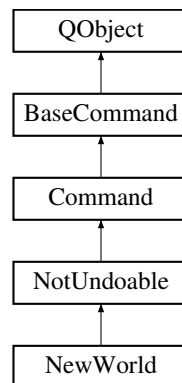
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/**NewMapFile.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/**NewMapFile.cpp**

8.76 NewWorld Class Reference

Choose the name of a file and create a new world at that location.

```
#include <NewWorld.h>
```

Inheritance diagram for NewWorld:



Public Member Functions

- **NewWorld** (QObject *parent=0)
- bool **execute** ()

Executes the command.

Private Attributes

- **FileManager** fileManager

The file manager to create the world with.

Additional Inherited Members

8.76.1 Detailed Description

Choose the name of a file and create a new world at that location.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.76.2 Constructor & Destructor Documentation

8.76.2.1 **NewWorld::NewWorld** (QObject * parent = 0)

8.76.3 Member Function Documentation

8.76.3.1 bool **NewWorld::execute** () [virtual]

Executes the command.

Returns

True, if successful.

Implements **NotUndoable** (p. 215).

8.76.4 Member Data Documentation**8.76.4.1 FileManager NewWorld::fileManager** [private]

The file manager to create the world with.

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

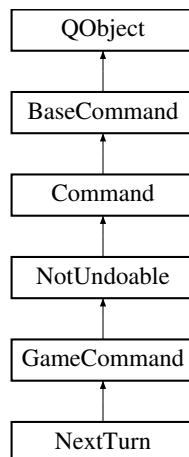
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/**NewWorld.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/**NewWorld.cpp**

8.77 NextTurn Class Reference

Begin the next turn.

```
#include <NextTurn.h>
```

Inheritance diagram for NextTurn:

**Public Member Functions**

- **NextTurn** (QObject *parent=0)
- virtual bool **execute** ()

Executes the command.

Additional Inherited Members**8.77.1 Detailed Description**

Begin the next turn.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.77.2 Constructor & Destructor Documentation

8.77.2.1 `NextTurn::NextTurn (QObject * parent = 0)`

8.77.3 Member Function Documentation

8.77.3.1 `bool NextTurn::execute () [virtual]`

Executes the command.

Returns

True, if successful.

Implements **GameCommand** (p. 133).

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

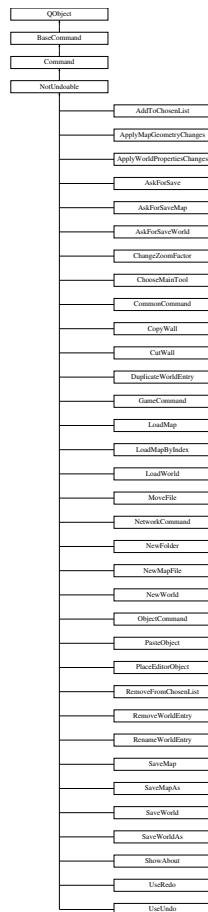
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/**NextTurn.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/**NextTurn.cpp**

8.78 NotUndoable Class Reference

Abstract class to be inherited by Commands which are not undoable.

```
#include <NotUndoable.h>
```

Inheritance diagram for NotUndoable:



Public Member Functions

- **NotUndoable** (bool **clearsUndoRedo**, QObject *parent=0)
- virtual bool **isExecutable** () const
Check if the command is executable.
- virtual bool **execute** ()=0
Executes the command.
- virtual void **undo** ()
Undo changes made by the command.
- virtual void **redo** ()
Redo changes made by the undoing of the command.
- virtual bool **isUndoable** ()
Determines undoability.
- virtual bool **isClearingUndoRedo** ()
Determines, if undo and redo should be cleared.

Private Attributes

- const bool **clearsUndoRedo**
True, if this command clears everything in the undo and redo stack.

Additional Inherited Members

8.78.1 Detailed Description

Abstract class to be inherited by Commands which are not undoable.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.78.2 Constructor & Destructor Documentation

8.78.2.1 `NotUndoable::NotUndoable (bool clearsUndoRedo, QObject * parent = 0)` `[explicit]`

8.78.3 Member Function Documentation

8.78.3.1 `virtual bool NotUndoable::execute ()` `[pure virtual]`

Executes the command.

Returns

True, if successful.

Implements **Command** (p. 94).

Implemented in **CreateCharacter** (p. 109), **SetupNewGame** (p. 325), **ShowGameOptions** (p. 330), **Connect-To** (p. 100), **LoadGame** (p. 150), **FirstTurn** (p. 130), **BroadcastCommand** (p. 69), **SaveEMailGame** (p. 297), **ApplyMapGeometryChanges** (p. 50), **CopyFolder** (p. 105), **AddToChosenList** (p. 48), **NewMapFile** (p. 210), **BroadcastUpdateCharacter** (p. 73), **RemoveFolder** (p. 275), **FindPath** (p. 129), **LoadMap** (p. 152), **LoadMapByIndex** (p. 153), **PlaceEditorObject** (p. 244), **LoadWorld** (p. 157), **MoveFile** (p. 201), **RemoveFromChosenList** (p. 277), **MoveCharacter** (p. 199), **ObjectCommand** (p. 227), **CopyWall** (p. 106), **DuplicateWorldEntry** (p. 112), **RemoveWorldEntry** (p. 281), **SpawnGameObject** (p. 333), **PreloadWorld** (p. 257), **ResumeEMailGame** (p. 293), **AskForSaveMap** (p. 54), **ApplyWorldPropertiesChanges** (p. 52), **AskForSaveWorld** (p. 56), **NewFolder** (p. 208), **NewWorld** (p. 211), **RenameWorldEntry** (p. 283), **SaveMap** (p. 302), **SaveMapAs** (p. 303), **SaveWorld** (p. 304), **SaveWorldAs** (p. 306), **PreloadAllMaps** (p. 255), **SaveGame** (p. 299), **SpawnPlayerOnCurrentMap** (p. 335), **CutWall** (p. 111), **PasteObject** (p. 242), **ChooseMainTool** (p. 77), **SendUpdateCharacterToServer** (p. 313), **ServerCommand** (p. 316), **UseRedo** (p. 359), **Quit** (p. 262), **ChangeZoomFactor** (p. 76), **LoadStartMapFromIndex** (p. 155), **ChangePlayerIDOnServer** (p. 74), **RequestMapFromServer** (p. 288), **RequestPlayerFromServer** (p. 290), **SendSpawnCharacterToServer** (p. 308), **ReceiveSpawnCharacterFrom** (p. 270), **BroadcastSpawnCharacter** (p. 71), **CommonCommand** (p. 98), **GameCommand** (p. 133), **ClientCommand** (p. 80), **NetworkCommand** (p. 202), **ReceiveUpdateCharacterFrom** (p. 271), **UseUndo** (p. 361), **AskForSave** (p. 53), **EndTurn** (p. 125), **NextTurn** (p. 213), **YourTurnInformation** (p. 374), **ClearPath** (p. 78), **ReceiveInitialPlayerInformation** (p. 264), **ReceiveMap** (p. 266), **ReceivePlayer** (p. 267), **ReceivePlayerModels** (p. 268), **RequestPlayerModelsFromServer** (p. 291), **SendToServerEndTurn** (p. 311), **ClientDisconnected** (p. 81), **ReceiveChangePlayerID** (p. 263), **ReplyToRequestForMap** (p. 284), **ReplyToRequestForPlayer** (p. 286), **ReplyToRequestForPlayerModels** (p. 287), **SendInitialPlayerInformation** (p. 307), **SendToClientNextTurn** (p. 310), **ShowAbout** (p. 327), and **ShowDescription** (p. 328).

8.78.3.2 `bool NotUndoable::isClearingUndoRedo ()` `[virtual]`

Determines, if undo and redo should be cleared.

Returns

True, if undo and redo should be cleared.

Implements **Command** (p. 95).

8.78.3.3 bool NotUndoable::isExecutable () const [virtual]

Check if the command is executable.

Returns

If the command is executable.

Implements **Command** (p. 95).

Reimplemented in **ObjectCommand** (p. 227), **ServerCommand** (p. 316), **CommonCommand** (p. 99), **GameCommand** (p. 133), **ClientCommand** (p. 80), and **NetworkCommand** (p. 203).

8.78.3.4 bool NotUndoable::isUndoable () [virtual]

Determines undoability.

Returns

True, if this command can be undone.

Implements **Command** (p. 95).

8.78.3.5 void NotUndoable::redo () [virtual]

Redo changes made by the undoing of the command.

Implements **Command** (p. 95).

8.78.3.6 void NotUndoable::undo () [virtual]

Undo changes made by the command.

Implements **Command** (p. 96).

8.78.4 Member Data Documentation**8.78.4.1 const bool NotUndoable::clearsUndoRedo [private]**

True, if this command clears everything in the undo and redo stack.

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

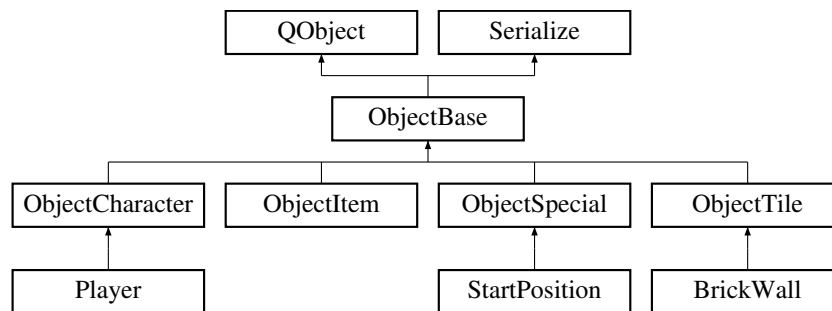
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/**NotUndoable.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/**NotUndoable.cpp**

8.79 ObjectBase Class Reference

The baseclass for the Objects to be placed into the game's mainView.


```
#include <ObjectBase.h>
```

Inheritance diagram for ObjectBase:



Public Types

- enum **ObjectType** { **ITEM**, **CHARACTER**, **SPECIAL**, **TILE** }

Specifies the type of the object via an enum value.

Public Member Functions

- **ObjectBase** ()
- **ObjectBase** (const **ObjectBase** &objectBase)
- **ObjectBase** (const QString &name, const QString &description, const QString &imagePath, bool **isVisible**=true, QObject *parent=0)
- **ObjectBase** (const QString &name="", const QString &description="", bool **isVisible**=true, QObject *parent=0)
- void **clearImageList** ()
Clear the list of images and set the usedImageIndex to -1.
- void **appendImage** (const QString &imagePath)
Load and append an image from a given path. The usedImageIndex will be set to 0, if it was -1 before.
- void **appendImages** (const QVector< QString > &imagePathList)
Load and append an image from a given path. The usedImageIndex will be set to 0, if it was -1 before.
- void **loadImage** (const QString &imagePath)
Loads an image from a path. Clears the vector of images and sets a new one at position. usedImageIndex to 0.
- void **loadImages** (const QVector< QString > &imagePathList)
Clears and then appends a new list of images. usedImageIndex will be set to 0.
- void **toggleVisibility** ()
Toggle visibility of an object.
- virtual **ObjectBase** * **copyBase** () const =0
Create a copy of the object.
- virtual bool **isSet** () const
Checks if position is (-1,-1,-1) or not. If it's a mix, something went wrong.
- void **unsetPosition** ()
Set the positions to (-1,-1,-1) to indicate the unset state.
- void **updateBase** (const **ObjectBase** &object)
Updates this object with another object.
- virtual void **update** (const **ObjectBase** &object)=0
- virtual QWidget * **createInformationWidget** (QWidget *parent) const =0
Create an information widget from an object to show informations about it.
- virtual **ObjectType** **getType** () const =0
- virtual int **getObjectIDNumber** () const =0

- QVector< QString > **getImagePathList** () const
- QString **getCurrentImagePath** () const
- int **getUsedImageIndex** () const
- QString **getObjectName** () const
- bool **getIsVisible** () const
- unsigned int **getSpawnID** () const
- int **getX** () const
- int **getY** () const
- int **getZ** () const
- virtual QMap< QString, **BaseCommand** * > **getCommandList** () const =0
- QString **getDescription** () const
- void **setUsedImageIndex** (int **usedImageIndex**)
- void **setIsVisible** (bool **isVisible**)
- void **setSpawnID** (unsigned int **spawnID**)
- void **setPosition** (int **x**, int **y**, int **z**)
- void **setDescription** (const QString &**description**)
- QImage * **refImage** ()
Reference the image given by the usedImageIndex.
- QImage * **refImage** (int **index**)
Reference the image given by the given index.
- virtual void **serialize** (QDataStream &**dataStream**) const
- virtual void **serialize** (QIODevice ***device**) const =0
- virtual void **deserialize** (QDataStream &**dataStream**)
- virtual void **deserialize** (QIODevice ***device**)=0

Protected Member Functions

- void **initObjectBase** (const QString &**name**="", const QString &**description**="", int **usedImageIndex**=-1, bool **isVisible**=true)
Initialize Attributes.
- void **initObjectBase** (const QVector< QString > &**imagePathList**, const QString &**name**="", const QString &**description**="", int **usedImageIndex**=-1, bool **isVisible**=true)

Protected Attributes

- int **x**
- int **y**
- int **z**
The position of the objects. (-1,-1,-1), if not set anywhere.
- unsigned int **spawnID**
Denotes the spawn of that object. UINT_MAX when initialized.
- QString **description**
Description of this object.
- QVector< **ObjectImage** > **imageList**
The list of images shown and their respective paths.
- int **usedImageIndex**
- bool **isVisible**
Represents the image to be shown in imageList. Can't be less than 0 or more than the imageList.size()-1. In case of -1, there is no image to be shown in the list.

8.79.1 Detailed Description

The baseclass for the Objects to be placed into the game's mainView.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.79.2 Member Enumeration Documentation

8.79.2.1 enum ObjectBase::ObjectType

Specifies the type of the object via an enum value.

Enumerator

ITEM

CHARACTER

SPECIAL

TILE

8.79.3 Constructor & Destructor Documentation

8.79.3.1 ObjectBase::ObjectBase () [explicit]

8.79.3.2 ObjectBase::ObjectBase (const ObjectBase & objectBase) [explicit]

8.79.3.3 ObjectBase::ObjectBase (const QString & name, const QString & description, const QString & imagePath, bool isVisible = true, QObject * parent = 0) [explicit]

8.79.3.4 ObjectBase::ObjectBase (const QString & name = "", const QString & description = "", bool isVisible = true, QObject * parent = 0) [explicit]

8.79.4 Member Function Documentation

8.79.4.1 void ObjectBase::appendImage (const QString & imagePath)

Load and append an image from a given path. The *usedImageIndex* will be set to 0, if it was -1 before.

Parameters

<i>imagePath</i>	The path to the image to be loaded.
------------------	-------------------------------------

8.79.4.2 void ObjectBase::appendImages (const QVector< QString > & imagePathList)

Load and append an image from a given path. The *usedImageIndex* will be set to 0, if it was -1 before.

Parameters

<i>imagePath</i>	The path to the image to be loaded.
------------------	-------------------------------------

8.79.4.3 void ObjectBase::clearImageList ()

Clear the list of images and set the *usedImageIndex* to -1.

8.79.4.4 virtual ObjectBase* ObjectBase::copyBase () const [pure virtual]

Create a copy of the object.

Implemented in **ObjectTile** (p. 239), **ObjectSpecial** (p. 235), **ObjectCharacter** (p. 225), and **ObjectItem** (p. 232).

8.79.4.5 virtual QWidget* ObjectBase::createInformationWidget (QWidget * parent) const [pure virtual]

Create an information widget from an object to show informations about it.

Parameters

<i>parent</i>	The parent for the created information widget.
---------------	--

Returns

The widget that represents this object.

Implemented in **Player** (p. 251), **BrickWall** (p. 67), and **StartPosition** (p. 349).

8.79.4.6 void ObjectBase::deserialize (QDataStream & dataStream) [virtual]

Reimplemented in **ObjectTile** (p. 239), **ObjectSpecial** (p. 236), **ObjectCharacter** (p. 225), **ObjectItem** (p. 232), **Player** (p. 251), **BrickWall** (p. 67), and **StartPosition** (p. 349).

8.79.4.7 virtual void ObjectBase::deserialize (QIODevice * device) [pure virtual]

Implements **Serialize** (p. 314).

Implemented in **ObjectTile** (p. 239), **ObjectSpecial** (p. 236), **ObjectCharacter** (p. 225), **ObjectItem** (p. 232), **Player** (p. 251), **BrickWall** (p. 67), and **StartPosition** (p. 349).

8.79.4.8 virtual QMap<QString, BaseCommand *> ObjectBase::getCommandList () const [pure virtual]

Implemented in **Player** (p. 251), **BrickWall** (p. 67), and **StartPosition** (p. 349).

8.79.4.9 QString ObjectBase::getCurrentImagePath () const

8.79.4.10 QString ObjectBase::getDescription () const

8.79.4.11 QVector<QString > ObjectBase::getImagePathList () const

8.79.4.12 bool ObjectBase::getIsVisible () const

8.79.4.13 virtual int ObjectBase::getObjectIDNumber () const [pure virtual]

Implemented in **ObjectTile** (p. 239), **ObjectSpecial** (p. 236), **ObjectCharacter** (p. 225), and **ObjectItem** (p. 232).

8.79.4.14 QString ObjectBase::getObjectName () const

8.79.4.15 unsigned int ObjectBase::getSpawnID () const

8.79.4.16 virtual ObjectType ObjectBase::getType () const [pure virtual]

Implemented in **ObjectTile** (p. 239), **ObjectSpecial** (p. 236), **ObjectCharacter** (p. 225), and **ObjectItem** (p. 232).

8.79.4.17 int ObjectBase::getUsedImageIndex () const

8.79.4.18 int ObjectBase::getX () const

8.79.4.19 int ObjectBase::getY () const

8.79.4.20 int ObjectBase::getZ () const

8.79.4.21 void ObjectBase::initObjectBase (const QString & name = "", const QString & description = "", int usedImageIndex = -1, bool isVisible = true) [protected]

Initialize Attributes.

8.79.4.22 void ObjectBase::initObjectBase (const QVector< QString > & imagePathList, const QString & name = "", const QString & description = "", int usedImageIndex = -1, bool isVisible = true) [protected]

8.79.4.23 bool ObjectBase::isSet () const [virtual]

Checks if position is (-1,-1,-1) or not. If it's a mix, something went wrong.

Returns

8.79.4.24 void ObjectBase::loadImage (const QString & imagePath)

Loads an image from a path. Clears the vector of images and sets a new one at position. *usedImageIndex* to 0.

Parameters

<i>imagePath</i>	Is the path to the image used.
------------------	--------------------------------

8.79.4.25 void ObjectBase::loadImages (const QVector< QString > & imagePathList)

Clears and then appends a new list of images. *usedImageIndex* will be set to 0.

Parameters

<i>imagePathList</i>	A list containing the paths to all the images to be loaded.
----------------------	---

8.79.4.26 QImage * ObjectBase::refImage ()

Reference the image given by the *usedImageIndex*.

Returns

A image from the *imageList* or NULL, if the *usedImageIndex* was out-of-range or if the object isn't visible.

8.79.4.27 QImage * ObjectBase::refImage (int *index*)

Reference the image given by the given *index*.

Parameters

<i>index</i>	The index of whichever image from the <i>imageList</i> .
--------------	--

Returns

A image from the *imageList* or NULL, if the *index* was out-of-range or if the object isn't visible.

8.79.4.28 void ObjectBase::serialize (QDataStream & *dataStream*) const [virtual]

Reimplemented in **ObjectTile** (p. 239), **ObjectSpecial** (p. 236), **ObjectCharacter** (p. 225), **ObjectItem** (p. 233), **Player** (p. 252), **BrickWall** (p. 68), and **StartPosition** (p. 349).

8.79.4.29 virtual void ObjectBase::serialize (QIODevice * *device*) const [pure virtual]

Implements **Serialize** (p. 314).

Implemented in **ObjectTile** (p. 239), **ObjectSpecial** (p. 236), **ObjectCharacter** (p. 226), **ObjectItem** (p. 233), **Player** (p. 252), **BrickWall** (p. 68), and **StartPosition** (p. 349).

8.79.4.30 void ObjectBase::setDescription (const QString & *description*)

8.79.4.31 void ObjectBase::setIsVisible (bool *isVisible*)

8.79.4.32 void ObjectBase::setPosition (int *x*, int *y*, int *z*)

8.79.4.33 void ObjectBase::setSpawnID (unsigned int *spawnID*)

8.79.4.34 void ObjectBase::setUsedImageIndex (int *usedImageIndex*)

8.79.4.35 void ObjectBase::toggleVisibility ()

Toggle visibility of an object.

8.79.4.36 void ObjectBase::unsetPosition ()

Set the positions to (-1,-1,-1) to indicate the unset state.

8.79.4.37 virtual void ObjectBase::update (const ObjectBase & *object*) [pure virtual]

Implemented in **ObjectTile** (p. 239), **ObjectSpecial** (p. 236), **ObjectCharacter** (p. 226), **ObjectItem** (p. 233), **Player** (p. 252), **BrickWall** (p. 68), and **StartPosition** (p. 350).

8.79.4.38 void ObjectBase::updateBase (const ObjectBase & *object*)

Updates this object with another object.

Parameters

<i>object</i>	The object to be updating with.
---------------	---------------------------------

8.79.5 Member Data Documentation

8.79.5.1 QString ObjectBase::description [protected]

Description of this object.

8.79.5.2 QVector<ObjectImage> ObjectBase::imageList [protected]

The list of images shown and their respective paths.

8.79.5.3 bool ObjectBase::isVisible [protected]

Represents the image to be shown in *imageList*. Can't be less than 0 or more than the *imageList.size()-1*. In case of -1, there is no image to be shown in the list.

True, if you want to show the image.

8.79.5.4 unsigned int ObjectBase::spawnID [protected]

Denotes the spawn of that object. UINT_MAX when initialized.

8.79.5.5 int ObjectBase::usedImageIndex [protected]

8.79.5.6 int ObjectBase::x [protected]

8.79.5.7 int ObjectBase::y [protected]

8.79.5.8 int ObjectBase::z [protected]

The position of the objects. (-1,-1,-1), if not set anywhere.

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

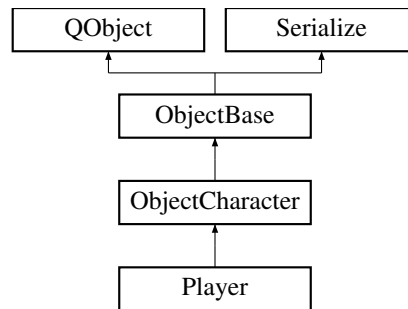
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/**ObjectBase.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/**ObjectBase.cpp**

8.80 ObjectCharacter Class Reference

The baseclass for the Characters to be placed into the game's mainView.

```
#include <ObjectCharacter.h>
```

Inheritance diagram for ObjectCharacter:



Public Member Functions

- **ObjectCharacter** (const QString &name="", const QString &description="", const QString &imagePath="", bool isVisible=true, QObject *parent=0)
- **ObjectCharacter** (const **ObjectCharacter** &objectCharacter)
- virtual **ObjectCharacter** * **copy** () const =0
- virtual **ObjectBase** * **copyBase** () const
Create a copy of the object.
- void **updateCharacter** (const **ObjectCharacter** &character)
- virtual void **update** (const **ObjectBase** &object)=0
- virtual **ObjectBase::ObjectType** **getType** () const
- virtual **ObjectID::CharacterID** **getObjectID** () const =0
- virtual int **getObjectIDNumber** () const
- virtual void **serialize** (QDataStream &dataStream) const
- virtual void **serialize** (QIODevice *device) const =0
- virtual void **deserialize** (QDataStream &dataStream)
- virtual void **deserialize** (QIODevice *device)=0

Protected Member Functions

- void **initObjectCharacter** ()
Initialize attributes.

Additional Inherited Members

8.80.1 Detailed Description

The baseclass for the Characters to be placed into the game's mainView.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.80.2 Constructor & Destructor Documentation

8.80.2.1 `ObjectCharacter::ObjectCharacter (const QString & name = "", const QString & description = "", const QString & imagePath = "", bool isVisible = true, QObject * parent = 0)` `[explicit]`

8.80.2.2 `ObjectCharacter::ObjectCharacter (const ObjectCharacter & objectCharacter)` `[explicit]`

8.80.3 Member Function Documentation

8.80.3.1 `virtual ObjectCharacter* ObjectCharacter::copy () const` `[pure virtual]`

Implemented in **Player** (p. 251).

8.80.3.2 `ObjectBase * ObjectCharacter::copyBase () const` `[virtual]`

Create a copy of the object.

Implements **ObjectBase** (p. 220).

8.80.3.3 `void ObjectCharacter::deserialize (QDataStream & dataStream)` `[virtual]`

Reimplemented from **ObjectBase** (p. 220).

Reimplemented in **Player** (p. 251).

8.80.3.4 `virtual void ObjectCharacter::deserialize (QIODevice * device)` `[pure virtual]`

Implements **ObjectBase** (p. 220).

Implemented in **Player** (p. 251).

8.80.3.5 `virtual ObjectID::CharacterID ObjectCharacter::getObjectID () const` `[pure virtual]`

Implemented in **Player** (p. 251).

8.80.3.6 `int ObjectCharacter::getObjectIDNumber () const` `[virtual]`

Implements **ObjectBase** (p. 220).

8.80.3.7 `ObjectBase::ObjectType ObjectCharacter::getType () const` `[virtual]`

Implements **ObjectBase** (p. 221).

8.80.3.8 `void ObjectCharacter::initObjectCharacter ()` `[protected]`

Initialize attributes.

8.80.3.9 `void ObjectCharacter::serialize (QDataStream & dataStream) const` `[virtual]`

Reimplemented from **ObjectBase** (p. 222).

Reimplemented in **Player** (p. 252).

8.80.3.10 `virtual void ObjectCharacter::serialize (QIODevice * device) const` [pure virtual]

Implements **ObjectBase** (p. 222).

Implemented in **Player** (p. 252).

8.80.3.11 `virtual void ObjectCharacter::update (const ObjectBase & object)` [pure virtual]

Implements **ObjectBase** (p. 222).

Implemented in **Player** (p. 252).

8.80.3.12 `void ObjectCharacter::updateCharacter (const ObjectCharacter & character)`

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

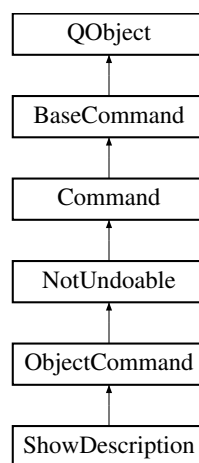
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/**ObjectCharacter.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/**ObjectCharacter.cpp**

8.81 ObjectCommand Class Reference

A command corresponding and holding to a given object.

```
#include <ObjectCommand.h>
```

Inheritance diagram for ObjectCommand:



Public Member Functions

- **ObjectCommand** (const **ObjectBase** ***object**, QObject *parent=0)
- virtual bool **isExecutable** () const
Check if the command is executable.
- virtual bool **execute** ()=0
Executes the command.

Protected Attributes

- const **ObjectBase** * **object**
The object this command works with.

8.81.1 Detailed Description

A command corresponding and holding to a given object.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.81.2 Constructor & Destructor Documentation

8.81.2.1 `ObjectCommand::ObjectCommand (const ObjectBase * object, QObject * parent = 0)`

8.81.3 Member Function Documentation

8.81.3.1 `virtual bool ObjectCommand::execute ()` [pure virtual]

Executes the command.

Returns

True, if successful.

Implements **NotUndoable** (p. 215).

Implemented in **ShowDescription** (p. 328).

8.81.3.2 `bool ObjectCommand::isExecutable () const` [virtual]

Check if the command is executable.

Returns

If the command is executable.

Reimplemented from **NotUndoable** (p. 216).

8.81.4 Member Data Documentation

8.81.4.1 `const ObjectBase* ObjectCommand::object` [protected]

The object this command works with.

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

- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Object/**ObjectCommand.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Object/**ObjectCommand.cpp**

8.82 ObjectID Class Reference

Information and management of spawnIDs for the corresponding ObjectIDs.

```
#include <ObjectID.h>
```

Public Types

- enum **CharacterID** { **PLAYER** = 0 }
- enum **ItemID** { **KNIFE** = 0 }
- enum **TileID** { **BRICK_WALL** = 0 }
- enum **SpecialID** { **START_POSITION** = 0 }

Public Member Functions

- **ObjectID** ()
- unsigned int **getNewSpawnIDFor** (**ObjectID::CharacterID** id)
- unsigned int **getNewSpawnIDFor** (**ObjectID::ItemID** id)
- unsigned int **getNewSpawnIDFor** (**ObjectID::SpecialID** id)
- unsigned int **getNewSpawnIDFor** (**ObjectID::TileID** id)
- virtual void **serialize** (QDataStream &stream) const
- virtual void **deserialize** (QDataStream &stream)

Private Member Functions

- unsigned int **getNewSpawnIDFromMap** (QMap< int, unsigned int > &map, int id)
- void **serialize** (const QMap< int, unsigned int > &map, QDataStream &stream) const
- void **deserialize** (QMap< int, unsigned int > &map, QDataStream &stream)

Private Attributes

- QMap< int, unsigned int > **characterSpawnIDMap**
Contains all the possible spawnIDs for characters. Start at 1, if no ID existed before.
- QMap< int, unsigned int > **itemSpawnIDMap**
Contains all the possible spawnIDs for items. Start at 1, if no ID existed before.
- QMap< int, unsigned int > **specialSpawnIDMap**
Contains all the possible spawnIDs for specials. Start at 1, if no ID existed before.
- QMap< int, unsigned int > **tileSpawnIDMap**
Contains all the possible spawnIDs for tiles. Start at 1, if no ID existed before.

8.82.1 Detailed Description

Information and management of spawnIDs for the corresponding ObjectIDs.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.82.2 Member Enumeration Documentation

8.82.2.1 enum ObjectID::CharacterID

Enumerator

PLAYER

8.82.2.2 enum ObjectID::ItemID

Enumerator

KNIFE

8.82.2.3 enum ObjectID::SpecialID

Enumerator

START_POSITION

8.82.2.4 enum ObjectID::TileID

Enumerator

BRICK_WALL

8.82.3 Constructor & Destructor Documentation

8.82.3.1 ObjectID::ObjectID ()

8.82.4 Member Function Documentation

8.82.4.1 void ObjectID::deserialize (QMap< int, unsigned int > & *map*, QDataStream & *stream*) [private]8.82.4.2 void ObjectID::deserialize (QDataStream & *stream*) [virtual]8.82.4.3 unsigned int ObjectID::getNewSpawnIDFor (ObjectID::CharacterID *id*)8.82.4.4 unsigned int ObjectID::getNewSpawnIDFor (ObjectID::ItemID *id*)8.82.4.5 unsigned int ObjectID::getNewSpawnIDFor (ObjectID::SpecialID *id*)8.82.4.6 unsigned int ObjectID::getNewSpawnIDFor (ObjectID::TileID *id*)8.82.4.7 unsigned int ObjectID::getNewSpawnIDFromMap (QMap< int, unsigned int > & *map*, int *id*) [private]8.82.4.8 void ObjectID::serialize (const QMap< int, unsigned int > & *map*, QDataStream & *stream*) const [private]8.82.4.9 void ObjectID::serialize (QDataStream & *stream*) const [virtual]

8.82.5 Member Data Documentation

8.82.5.1 QMap<int, unsigned int> ObjectID::characterSpawnIDMap [private]

Contains all the possible spawnIDs for characters. Start at 1, if no ID existed before.

8.82.5.2 QMap<int, unsigned int> ObjectID::itemSpawnIDMap [private]

Contains all the possible spawnIDs for items. Start at 1, if no ID existed before.

8.82.5.3 QMap<int, unsigned int> ObjectID::specialSpawnIDMap [private]

Contains all the possible spawnIDs for specials. Start at 1, if no ID existed before.

8.82.5.4 QMap<int, unsigned int> ObjectID::tileSpawnIDMap [private]

Contains all the possible spawnIDs for tiles. Start at 1, if no ID existed before.

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

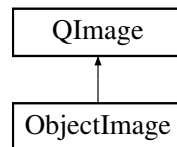
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/**ObjectID.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/**ObjectID.cpp**

8.83 ObjectImage Class Reference

Representing and QImage which knows the path it has been loaded from.

```
#include <ObjectImage.h>
```

Inheritance diagram for ObjectImage:



Public Member Functions

- **ObjectImage** ()
- **ObjectImage** (const QString &imagePath)
- QString **getPath** () const

Private Attributes

- QString **path**

8.83.1 Detailed Description

Representing and QImage which knows the path it has been loaded from.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.83.2 Constructor & Destructor Documentation

8.83.2.1 ObjectImage::ObjectImage () [explicit]

8.83.2.2 ObjectImage::ObjectImage (const QString & *imagePath*) [explicit]

8.83.3 Member Function Documentation

8.83.3.1 QString ObjectImage::getPath () const

8.83.4 Member Data Documentation

8.83.4.1 QString ObjectImage::path [private]

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

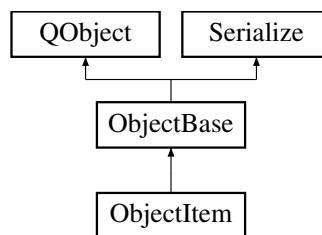
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/**ObjectImage.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/**ObjectImage.cpp**

8.84 ObjectItem Class Reference

The baseclass for the Items to be placed into the game's mainView.

```
#include <ObjectItem.h>
```

Inheritance diagram for ObjectItem:



Public Member Functions

- **ObjectItem** (const QString &name="", const QString &**description**="", const QString &imagePath="", bool **isVisible**=true, QObject *parent=0)
- **ObjectItem** (const **ObjectItem** &objectItem)
- virtual **ObjectItem** * **copy** () const =0
- virtual **ObjectBase** * **copyBase** () const
Create a copy of the object.
- virtual void **update** (const **ObjectBase** &object)=0
- void **updateItem** (const **ObjectItem** &item)
- virtual **ObjectBase::ObjectType** **getType** () const
- virtual **ObjectID::ItemID** **getObjectID** () const =0
- virtual int **getObjectIDNumber** () const
- virtual void **serialize** (QDataStream &dataStream) const
- virtual void **serialize** (QIODevice *device) const =0
- virtual void **deserialize** (QDataStream &dataStream)
- virtual void **deserialize** (QIODevice *device)=0

Protected Member Functions

- void **initObjectItem** ()
Initialize Attributes.

Additional Inherited Members

8.84.1 Detailed Description

The baseclass for the Items to be placed into the game's mainView.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.84.2 Constructor & Destructor Documentation

8.84.2.1 `ObjectItem::ObjectItem (const QString & name = " ", const QString & description = " ", const QString & imagePath = " ", bool isVisible = true, QObject * parent = 0) [explicit]`

8.84.2.2 `ObjectItem::ObjectItem (const ObjectItem & objectItem) [explicit]`

8.84.3 Member Function Documentation

8.84.3.1 `virtual ObjectItem* ObjectItem::copy () const [pure virtual]`

8.84.3.2 `ObjectBase * ObjectItem::copyBase () const [virtual]`

Create a copy of the object.

Implements **ObjectBase** (p. 220).

8.84.3.3 `void ObjectItem::deserialize (QDataStream & dataStream) [virtual]`

Reimplemented from **ObjectBase** (p. 220).

8.84.3.4 `virtual void ObjectItem::deserialize (QIODevice * device) [pure virtual]`

Implements **ObjectBase** (p. 220).

8.84.3.5 `virtual ObjectID::ItemID ObjectItem::getObjectID () const [pure virtual]`

8.84.3.6 `int ObjectItem::getObjectIDNumber () const [virtual]`

Implements **ObjectBase** (p. 220).

8.84.3.7 `ObjectBase::ObjectType ObjectItem::getType () const [virtual]`

Implements **ObjectBase** (p. 221).

8.84.3.8 `void ObjectItem::initObjectItem () [protected]`

Initialize Attributes.

8.84.3.9 `void ObjectItem::serialize (QDataStream & dataStream) const` [virtual]

Reimplemented from **ObjectBase** (p. 222).

8.84.3.10 `virtual void ObjectItem::serialize (QIODevice * device) const` [pure virtual]

Implements **ObjectBase** (p. 222).

8.84.3.11 `virtual void ObjectItem::update (const ObjectBase & object)` [pure virtual]

Implements **ObjectBase** (p. 222).

8.84.3.12 `void ObjectItem::updateItem (const ObjectItem & item)`

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

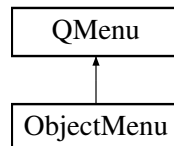
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/**ObjectItem.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/**ObjectItem.cpp**

8.85 ObjectMenu Class Reference

The menu which is created by the command actions created through the given object.

```
#include <ObjectMenu.h>
```

Inheritance diagram for ObjectMenu:



Public Member Functions

- **ObjectMenu** (**ObjectBase** ***object**, **Processor** ***processor**, QWidget ***parent**=0)

Private Slots

- void **triggerCommandByName** (const QString &**name**)

Private Attributes

- **Processor** * **processor**
- **ObjectBase** * **object**

The copy of a given object which is constant here. If any changes are to be made to this, they won't be shown on the actual object and vice versa.

8.85.1 Detailed Description

The menu which is created by the command actions created through the given object.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.85.2 Constructor & Destructor Documentation

8.85.2.1 `ObjectMenu::ObjectMenu (ObjectBase * object, Processor * processor, QWidget * parent = 0)`

8.85.3 Member Function Documentation

8.85.3.1 `void ObjectMenu::triggerCommandByName (const QString & name)` [private], [slot]

8.85.4 Member Data Documentation

8.85.4.1 `ObjectBase* ObjectMenu::object` [private]

The copy of a given object which is constant here. If any changes are to be made to this, they won't be shown on the actual object and vice versa.

8.85.4.2 `Processor* ObjectMenu::processor` [private]

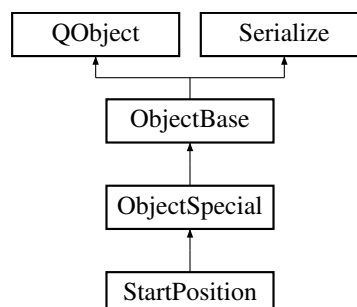
The documentation for this class was generated from the following files:

- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/**ObjectMenu.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/**ObjectMenu.cpp**

8.86 ObjectSpecial Class Reference

```
#include <ObjectSpecial.h>
```

Inheritance diagram for ObjectSpecial:



Public Member Functions

- **ObjectSpecial** (int **maxAmountPerStack**=1, bool **visibleInGame**=false, const QString &**name**="", const QString &**description**="", const QString &**imagePath**="", bool **isVisible**=true, QObject ***parent**=0)
- **ObjectSpecial** (const **ObjectSpecial** &**objectSpecial**)
- virtual **ObjectSpecial** * **copy** () const =0
- virtual **ObjectBase** * **copyBase** () const
Create a copy of the object.
- virtual void **update** (const **ObjectBase** &**object**)=0
- void **updateSpecial** (const **ObjectSpecial** &**special**)
- virtual **ObjectBase::ObjectType** **getType** () const
- virtual **ObjectID::SpecialID** **getObjectID** () const =0
- virtual int **getObjectIDNumber** () const
- int **getMaxAmountPerStack** () const
- bool **isVisibleInGame** () const
- void **setMaxAmountPerStack** (int **maxAmountPerStack**)
- void **setVisibleInGame** (bool **visibleInGame**)
- virtual void **serialize** (QDataStream &**dataStream**) const
- virtual void **serialize** (QIODevice ***device**) const =0
- virtual void **deserialize** (QDataStream &**dataStream**)
- virtual void **deserialize** (QIODevice ***device**)=0

Protected Member Functions

- void **initObjectSpecial** (int **maxAmountPerStack**=1, bool **visibleInGame**=false)
Initialize Attributes.

Private Attributes

- int **maxAmountPerStack**
- bool **visibleInGame**

Additional Inherited Members

8.86.1 Constructor & Destructor Documentation

8.86.1.1 **ObjectSpecial::ObjectSpecial** (int *maxAmountPerStack* = 1, bool *visibleInGame* = false, const QString & *name* = " ", const QString & *description* = " ", const QString & *imagePath* = " ", bool *isVisible* = true, QObject * *parent* = 0) [explicit]

8.86.1.2 **ObjectSpecial::ObjectSpecial** (const **ObjectSpecial** & *objectSpecial*) [explicit]

8.86.2 Member Function Documentation

8.86.2.1 virtual **ObjectSpecial*** **ObjectSpecial::copy** () const [pure virtual]

Implemented in **StartPosition** (p. 349).

8.86.2.2 **ObjectBase** * **ObjectSpecial::copyBase** () const [virtual]

Create a copy of the object.

Implements **ObjectBase** (p. 220).

8.86.2.3 `void ObjectSpecial::deserialize (QDataStream & dataStream) [virtual]`

Reimplemented from **ObjectBase** (p. 220).

Reimplemented in **StartPosition** (p. 349).

8.86.2.4 `virtual void ObjectSpecial::deserialize (QIODevice * device) [pure virtual]`

Implements **ObjectBase** (p. 220).

Implemented in **StartPosition** (p. 349).

8.86.2.5 `int ObjectSpecial::getMaxAmountPerStack () const`

8.86.2.6 `virtual ObjectID::SpecialID ObjectSpecial::getObjectID () const [pure virtual]`

Implemented in **StartPosition** (p. 349).

8.86.2.7 `int ObjectSpecial::getObjectIDNumber () const [virtual]`

Implements **ObjectBase** (p. 220).

8.86.2.8 `ObjectBase::ObjectType ObjectSpecial::getType () const [virtual]`

Implements **ObjectBase** (p. 221).

8.86.2.9 `void ObjectSpecial::initObjectSpecial (int maxAmountPerStack = 1, bool visibleInGame = false)
[protected]`

Initialize Attributes.

8.86.2.10 `bool ObjectSpecial::isVisibleInGame () const`

8.86.2.11 `void ObjectSpecial::serialize (QDataStream & dataStream) const [virtual]`

Reimplemented from **ObjectBase** (p. 222).

Reimplemented in **StartPosition** (p. 349).

8.86.2.12 `virtual void ObjectSpecial::serialize (QIODevice * device) const [pure virtual]`

Implements **ObjectBase** (p. 222).

Implemented in **StartPosition** (p. 349).

8.86.2.13 `void ObjectSpecial::setMaxAmountPerStack (int maxAmountPerStack)`

8.86.2.14 `void ObjectSpecial::setVisibleInGame (bool visibleInGame)`

8.86.2.15 `virtual void ObjectSpecial::update (const ObjectBase & object) [pure virtual]`

Implements **ObjectBase** (p. 222).

Implemented in **StartPosition** (p. 350).

8.86.2.16 void ObjectSpecial::updateSpecial (const ObjectSpecial & *special*)

8.86.3 Member Data Documentation

8.86.3.1 int ObjectSpecial::maxAmountPerStack [private]

8.86.3.2 bool ObjectSpecial::visibleInGame [private]

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

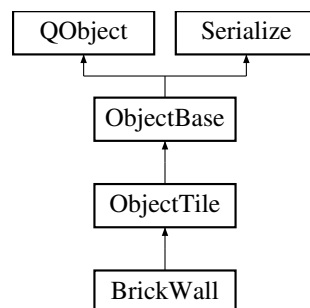
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectSpecial.h
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectSpecial.cpp

8.87 ObjectTile Class Reference

Represents a tile to be put into the mainView.

```
#include <ObjectTile.h>
```

Inheritance diagram for ObjectTile:



Public Types

- enum **TileType** { **FLOOR**, **WALL** }
Specifies the Type of Tile that is used in **ObjectTile** (p. 237).

Public Member Functions

- **ObjectTile** (const QString &name="", const QString &description="", **TileType** tileType=ObjectTile::FLOOR, const QString &imagePath="", bool isVisible=true, QObject *parent=0)
- **ObjectTile** (const **ObjectTile** &objectTile)
- virtual **ObjectTile** * **copy** () const =0
- virtual **ObjectBase** * **copyBase** () const
Create a copy of the object.
- void **update** (const **ObjectBase** &object)=0
- void **updateTile** (const **ObjectTile** &tile)
- virtual **ObjectBase::ObjectType** **getType** () const
- virtual **ObjectID::TileID** **getObjectID** () const =0
- virtual int **getObjectIDNumber** () const
- **TileType** **getTileType** () const
- void **setTileType** (**TileType** tileType)
- virtual void **serialize** (QDataStream &dataStream) const
- virtual void **serialize** (QIODevice *device) const =0

- virtual void **deserialize** (QDataStream &dataStream)
- virtual void **deserialize** (QIODevice *device)=0

Protected Member Functions

- void **initObjectTile** (**TileType** tileType=**ObjectTile::FLOOR**)
Initialize Attributes.

Private Attributes

- **TileType** tileType

Additional Inherited Members

8.87.1 Detailed Description

Represents a tile to be put into the mainView.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.87.2 Member Enumeration Documentation

8.87.2.1 enum **ObjectTile::TileType**

Specifies the Type of Tile that is used in **ObjectTile** (p. 237).

Enumerator

FLOOR

WALL

8.87.3 Constructor & Destructor Documentation

8.87.3.1 **ObjectTile::ObjectTile** (const QString & *name* = " ", const QString & *description* = " ", **TileType** *tileType* = **ObjectTile::FLOOR**, const QString & *imagePath* = " ", bool *isVisible* = true, QObject * *parent* = 0)
[explicit]

8.87.3.2 **ObjectTile::ObjectTile** (const **ObjectTile** & *objectTile*) [explicit]

8.87.4 Member Function Documentation

8.87.4.1 virtual **ObjectTile*** **ObjectTile::copy** () const [pure virtual]

Implemented in **BrickWall** (p. 67).

8.87.4.2 **ObjectBase** * **ObjectTile::copyBase** () const [virtual]

Create a copy of the object.

Implements **ObjectBase** (p. 220).

8.87.4.3 void **ObjectTile::deserialize** (**QDataStream** & *dataStream*) [virtual]

Reimplemented from **ObjectBase** (p. 220).

Reimplemented in **BrickWall** (p. 67).

8.87.4.4 virtual void **ObjectTile::deserialize** (**QIODevice** * *device*) [pure virtual]

Implements **ObjectBase** (p. 220).

Implemented in **BrickWall** (p. 67).

8.87.4.5 virtual **ObjectID::TileID** **ObjectTile::getObjectID** () const [pure virtual]

Implemented in **BrickWall** (p. 68).

8.87.4.6 int **ObjectTile::getObjectIDNumber** () const [virtual]

Implements **ObjectBase** (p. 220).

8.87.4.7 **ObjectTile::TileType** **ObjectTile::getTileType** () const

8.87.4.8 **ObjectBase::ObjectType** **ObjectTile::getType** () const [virtual]

Implements **ObjectBase** (p. 221).

8.87.4.9 void **ObjectTile::initObjectTile** (**TileType** *tileType* = **ObjectTile::FLOOR**) [protected]

Initialize Attributes.

8.87.4.10 void **ObjectTile::serialize** (**QDataStream** & *dataStream*) const [virtual]

Reimplemented from **ObjectBase** (p. 222).

Reimplemented in **BrickWall** (p. 68).

8.87.4.11 virtual void **ObjectTile::serialize** (**QIODevice** * *device*) const [pure virtual]

Implements **ObjectBase** (p. 222).

Implemented in **BrickWall** (p. 68).

8.87.4.12 void **ObjectTile::setTileType** (**TileType** *tileType*)

8.87.4.13 void **ObjectTile::update** (const **ObjectBase** & *object*) [pure virtual]

Implements **ObjectBase** (p. 222).

Implemented in **BrickWall** (p. 68).

8.87.4.14 void **ObjectTile::updateTile** (const **ObjectTile** & *tile*)

8.87.5 Member Data Documentation

8.87.5.1 **TileType** **ObjectTile::tileType** [private]

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

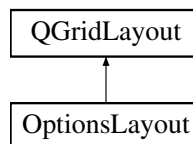
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/**ObjectTile.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/**ObjectTile.cpp**

8.88 OptionsLayout Class Reference

The layout for options to be chosen.

```
#include <OptionsLayout.h>
```

Inheritance diagram for OptionsLayout:



Signals

- void **currentIndexChangedTo** (int)

Public Member Functions

- **OptionsLayout** (QWidget *parent=0)
- void **addPropertyWidget** (const QString &listEntry, QWidget *widget)
- void **setPropertyListWidth** (int propertyListWidth)
- QPushButton * **refOkButton** ()
- QPushButton * **refApplyButton** ()

Private Slots

- void **showCurrentWidget** ()

Private Attributes

- List * **propertyList**
- QWidget * **stackedProperties**
- QStackedLayout * **stackLayout**
- QPushButton * **okButton**
- QPushButton * **applyButton**

8.88.1 Detailed Description

The layout for options to be chosen.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.88.2 Constructor & Destructor Documentation

8.88.2.1 OptionsLayout::OptionsLayout (QWidget * *parent* = 0)

8.88.3 Member Function Documentation

8.88.3.1 void OptionsLayout::addPropertyWidget (const QString & *listEntry*, QWidget * *widget*)

8.88.3.2 void OptionsLayout::currentIndexChangedTo (int) [signal]

8.88.3.3 QPushButton * OptionsLayout::refApplyButton ()

8.88.3.4 QPushButton * OptionsLayout::refOkButton ()

8.88.3.5 void OptionsLayout::setPropertyListWidth (int *propertyListWidth*)

8.88.3.6 void OptionsLayout::showCurrentWidget () [private],[slot]

8.88.4 Member Data Documentation

8.88.4.1 QPushButton* OptionsLayout::applyButton [private]

8.88.4.2 QPushButton* OptionsLayout::okButton [private]

8.88.4.3 List* OptionsLayout::propertyList [private]

8.88.4.4 QWidget* OptionsLayout::stackedProperties [private]

8.88.4.5 QStackedLayout* OptionsLayout::stackLayout [private]

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

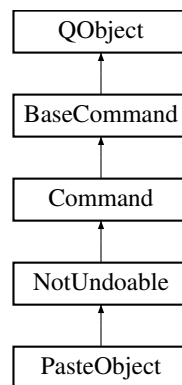
- /home/ryoga/git/pidgirl/pidgirl-engine/Common/OptionsLayout.h
- /home/ryoga/git/pidgirl/pidgirl-engine/Common/OptionsLayout.cpp

8.89 PasteObject Class Reference

Paste an object from the clipboard.

```
#include <PasteObject.h>
```

Inheritance diagram for PasteObject:



Public Member Functions

- **PasteObject** (**StackModel** ***stackModel**, QObject *parent=0)
- virtual bool **execute** ()
Executes the command.

Private Attributes

- **StackModel** * **stackModel**
The stack to paste the given clipboard object to.

Additional Inherited Members

8.89.1 Detailed Description

Paste an object from the clipboard.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.89.2 Constructor & Destructor Documentation

8.89.2.1 **PasteObject::PasteObject** (**StackModel** * *stackModel*, QObject * *parent* = 0)

8.89.3 Member Function Documentation

8.89.3.1 bool **PasteObject::execute** () [virtual]

Executes the command.

Returns

True, if successful.

Implements **NotUndoable** (p. 215).

8.89.4 Member Data Documentation**8.89.4.1 StackModel* PasteObject::stackModel** [private]

The stack to paste the given clipboard object to.

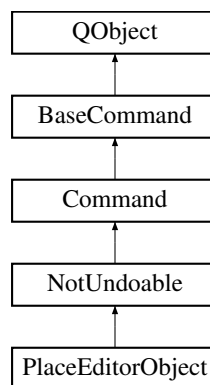
The documentation for this class was generated from the following files:

- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/**PasteObject.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/**PasteObject.cpp**

8.90 PlaceEditorObject Class Reference

```
#include <PlaceEditorObject.h>
```

Inheritance diagram for PlaceEditorObject:

**Public Member Functions**

- **PlaceEditorObject** (**StackModel** *stackModel, **ObjectBase** *object=0, **QObject** *parent=0)
- virtual bool **execute** ()
Executes the command.

Private Attributes

- **ObjectBase** * placeableObject
The object to be placed.
- **StackModel** * stackModel
The stack where the object will be placed to.

Additional Inherited Members**8.90.1 Constructor & Destructor Documentation**

8.90.1.1 `PlaceEditorObject::PlaceEditorObject (StackModel * stackModel, ObjectBase * object = 0, QObject * parent = 0)`

8.90.2 Member Function Documentation

8.90.2.1 `bool PlaceEditorObject::execute () [virtual]`

Executes the command.

Returns

True, if successful.

Implements **NotUndoable** (p. 215).

8.90.3 Member Data Documentation

8.90.3.1 `ObjectBase* PlaceEditorObject::placeableObject [private]`

The object to be placed.

8.90.3.2 `StackModel* PlaceEditorObject::stackModel [private]`

The stack where the object will be placed to.

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

- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/PlaceEditorObject.h`
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/PlaceEditorObject.cpp`

8.91 PlaceObject Class Reference

Place an object.

```
#include <PlaceEditorObject.h>
```

8.91.1 Detailed Description

Place an object.

Author

RyogaU

Version

0.5

Contact: `RyogaU@googlemail.com`

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

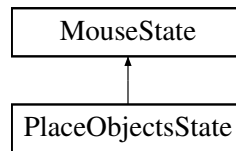
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/PlaceEditorObject.h`

8.92 PlaceObjectsState Class Reference

The state in which objects will be placed.

```
#include <PlaceObjectsState.h>
```

Inheritance diagram for PlaceObjectsState:



Public Member Functions

- **PlaceObjectsState** (**Processor** *processor, **StackModel** *targetStack)
- virtual **MouseState** * **nextState** (**MouseState::MouseEventType** type, **QMouseEvent** *event, **StackModel** *targetStack, **StackModel** *fromStack=0)

Determines which state is next, depending on the input. This method might destroy this state and changes occur.

Private Attributes

- **QSet**< **StackModel** * > **alreadyPlaced**

Additional Inherited Members

8.92.1 Detailed Description

The state in which objects will be placed.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.92.2 Constructor & Destructor Documentation

8.92.2.1 **PlaceObjectsState::PlaceObjectsState** (**Processor** * processor, **StackModel** * targetStack)

8.92.3 Member Function Documentation

8.92.3.1 **MouseState** * **PlaceObjectsState::nextState** (**MouseState::MouseEventType** type, **QMouseEvent** * event, **StackModel** * targetStack, **StackModel** * fromStack = 0) [virtual]

Determines which state is next, depending on the input. This method might destroy this state and changes occur.

Parameters

<i>event</i>	Holds the event that has been triggered.
<i>stack</i>	The stack where this method has been invoked from.

Returns

a pointer to the next state.

Implements **MouseState** (p. 197).

8.92.4 Member Data Documentation

8.92.4.1 QSet<StackModel *> PlaceObjectsState::alreadyPlaced [private]

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

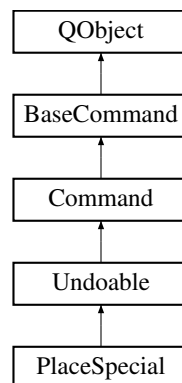
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/**PlaceObjectsState.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/**PlaceObjectsState.cpp**

8.93 PlaceSpecial Class Reference

Place a special on a chosen stack.

```
#include <PlaceSpecial.h>
```

Inheritance diagram for PlaceSpecial:



Public Member Functions

- **PlaceSpecial** (**ObjectSpecial** ***special**, **StackModel** ***stackModel**, **QObject** ***parent**=0)
- virtual bool **execute** ()
Executes the command.
- virtual void **undo** ()
Undo changes made by the command.
- virtual void **redo** ()
Redo changes made by the undoing of the command.

Private Attributes

- **ObjectSpecial * special**

The special to be placed.

- **StackModel * stackModel**

The stack where the object will be placed to.

Additional Inherited Members

8.93.1 Detailed Description

Place a special on a chosen stack.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.93.2 Constructor & Destructor Documentation

8.93.2.1 `PlaceSpecial::PlaceSpecial (ObjectSpecial * special, StackModel * stackModel, QObject * parent = 0)`
[explicit]

8.93.3 Member Function Documentation

8.93.3.1 `bool PlaceSpecial::execute ()` [virtual]

Executes the command.

Returns

True, if successful.

Implements **Undoable** (p. 354).

8.93.3.2 `void PlaceSpecial::redo ()` [virtual]

Redo changes made by the undoing of the command.

Implements **Undoable** (p. 355).

8.93.3.3 `void PlaceSpecial::undo ()` [virtual]

Undo changes made by the command.

Implements **Undoable** (p. 355).

8.93.4 Member Data Documentation

8.93.4.1 `ObjectSpecial* PlaceSpecial::special` [private]

The special to be placed.

8.93.4.2 `StackModel* PlaceSpecial::stackModel` [private]

The stack where the object will be placed to.

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

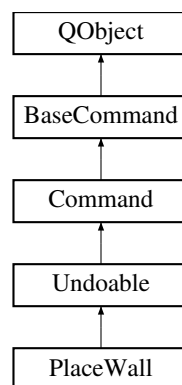
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/PlaceSpecial.h`
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/PlaceSpecial.cpp`

8.94 PlaceWall Class Reference

Place a wall on a chosen stack.

```
#include <PlaceWall.h>
```

Inheritance diagram for PlaceWall:



Public Member Functions

- **PlaceWall** (**ObjectTile** ***wall**, **StackModel** ***stackModel**, **QObject** *parent=0)
- virtual bool **execute** ()
Executes the command.
- virtual void **undo** ()
Undo changes made by the command.
- virtual void **redo** ()
Redo changes made by the undoing of the command.

Private Attributes

- **ObjectTile** * **wall**
The wall to be placed.
- **ObjectTile** * **oldWall**
The wall that has been replaced.
- **StackModel** * **stackModel**
The stack where the object will be placed to.

Additional Inherited Members

8.94.1 Detailed Description

Place a wall on a chosen stack.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.94.2 Constructor & Destructor Documentation

8.94.2.1 `PlaceWall::PlaceWall (ObjectTile * wall, StackModel * stackModel, QObject * parent = 0) [explicit]`

8.94.3 Member Function Documentation

8.94.3.1 `bool PlaceWall::execute () [virtual]`

Executes the command.

Returns

True, if successful.

Implements **Undoable** (p. 354).

8.94.3.2 `void PlaceWall::redo () [virtual]`

Redo changes made by the undoing of the command.

Implements **Undoable** (p. 355).

8.94.3.3 `void PlaceWall::undo () [virtual]`

Undo changes made by the command.

Implements **Undoable** (p. 355).

8.94.4 Member Data Documentation

8.94.4.1 `ObjectTile* PlaceWall::oldWall [private]`

The wall that has been replaced.

8.94.4.2 `StackModel* PlaceWall::stackModel [private]`

The stack where the object will be placed to.

8.94.4.3 ObjectTile* PlaceWall::wall [private]

The wall to be placed.

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

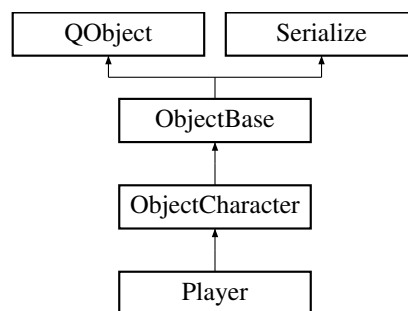
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/**PlaceWall.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/**PlaceWall.cpp**

8.95 Player Class Reference

The player object representing the player.

```
#include <Player.h>
```

Inheritance diagram for Player:



Public Member Functions

- **Player** (const QString &name=tr("Hero"), const QString &description=tr("Just another hero."), QObject *parent=0)
- **Player** (const **Player** &player)
- virtual **Player** * **copy** () const
- virtual void **update** (const **ObjectBase** &object)
- virtual QWidget * **createInformationWidget** (QWidget *parent) const
Create an information widget from an object to show informations about it.
- virtual **ObjectID::CharacterID** **getObjectID** () const
- virtual QMap< QString, **BaseCommand** * > **getCommandList** () const
- virtual void **serialize** (QDataStream &dataStream) const
- virtual void **serialize** (QIODevice *device) const
- virtual void **deserialize** (QDataStream &dataStream)
- virtual void **deserialize** (QIODevice *device)

Protected Member Functions

- void **initPlayer** ()

Additional Inherited Members

8.95.1 Detailed Description

The player object representing the player.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.95.2 Constructor & Destructor Documentation

8.95.2.1 `Player::Player (const QString & name = tr("Hero"), const QString & description = tr("Just another hero."), QObject * parent = 0)` [explicit]

8.95.2.2 `Player::Player (const Player & player)` [explicit]

8.95.3 Member Function Documentation

8.95.3.1 `Player * Player::copy ()` const [virtual]

Implements **ObjectCharacter** (p. 225).

8.95.3.2 `QWidget * Player::createInformationWidget (QWidget * parent)` const [virtual]

Create an information widget from an object to show informations about it.

Parameters

<i>parent</i>	The parent for the created information widget.
---------------	--

Returns

The widget that represents this object.

Implements **ObjectBase** (p. 220).

8.95.3.3 `void Player::deserialize (QDataStream & dataStream)` [virtual]

Reimplemented from **ObjectCharacter** (p. 225).

8.95.3.4 `void Player::deserialize (QIODevice * device)` [virtual]

Implements **ObjectCharacter** (p. 225).

8.95.3.5 `QMap< QString, BaseCommand * > Player::getCommandList ()` const [virtual]

Implements **ObjectBase** (p. 220).

8.95.3.6 `ObjectID::CharacterID Player::getObjectID ()` const [virtual]

Implements **ObjectCharacter** (p. 225).

8.95.3.7 `void Player::initPlayer ()` [protected]

8.95.3.8 `void Player::serialize (QDataStream & dataStream) const` [virtual]

Reimplemented from **ObjectCharacter** (p. 225).

8.95.3.9 `void Player::serialize (QIODevice * device) const` [virtual]

Implements **ObjectCharacter** (p. 226).

8.95.3.10 `void Player::update (const ObjectBase & object)` [virtual]

Implements **ObjectCharacter** (p. 226).

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

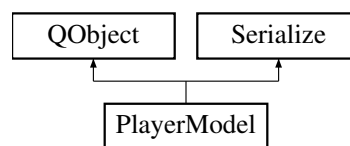
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/Character/**Player.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/Character/**Player.cpp**

8.96 PlayerModel Class Reference

This model represents the informations about a specific player.

```
#include <PlayerModel.h>
```

Inheritance diagram for PlayerModel:



Public Member Functions

- **PlayerModel** (unsigned int **ID**, const QString &localMapPath=QString(""), bool **activePlayer**=true, QObject *parent=0)
- **PlayerModel** (QObject *parent=0)
- unsigned int **getID** () const
- QString **getLocalMapPath** () const
- bool **isActivePlayer** () const
- bool **hasCharacterSpawned** () const
- void **setPlayerMapPath** (const QString &playerMapPath)
- void **setActivePlayer** (bool activePlayer)
- void **setCharacterSpwawned** (bool characterSpawned)
- virtual void **serialize** (QIODevice *device) const
- void **serialize** (QDataStream *dataStream) const
- virtual void **deserialize** (QIODevice *device)
- void **deserialize** (QDataStream *dataStream)

Private Member Functions

- void **init** (unsigned int **ID**=0, const QString &localMapPath=QString(""), bool **activePlayer**=true, bool **characterSpawned**=false)

Private Attributes

- unsigned int **ID**
*The ID of the **Player** (p. 250).*
- bool **activePlayer**
True, if the player is active and playing.
- bool **characterSpawned**
True, if the player has a character spawned.
- QString **playerMapPath**
The map path on which the player is located.

8.96.1 Detailed Description

This model represents the informations about a specific player.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.96.2 Constructor & Destructor Documentation

8.96.2.1 `PlayerModel::PlayerModel (unsigned int ID, const QString & localMapPath = QString(""), bool activePlayer = true, QObject * parent = 0)` [explicit]

8.96.2.2 `PlayerModel::PlayerModel (QObject * parent = 0)` [explicit]

8.96.3 Member Function Documentation

8.96.3.1 `void PlayerModel::deserialize (QIODevice * device)` [virtual]

Implements **Serialize** (p. 314).

8.96.3.2 `void PlayerModel::deserialize (QDataStream * dataStream)`

8.96.3.3 `unsigned int PlayerModel::getID ()` const

8.96.3.4 `QString PlayerModel::getLocalMapPath ()` const

8.96.3.5 `bool PlayerModel::hasCharacterSpawned ()` const

8.96.3.6 `void PlayerModel::init (unsigned int ID = 0, const QString & localMapPath = QString(""), bool activePlayer = true, bool characterSpawned = false)` [private]

8.96.3.7 `bool PlayerModel::isActivePlayer ()` const

8.96.3.8 `void PlayerModel::serialize (QIODevice * device)` const [virtual]

Implements **Serialize** (p. 314).

8.96.3.9 void PlayerModel::serialize (QDataStream * *dataStream*) const

8.96.3.10 void PlayerModel::setActivePlayer (bool *activePlayer*)

8.96.3.11 void PlayerModel::setCharacterSpwawned (bool *characterSpawned*)

8.96.3.12 void PlayerModel::setPlayerMapPath (const QString & *playerMapPath*)

8.96.4 Member Data Documentation

8.96.4.1 bool PlayerModel::activePlayer [private]

True, if the player is active and playing.

8.96.4.2 bool PlayerModel::characterSpawned [private]

True, if the player has a character spawned.

8.96.4.3 unsigned int PlayerModel::ID [private]

The ID of the **Player** (p. 250).

8.96.4.4 QString PlayerModel::playerMapPath [private]

The map path on which the player is located.

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

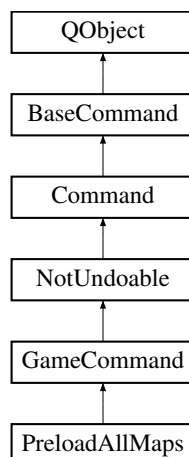
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/**PlayerModel.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/**PlayerModel.cpp**

8.97 PreloadAllMaps Class Reference

Preload the maps if it's a local game or a server.

```
#include <PreloadAllMaps.h>
```

Inheritance diagram for PreloadAllMaps:



Public Member Functions

- **PreloadAllMaps** (QObject *parent=0)
- virtual bool **execute** ()

Executes the command.

Private Member Functions

- QList< QFileInfoList > **preload** (QFileInfoList entries)

Preload maps from a list of files.

Private Attributes

- **FileManager** fileManager

The file manager to preload all the maps in the world folder.

Additional Inherited Members

8.97.1 Detailed Description

Preload the maps if it's a local game or a server.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.97.2 Constructor & Destructor Documentation

8.97.2.1 PreloadAllMaps::PreloadAllMaps (QObject * parent = 0)

8.97.3 Member Function Documentation

8.97.3.1 bool PreloadAllMaps::execute () [virtual]

Executes the command.

Returns

True, if successful.

Implements **GameCommand** (p. 133).

8.97.3.2 QList< QFileInfoList > PreloadAllMaps::preload (QFileInfoList entries) [private]

Preload maps from a list of files.

Parameters

<i>Entries</i>	in the current folder.
----------------	------------------------

Returns

New list of files in found sub folders.

8.97.4 Member Data Documentation

8.97.4.1 FileManager PreloadAllMaps::fileManager [private]

The file manager to preload all the maps in the world folder.

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

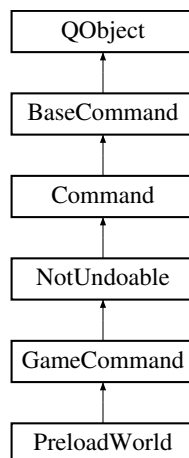
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/**PreloadAllMaps.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/**PreloadAllMaps.cpp**

8.98 PreloadWorld Class Reference

Preload the world.

```
#include <PreloadWorld.h>
```

Inheritance diagram for PreloadWorld:



Public Member Functions

- **PreloadWorld** (QObject *parent=0)
- virtual bool **execute** ()
Executes the command.

Private Attributes

- **FileManager** fileManager
The file manager to preload everything.

Additional Inherited Members

8.98.1 Detailed Description

Preload the world.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.98.2 Constructor & Destructor Documentation

8.98.2.1 PreloadWorld::PreloadWorld (QObject * *parent* = 0)

8.98.3 Member Function Documentation

8.98.3.1 bool PreloadWorld::execute () [virtual]

Executes the command.

Returns

True, if successful.

Implements **GameCommand** (p. 133).

8.98.4 Member Data Documentation

8.98.4.1 FileManager PreloadWorld::fileManager [private]

The file manager to preload everything.

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

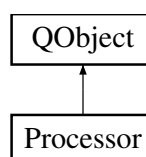
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/**PreloadWorld.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/**PreloadWorld.cpp**

8.99 Processor Class Reference

Process commands and manages them accordingly.

```
#include <Processor.h>
```

Inheritance diagram for Processor:



Signals

- void **commandProcessed** ()
Emmited whenever a command is processed.

Public Member Functions

- **Processor** (QObject *parent=0)
- bool **isExecutable** (**BaseCommand** *command)
Check if the command can be executed.
- bool **execute** (**BaseCommand** *command)
Execute a given command.
- void **setEditorModel** (**EditorModel** *editorModel)
- void **setWorldModel** (**WorldModel** *worldModel)
- void **setGameModel** (**GameModel** *gameModel)
- void **setMapModel** (**MapModel** *mapModel)
- void **setNetworkModel** (**NetworkModel** *networkModel)
- **UndoRedo** * **refUndoRedo** () const

Private Attributes

- **UndoRedo** * **undoRedo**
The Undo and Redo management object.
- **EditorModel** * **editorModel**
- **WorldModel** * **worldModel**
- **GameModel** * **gameModel**
- **MapModel** * **mapModel**
- **NetworkModel** * **networkModel**

8.99.1 Detailed Description

Process commands and manages them accordingly.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.99.2 Constructor & Destructor Documentation

8.99.2.1 **Processor::Processor** (QObject * *parent* = 0) [explicit]

8.99.3 Member Function Documentation

8.99.3.1 void **Processor::commandProcessed** () [signal]

Emmited whenever a command is processed.

8.99.3.2 `bool Processor::execute (BaseCommand * command)`

Execute a given command.

Parameters

<i>command</i>	The command to be executed.
----------------	-----------------------------

Returns

True if the given command was executed successfully. Otherwise, false.

Note

The given command will have the processor set as its parent may be deleted when executed.

8.99.3.3 bool Processor::isExecutable (BaseCommand * *command*)

Check if the command can be executed.

Parameters

<i>command</i>	The command to be checked.
----------------	----------------------------

Returns

True if command is executable. Otherwise, false.

8.99.3.4 UndoRedo * Processor::refUndoRedo () const

8.99.3.5 void Processor::setEditorModel (EditorModel * *editorModel*)

8.99.3.6 void Processor::setGameModel (GameModel * *gameModel*)

8.99.3.7 void Processor::setMapModel (MapModel * *mapModel*)

8.99.3.8 void Processor::setNetworkModel (NetworkModel * *networkModel*)

8.99.3.9 void Processor::setWorldModel (WorldModel * *worldModel*)

8.99.4 Member Data Documentation

8.99.4.1 EditorModel* Processor::editorModel [private]

8.99.4.2 GameModel* Processor::gameModel [private]

8.99.4.3 MapModel* Processor::mapModel [private]

8.99.4.4 NetworkModel* Processor::networkModel [private]

8.99.4.5 UndoRedo* Processor::undoRedo [private]

The Undo and Redo management object.

8.99.4.6 WorldModel* Processor::worldModel [private]

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

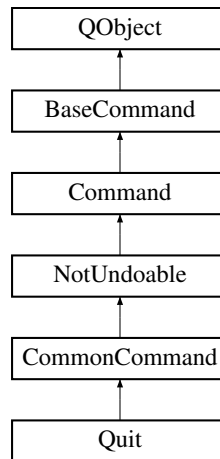
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/**Processor.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/**Processor.cpp**

8.100 Quit Class Reference

Command (p. 93) to quit the running application.

```
#include <Quit.h>
```

Inheritance diagram for Quit:



Public Member Functions

- **Quit** (bool **askToSaveGame**=false, QObject *parent=0)
- virtual bool **execute** ()

Executes the command.

Private Attributes

- bool **askToSaveGame**

Additional Inherited Members

8.100.1 Detailed Description

Command (p. 93) to quit the running application.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.100.2 Constructor & Destructor Documentation

8.100.2.1 **Quit::Quit** (bool *askToSaveGame* = false, QObject * *parent* = 0)

8.100.3 Member Function Documentation

8.100.3.1 bool Quit::execute () [virtual]

Executes the command.

Returns

True, if successful.

Implements **CommonCommand** (p. 98).

8.100.4 Member Data Documentation

8.100.4.1 bool Quit::askToSaveGame [private]

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

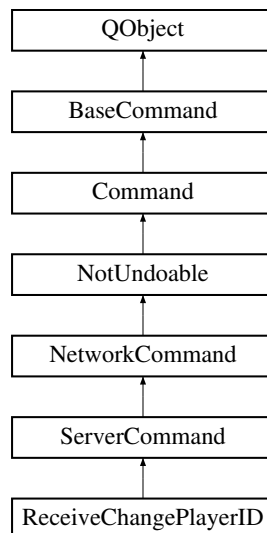
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/**Quit.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/**Quit.cpp**

8.101 ReceiveChangePlayerID Class Reference

Receive the player ID to which a player connection should change.

```
#include <ReceiveChangePlayerID.h>
```

Inheritance diagram for ReceiveChangePlayerID:



Public Member Functions

- **ReceiveChangePlayerID** (unsigned int **playerID**, QObject *parent=0)
- virtual bool **execute** ()

Executes the command.

Additional Inherited Members

8.101.1 Detailed Description

Receive the player ID to which a player connection should change.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.101.2 Constructor & Destructor Documentation

8.101.2.1 `ReceiveChangePlayerID::ReceiveChangePlayerID (unsigned int playerID, QObject * parent = 0)`

8.101.3 Member Function Documentation

8.101.3.1 `bool ReceiveChangePlayerID::execute () [virtual]`

Executes the command.

Returns

True, if successful.

Implements **ServerCommand** (p.316).

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

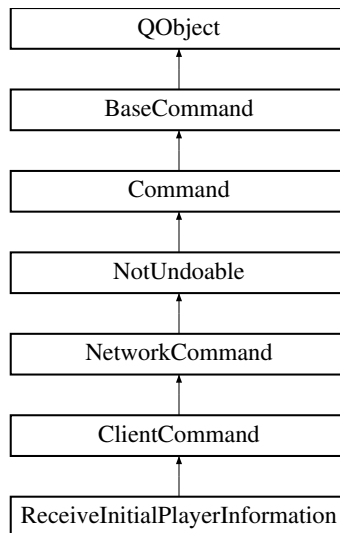
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/ReceiveChangePlayerID.h`
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/ReceiveChangePlayerID.cpp`

8.102 ReceiveInitialPlayerInformation Class Reference

Receive the initial player information.

```
#include <ReceiveInitialPlayerInformation.h>
```

Inheritance diagram for ReceiveInitialPlayerInformation:



Public Member Functions

- **ReceiveInitialPlayerInformation** (`QObject *parent=0`)
- virtual bool **execute** ()
Executes the command.

Additional Inherited Members

8.102.1 Detailed Description

Receive the initial player information.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.102.2 Constructor & Destructor Documentation

8.102.2.1 `ReceiveInitialPlayerInformation::ReceiveInitialPlayerInformation (QObject * parent = 0)`

8.102.3 Member Function Documentation

8.102.3.1 `bool ReceiveInitialPlayerInformation::execute () [virtual]`

Executes the command.

Returns

True, if successful.

Implements **ClientCommand** (p. 80).

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

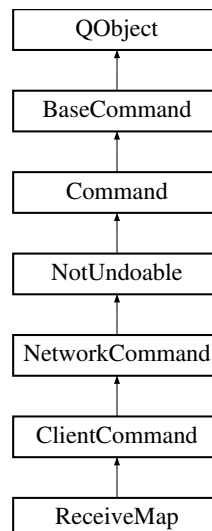
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/**ReceiveInitialPlayerInformation.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/**ReceiveInitialPlayerInformation.cpp**

8.103 ReceiveMap Class Reference

Receive and load a map from the server.

```
#include <ReceiveMap.h>
```

Inheritance diagram for ReceiveMap:

**Public Member Functions**

- **ReceiveMap** (QObject *parent=0)
- virtual bool **execute** ()
Executes the command.

Additional Inherited Members

8.103.1 Detailed Description

Receive and load a map from the server.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.103.2 Constructor & Destructor Documentation**8.103.2.1** ReceiveMap::ReceiveMap (QObject * *parent* = 0)**8.103.3 Member Function Documentation****8.103.3.1** bool ReceiveMap::execute () [virtual]

Executes the command.

Returns

True, if successful.

Implements **ClientCommand** (p. 80).

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

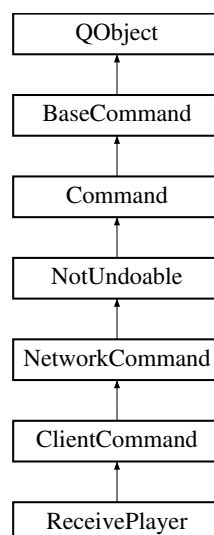
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/**ReceiveMap.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/**ReceiveMap.cpp**

8.104 ReceivePlayer Class Reference

Receive a player object from the server.

#include <ReceivePlayer.h>

Inheritance diagram for ReceivePlayer:

**Public Member Functions**

- **ReceivePlayer** (QObject *parent=0)
- virtual bool **execute** ()
Executes the command.

Additional Inherited Members

8.104.1 Detailed Description

Receive a player object from the server.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.104.2 Constructor & Destructor Documentation

8.104.2.1 ReceivePlayer::ReceivePlayer (QObject * *parent* = 0)

8.104.3 Member Function Documentation

8.104.3.1 bool ReceivePlayer::execute () [virtual]

Executes the command.

Returns

True, if successful.

Implements **ClientCommand** (p. 80).

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

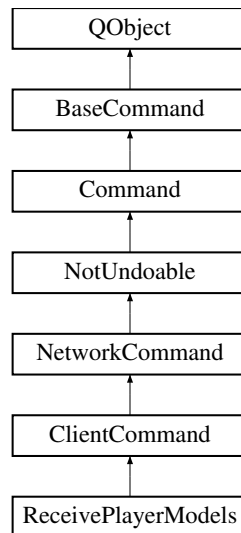
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/**ReceivePlayer.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/**ReceivePlayer.cpp**

8.105 ReceivePlayerModels Class Reference

Receive the requested player models.

```
#include <ReceivePlayerModels.h>
```

Inheritance diagram for ReceivePlayerModels:



Public Member Functions

- **ReceivePlayerModels** (`QObject *parent=0`)
- virtual bool **execute** ()
Executes the command.

Additional Inherited Members

8.105.1 Detailed Description

Receive the requested player models.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.105.2 Constructor & Destructor Documentation

8.105.2.1 `ReceivePlayerModels::ReceivePlayerModels (QObject * parent = 0)`

8.105.3 Member Function Documentation

8.105.3.1 `bool ReceivePlayerModels::execute () [virtual]`

Executes the command.

Returns

True, if successful.

Implements **ClientCommand** (p. 80).

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

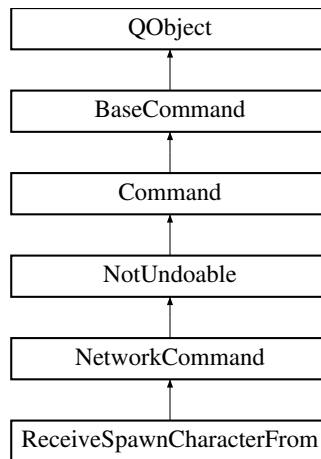
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/**ReceivePlayerModels.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/**ReceivePlayerModels.cpp**

8.106 ReceiveSpawnCharacterFrom Class Reference

Receive the character that has been spawned.

```
#include <ReceiveSpawnCharacterFrom.h>
```

Inheritance diagram for ReceiveSpawnCharacterFrom:

**Public Member Functions**

- **ReceiveSpawnCharacterFrom** (unsigned int **senderID**, QObject *parent=0)
- virtual bool **execute** ()
Executes the command.

Private Attributes

- unsigned int **senderID**
The sender ID that the character sent by.

Additional Inherited Members

8.106.1 Detailed Description

Receive the character that has been spawned.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.106.2 Constructor & Destructor Documentation**8.106.2.1** `ReceiveSpawnCharacterFrom::ReceiveSpawnCharacterFrom (unsigned int senderID, QObject * parent = 0)`**8.106.3 Member Function Documentation****8.106.3.1** `bool ReceiveSpawnCharacterFrom::execute () [virtual]`

Executes the command.

Returns

True, if successful.

Implements **NetworkCommand** (p. 202).**8.106.4 Member Data Documentation****8.106.4.1** `unsigned int ReceiveSpawnCharacterFrom::senderID [private]`

The sender ID that the character sent by.

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

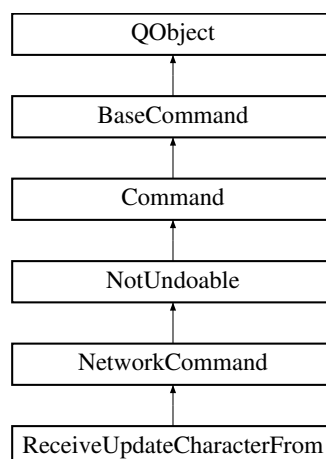
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/**ReceiveSpawnCharacterFrom.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/**ReceiveSpawnCharacterFrom.cpp**

8.107 ReceiveUpdateCharacterFrom Class Reference

Receive the Updated Character from any given sender ID.

`#include <ReceiveUpdateCharacterFrom.h>`

Inheritance diagram for ReceiveUpdateCharacterFrom:



Public Member Functions

- **ReceiveUpdateCharacterFrom** (unsigned int **senderID**, QObject *parent=0)
- virtual bool **execute** ()

Executes the command.

Private Attributes

- unsigned int **senderID**

The sender ID that the character sent by.

Additional Inherited Members

8.107.1 Detailed Description

Receive the Updated Character from any given sender ID.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.107.2 Constructor & Destructor Documentation

8.107.2.1 `ReceiveUpdateCharacterFrom::ReceiveUpdateCharacterFrom (unsigned int senderID, QObject * parent = 0)`

8.107.3 Member Function Documentation

8.107.3.1 `bool ReceiveUpdateCharacterFrom::execute () [virtual]`

Executes the command.

Returns

True, if successful.

Implements **NetworkCommand** (p. 202).

8.107.4 Member Data Documentation

8.107.4.1 `unsigned int ReceiveUpdateCharacterFrom::senderID [private]`

The sender ID that the character sent by.

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

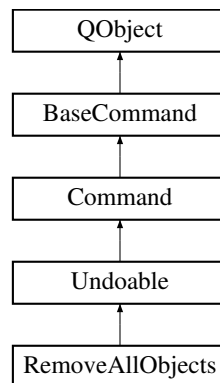
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/**ReceiveUpdateCharacterFrom.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/**ReceiveUpdateCharacterFrom.cpp**

8.108 RemoveAllObjects Class Reference

Remove all objects from a stackmodel.

```
#include <RemoveAllObjects.h>
```

Inheritance diagram for RemoveAllObjects:



Public Member Functions

- **RemoveAllObjects** (**StackModel** ***stackModel**, **QObject** ***parent**=0)
- virtual bool **execute** ()
Executes the command.
- virtual void **undo** ()
Undo changes made by the command.
- virtual void **redo** ()
Redo changes made by the undoing of the command.

Private Attributes

- **StackModel** * **stackModel**
The stack from which to remove objects.
- **ObjectCharacter** * **undoCharacter**
The character removed.
- **QList**< **ObjectItem** * > **undoItemList**
The items removed.
- **ObjectTile** * **undoWall**
The wall removed.
- **ObjectTile** * **undoFloor**
The floor removed.
- **QList**< **ObjectSpecial** * > **undoSpecialList**
The specials removed.

Additional Inherited Members

8.108.1 Detailed Description

Remove all objects from a stackmodel.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.108.2 Constructor & Destructor Documentation

8.108.2.1 RemoveAllObjects::RemoveAllObjects (**StackModel** * *stackModel*, **QObject** * *parent* = 0)

8.108.3 Member Function Documentation

8.108.3.1 bool RemoveAllObjects::execute () [virtual]

Executes the command.

Returns

True, if successful.

Implements **Undoable** (p. 354).

8.108.3.2 void RemoveAllObjects::redo () [virtual]

Redo changes made by the undoing of the command.

Implements **Undoable** (p. 355).

8.108.3.3 void RemoveAllObjects::undo () [virtual]

Undo changes made by the command.

Implements **Undoable** (p. 355).

8.108.4 Member Data Documentation

8.108.4.1 **StackModel*** RemoveAllObjects::stackModel [private]

The stack from which to remove objects.

8.108.4.2 **ObjectCharacter*** RemoveAllObjects::undoCharacter [private]

The character removed.

8.108.4.3 **ObjectTile*** RemoveAllObjects::undoFloor [private]

The floor removed.

8.108.4.4 **QList**<**ObjectItem** *> RemoveAllObjects::undoltemList [private]

The items removed.

8.108.4.5 `QList<ObjectSpecial*> RemoveAllObjects::undoSpecialList` [private]

The specials removed.

8.108.4.6 `ObjectTile* RemoveAllObjects::undoWall` [private]

The wall removed.

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

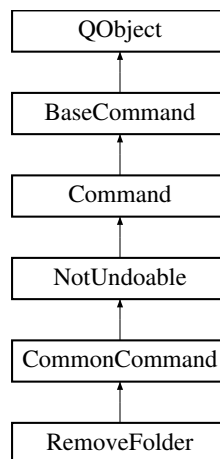
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/RemoveAllObjects.h`
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/RemoveAllObjects.cpp`

8.109 RemoveFolder Class Reference

Remove the a folder recursively.

```
#include <RemoveFolder.h>
```

Inheritance diagram for RemoveFolder:



Public Member Functions

- **RemoveFolder** (const `QFileInfo &folder`, `QObject *parent=0`)
- virtual bool **execute** ()
Executes the command.

Private Member Functions

- void **removeDirRecursive** (const `QDir &next`)
Remove directories recursive.
- void **removeEmpty** (const `QDir &target`)
Remove an empty directory.

Private Attributes

- `QFileInfo` **folder**
The folder to be recursively removed.

Additional Inherited Members

8.109.1 Detailed Description

Remove the a folder recursively.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.109.2 Constructor & Destructor Documentation

8.109.2.1 RemoveFolder::RemoveFolder (const QFileInfo & *folder*, QObject * *parent* = 0)

8.109.3 Member Function Documentation

8.109.3.1 bool RemoveFolder::execute () [virtual]

Executes the command.

Returns

True, if successful.

Implements **CommonCommand** (p. 98).

8.109.3.2 void RemoveFolder::removeDirRecursive (const QDir & *next*) [private]

Remove directories recursive.

Parameters

<i>next</i>	The next directory to go through and queck for new directories to delete.
-------------	---

8.109.3.3 void RemoveFolder::removeEmpty (const QDir & *target*) [private]

Remove an empty directory.

Parameters

<i>target</i>	The directory that should be empty.
---------------	-------------------------------------

8.109.4 Member Data Documentation

8.109.4.1 QFileInfo RemoveFolder::folder [private]

The folder to be recursively removed.

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

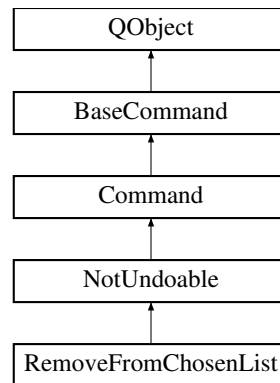
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/**RemoveFolder.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/**RemoveFolder.cpp**

8.110 RemoveFromChosenList Class Reference

Remove entry of list model.

```
#include <RemoveFromChosenList.h>
```

Inheritance diagram for RemoveFromChosenList:



Public Member Functions

- **RemoveFromChosenList** (QStandardItemModel ***chosenListModel**, const QModelIndex &**currentIndex**, QObject *parent=0)
- virtual bool **execute** ()

Executes the command.

Private Attributes

- QStandardItemModel * **chosenListModel**

The chosen list entry list.

- QModelIndex **currentIndex**

The index to be removed from the chosenListModel.

Additional Inherited Members

8.110.1 Detailed Description

Remove entry of list model.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.110.2 Constructor & Destructor Documentation

8.110.2.1 `RemoveFromChosenList::RemoveFromChosenList (QStandardItemModel * chosenListModel, const QModelIndex & currentIndex, QObject * parent = 0)`

8.110.3 Member Function Documentation

8.110.3.1 `bool RemoveFromChosenList::execute () [virtual]`

Executes the command.

Returns

True, if successful.

Implements **NotUndoable** (p. 215).

8.110.4 Member Data Documentation

8.110.4.1 `QStandardItemModel* RemoveFromChosenList::chosenListModel [private]`

The chosen list entry list.

8.110.4.2 `QModelIndex RemoveFromChosenList::currentIndex [private]`

The index to be removed from the chosenListModel.

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

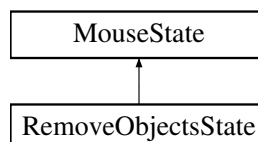
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/RemoveFromChosenList.h`
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/RemoveFromChosenList.cpp`

8.111 RemoveObjectsState Class Reference

Remove objects from stacks.

```
#include <RemoveObjectsState.h>
```

Inheritance diagram for RemoveObjectsState:



Public Member Functions

- **RemoveObjectsState (Processor *processor, StackModel *targetStack)**
- **virtual MouseState * nextState (MouseState::MouseEventType type, QMouseEvent *event, StackModel *targetStack, StackModel *fromStack=0)**

Determines which state is next, depending on the input. This method might destroy this state and changes occur.

Private Attributes

- `QSet< StackModel * > alreadyPlaced`

Additional Inherited Members

8.111.1 Detailed Description

Remove objects from stacks.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.111.2 Constructor & Destructor Documentation

8.111.2.1 `RemoveObjectsState::RemoveObjectsState (Processor * processor, StackModel * targetStack)`

8.111.3 Member Function Documentation

8.111.3.1 `MouseState * RemoveObjectsState::nextState (MouseState::MouseEventType type, QMouseEvent * event, StackModel * targetStack, StackModel * fromStack = 0) [virtual]`

Determines which state is next, depending on the input. This method might destroy this state and changes occur.

Parameters

<i>event</i>	Holds the event that has been triggered.
<i>stack</i>	The stack where this method has been invoked from.

Returns

a pointer to the next state.

Implements **MouseState** (p. 197).

8.111.4 Member Data Documentation

8.111.4.1 `QSet<StackModel *> RemoveObjectsState::alreadyPlaced [private]`

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

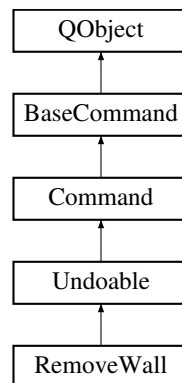
- `/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/RemoveObjectsState.h`
- `/home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/RemoveObjectsState.cpp`

8.112 RemoveWall Class Reference

Remove a wall from a chosen stack.

```
#include <RemoveWall.h>
```

Inheritance diagram for RemoveWall:



Public Member Functions

- **RemoveWall** (**StackModel** ***stackModel**, **QObject** ***parent**=0)
- virtual bool **execute** ()
Executes the command.
- virtual void **undo** ()
Undo changes made by the command.
- virtual void **redo** ()
Redo changes made by the undoing of the command.

Private Attributes

- **StackModel** * **stackModel**
The stack from which to remove the wall from.
- **ObjectTile** * **oldWall**
The wall which has been removed.

Additional Inherited Members

8.112.1 Detailed Description

Remove a wall from a chosen stack.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.112.2 Constructor & Destructor Documentation

8.112.2.1 `RemoveWall::RemoveWall (StackModel * stackModel, QObject * parent = 0)` `[explicit]`

8.112.3 Member Function Documentation

8.112.3.1 `bool RemoveWall::execute ()` `[virtual]`

Executes the command.

Returns

True, if successful.

Implements **Undoable** (p. 354).

8.112.3.2 `void RemoveWall::redo ()` `[virtual]`

Redo changes made by the undoing of the command.

Implements **Undoable** (p. 355).

8.112.3.3 `void RemoveWall::undo ()` `[virtual]`

Undo changes made by the command.

Implements **Undoable** (p. 355).

8.112.4 Member Data Documentation

8.112.4.1 `ObjectTile* RemoveWall::oldWall` `[private]`

The wall which has been removed.

8.112.4.2 `StackModel* RemoveWall::stackModel` `[private]`

The stack from which to remove the wall from.

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

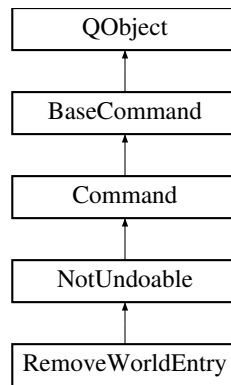
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/RemoveWall.h`
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/RemoveWall.cpp`

8.113 RemoveWorldEntry Class Reference

Remove an entry of the world.

```
#include <RemoveWorldEntry.h>
```

Inheritance diagram for RemoveWorldEntry:



Public Member Functions

- **RemoveWorldEntry** (const QModelIndex &**index**, bool **mayAsk**, QObject *parent=0)
- virtual bool **execute** ()
Executes the command.

Private Attributes

- QModelIndex **index**
Index to be removed.
- bool **mayAsk**
True, if it should be asked for unsaved changes to be saved.

Additional Inherited Members

8.113.1 Detailed Description

Remove an entry of the world.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.113.2 Constructor & Destructor Documentation

8.113.2.1 RemoveWorldEntry::RemoveWorldEntry (const QModelIndex & *index*, bool *mayAsk*, QObject * *parent* = 0)

8.113.3 Member Function Documentation

8.113.3.1 bool RemoveWorldEntry::execute () [virtual]

Executes the command.

Returns

True, if successful.

Implements **NotUndoable** (p. 215).

8.113.4 Member Data Documentation**8.113.4.1 QModelIndex RemoveWorldEntry::index** [private]

Index to be removed.

8.113.4.2 bool RemoveWorldEntry::mayAsk [private]

True, if it should be asked for unsaved changes to be saved.

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

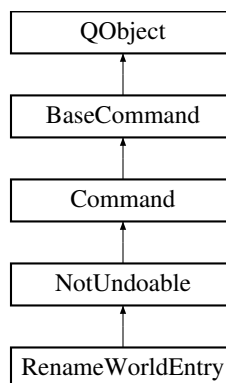
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/**RemoveWorldEntry.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/**RemoveWorldEntry.cpp**

8.114 RenameWorldEntry Class Reference

Renames an entry of the world.

```
#include <RenameWorldEntry.h>
```

Inheritance diagram for RenameWorldEntry:

**Public Member Functions**

- **RenameWorldEntry** (const QModelIndex &**index**, QObject *parent=0)
- virtual bool **execute** ()

Executes the command.

Private Attributes

- QModelIndex **index**

Index pointing to the entry to be renamed.

Additional Inherited Members

8.114.1 Detailed Description

Renames an entry of the world.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.114.2 Constructor & Destructor Documentation

8.114.2.1 `RenameWorldEntry::RenameWorldEntry (const QModelIndex & index, QObject * parent = 0)`

8.114.3 Member Function Documentation

8.114.3.1 `bool RenameWorldEntry::execute () [virtual]`

Executes the command.

Returns

True, if successful.

Implements **NotUndoable** (p. 215).

8.114.4 Member Data Documentation

8.114.4.1 `QModelIndex RenameWorldEntry::index [private]`

Index pointing to the entry to be renamed.

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

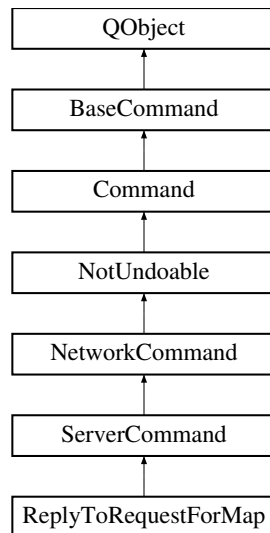
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/RenameWorldEntry.h`
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/RenameWorldEntry.cpp`

8.115 ReplyToRequestForMap Class Reference

Reply a requested map to a player.

```
#include <ReplyToRequestForMap.h>
```

Inheritance diagram for ReplyToRequestForMap:



Public Member Functions

- **ReplyToRequestForMap** (unsigned int **playerID**, `QObject *parent=0`)
- virtual bool **execute** ()
Executes the command.

Additional Inherited Members

8.115.1 Detailed Description

Reply a requested map to a player.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.115.2 Constructor & Destructor Documentation

8.115.2.1 `ReplyToRequestForMap::ReplyToRequestForMap (unsigned int playerID, QObject * parent = 0)`

8.115.3 Member Function Documentation

8.115.3.1 `bool ReplyToRequestForMap::execute ()` [virtual]

Executes the command.

Returns

True, if successful.

Implements **ServerCommand** (p. 316).

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

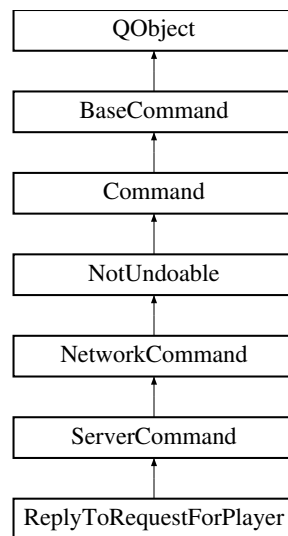
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/**ReplyToRequestForMap.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/**ReplyToRequestForMap.cpp**

8.116 ReplyToRequestForPlayer Class Reference

Reply a requested player object to a given player.

```
#include <ReplyToRequestForPlayer.h>
```

Inheritance diagram for ReplyToRequestForPlayer:

**Public Member Functions**

- **ReplyToRequestForPlayer** (unsigned int **playerID**, QObject *parent=0)
- virtual bool **execute** ()

Executes the command.

Additional Inherited Members

8.116.1 Detailed Description

Reply a requested player object to a given player.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.116.2 Constructor & Destructor Documentation

8.116.2.1 `ReplyToRequestForPlayer::ReplyToRequestForPlayer (unsigned int playerID, QObject * parent = 0)`

8.116.3 Member Function Documentation

8.116.3.1 `bool ReplyToRequestForPlayer::execute () [virtual]`

Executes the command.

Returns

True, if successful.

Implements **ServerCommand** (p. 316).

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

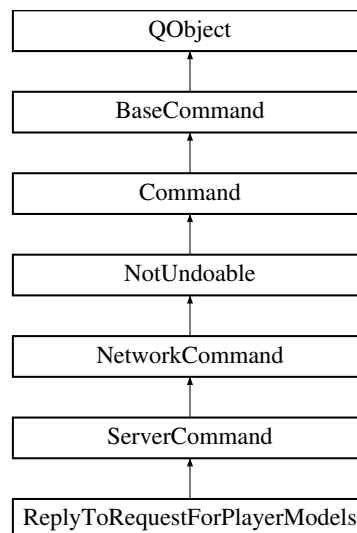
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/ReplyToRequestForPlayer.h`
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/ReplyToRequestForPlayer.cpp`

8.117 ReplyToRequestForPlayerModels Class Reference

Reply requested player models to a player.

```
#include <ReplyToRequestForPlayerModels.h>
```

Inheritance diagram for ReplyToRequestForPlayerModels:



Public Member Functions

- **ReplyToRequestForPlayerModels** (unsigned int **playerID**, QObject *parent=0)
- virtual bool **execute** ()

Executes the command.

Additional Inherited Members

8.117.1 Detailed Description

Reply requested player models to a player.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.117.2 Constructor & Destructor Documentation

8.117.2.1 ReplyToRequestForPlayerModels::ReplyToRequestForPlayerModels (unsigned int *playerID*, QObject * *parent* = 0)

8.117.3 Member Function Documentation

8.117.3.1 bool ReplyToRequestForPlayerModels::execute () [virtual]

Executes the command.

Returns

True, if successful.

Implements **ServerCommand** (p. 316).

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

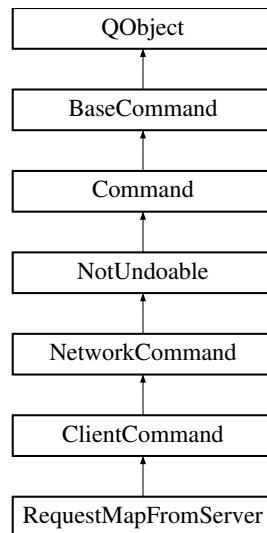
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/**ReplyToRequestForPlayerModels.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/**ReplyToRequestForPlayerModels.cpp**

8.118 RequestMapFromServer Class Reference

Request a specific map from the server.

```
#include <RequestMapFromServer.h>
```

Inheritance diagram for RequestMapFromServer:



Public Member Functions

- **RequestMapFromServer** (const QString &**localMapPath**, QObject *parent=0)
- virtual bool **execute** ()
Executes the command.

Private Attributes

- QString **localMapPath**
The local path of the map to be requested.

Additional Inherited Members

8.118.1 Detailed Description

Request a specific map from the server.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.118.2 Constructor & Destructor Documentation

8.118.2.1 `RequestMapFromServer::RequestMapFromServer (const QString & localMapPath, QObject * parent = 0)`

8.118.3 Member Function Documentation

8.118.3.1 `bool RequestMapFromServer::execute () [virtual]`

Executes the command.

Returns

True, if successful.

Implements **ClientCommand** (p. 80).

8.118.4 Member Data Documentation**8.118.4.1 QString RequestMapFromServer::localMapPath** [private]

The local path of the map to be requested.

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

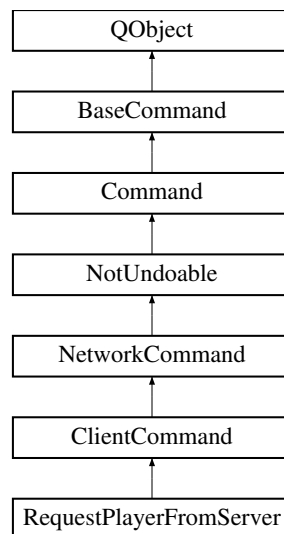
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/**RequestMapFromServer.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/**RequestMapFromServer.cpp**

8.119 RequestPlayerFromServer Class Reference

Request a specific player from the server.

```
#include <RequestPlayerFromServer.h>
```

Inheritance diagram for RequestPlayerFromServer:

**Public Member Functions**

- **RequestPlayerFromServer** (unsigned int **playerID**, QObject *parent=0)
- virtual bool **execute** ()

Executes the command.

Private Attributes

- unsigned int **playerID**

The player ID of the player to be requested.

Additional Inherited Members

8.119.1 Detailed Description

Request a specific player from the server.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.119.2 Constructor & Destructor Documentation

8.119.2.1 `RequestPlayerFromServer::RequestPlayerFromServer (unsigned int playerID, QObject * parent = 0)`

8.119.3 Member Function Documentation

8.119.3.1 `bool RequestPlayerFromServer::execute () [virtual]`

Executes the command.

Returns

True, if successful.

Implements **ClientCommand** (p. 80).

8.119.4 Member Data Documentation

8.119.4.1 `unsigned int RequestPlayerFromServer::playerID [private]`

The player ID of the player to be requested.

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

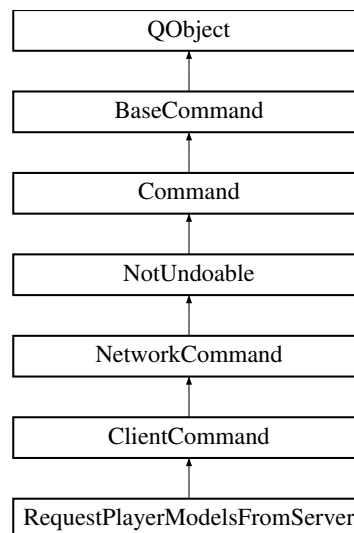
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/RequestPlayerFromServer.h`
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/RequestPlayerFromServer.cpp`

8.120 RequestPlayerModelsFromServer Class Reference

Request all player models from the server.

```
#include <RequestPlayerModelsFromServer.h>
```

Inheritance diagram for RequestPlayerModelsFromServer:



Public Member Functions

- **RequestPlayerModelsFromServer** (QObject *parent=0)
- virtual bool **execute** ()
Executes the command.

Additional Inherited Members

8.120.1 Detailed Description

Request all player models from the server.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.120.2 Constructor & Destructor Documentation

8.120.2.1 `RequestPlayerModelsFromServer::RequestPlayerModelsFromServer (QObject * parent = 0)`

8.120.3 Member Function Documentation

8.120.3.1 `bool RequestPlayerModelsFromServer::execute () [virtual]`

Executes the command.

Returns

True, if successful.

Implements **ClientCommand** (p. 80).

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

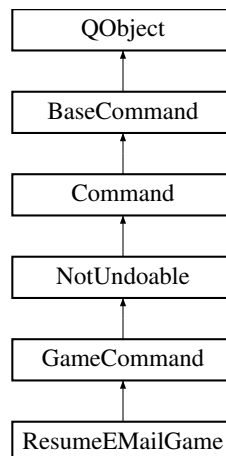
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/**RequestPlayerModelsFrom-Server.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/**RequestPlayerModelsFrom-Server.cpp**

8.121 ResumeEMailGame Class Reference

Resume an E-Mail Game.

```
#include <ResumeEMailGame.h>
```

Inheritance diagram for ResumeEMailGame:



Public Member Functions

- **ResumeEMailGame** (QObject *parent=0)
- virtual bool **execute** ()
Executes the command.

Private Attributes

- **FileManager** fileManager

Additional Inherited Members

8.121.1 Detailed Description

Resume an E-Mail Game.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.121.2 Constructor & Destructor Documentation

8.121.2.1 ResumeEMailGame::ResumeEMailGame (QObject * *parent* = 0)

8.121.3 Member Function Documentation

8.121.3.1 bool ResumeEMailGame::execute () [virtual]

Executes the command.

Returns

True, if successful.

Implements **GameCommand** (p. 133).

8.121.4 Member Data Documentation

8.121.4.1 FileManager ResumeEMailGame::fileManager [private]

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

- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/**ResumeEMailGame.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/**ResumeEMailGame.cpp**

8.122 RSAMethod Class Reference

The RSA Method implemented wrapped around the Crypto++ library. This is written to be completely independent of QT.

```
#include <RSAMethod.h>
```

Public Member Functions

- **RSAMethod** (const string &**privateKeyFilePath**, const string &**publicKeyFilePath**)
Loads a key pair from two different files.
- **RSAMethod** (unsigned int keySize, const string &**privateKeyFilePath**, const string &**publicKeyFilePath**)
Generates a key pair and saves them sperately in two different files.
- string **encryptString** (const string &plain)
Encrypt a given string with the loaded keys.
- string **decryptString** (const string &cipher)
Decrypt a given string with the loaded keys.
- size_t **getMaxCipherTextLength** ()
- size_t **getMaxPlainTextLength** ()

Private Member Functions

- void **init** ()
- void **generateKeys** (unsigned int keySize)
Generates and loads the private and public keys.
- void **loadKeys** ()
Load keys from the given file paths.
- void **load** (const string &filename, BufferedTransformation &buffer)
- template<class T >
void **loadKey** (const string &filename, T &key)
- void **save** (const string &filename, const BufferedTransformation &buffer)
- template<class T >
void **saveKey** (const string &filename, const T &key)

Private Attributes

- AutoSeededRandomPool **randomNumberGenerator**
- RSA::PrivateKey **privateKey**
- string **privateKeyFilePath**
- size_t **maxCipherTextLength**
- RSA::PublicKey **publicKey**
- string **publicKeyFilePath**
- size_t **maxPlainTextLength**

8.122.1 Detailed Description

The RSA Method implemented wrapped around the Crypto++ library. This is written to be completely independent of QT.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.122.2 Constructor & Destructor Documentation

8.122.2.1 RSAMethod::RSAMethod (const string & *privateKeyFilePath*, const string & *publicKeyFilePath*)

Loads a key pair from two different files.

Parameters

<i>keyDirectory</i>	Where to put the keys.
---------------------	------------------------

8.122.2.2 RSAMethod::RSAMethod (unsigned int *keySize*, const string & *privateKeyFilePath*, const string & *publicKeyFilePath*)

Generates a key pair and saves them sperately in two different files.

Parameters

<i>keySize</i>	The RSA key size in bits.
<i>keyDirectory</i>	Where to get the keys from.

8.122.3 Member Function Documentation

8.122.3.1 string RSAMethod::decryptString (const string & *cipher*)

Decrypt a given string with the loaded keys.

Parameters

<i>cipherText</i>	The ciphertext to be decrypted.
-------------------	---------------------------------

Returns

The recovered plaintext.

8.122.3.2 string RSAMethod::encryptString (const string & *plain*)

Encrypt a given string with the loaded keys.

Parameters

<i>plainText</i>	The plaintext to be encrypted.
------------------	--------------------------------

Returns

The ciphertext.

8.122.3.3 void RSAMethod::generateKeys (unsigned int *keySize*) [private]

Generates and loads the private and public keys.

Parameters

<i>keySize</i>	The size of the key in bits. 1024 Bit Key is currently good enough.
----------------	---

8.122.3.4 size_t RSAMethod::getMaxCipherTextLength ()

8.122.3.5 size_t RSAMethod::getMaxPlainTextLength ()

8.122.3.6 void RSAMethod::init () [private]

8.122.3.7 void RSAMethod::load (const string & *filename*, BufferedTransformation & *buffer*) [private]8.122.3.8 template<class T> void RSAMethod::loadKey (const string & *filename*, T & *key*) [inline], [private]

8.122.3.9 void RSAMethod::loadKeys () [private]

Load keys from the given file paths.

8.122.3.10 `void RSAMethod::save (const string & filename, const BufferedTransformation & buffer)` [private]

8.122.3.11 `template<class T > void RSAMethod::saveKey (const string & filename, const T & key)` [inline], [private]

8.122.4 Member Data Documentation

8.122.4.1 `size_t RSAMethod::maxCipherTextLength` [private]

8.122.4.2 `size_t RSAMethod::maxPlainTextLength` [private]

8.122.4.3 `RSA::PrivateKey RSAMethod::privateKey` [private]

8.122.4.4 `string RSAMethod::privateKeyFilePath` [private]

8.122.4.5 `RSA::PublicKey RSAMethod::publicKey` [private]

8.122.4.6 `string RSAMethod::publicKeyFilePath` [private]

8.122.4.7 `AutoSeededRandomPool RSAMethod::randomNumberGenerator` [private]

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

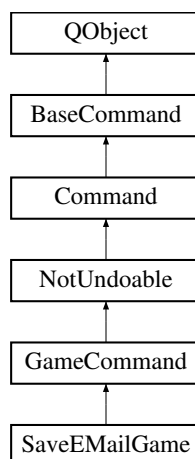
- `/home/ryoga/git/pidgirl/pidgirl-engine/RSA/RSAMethod.h`
- `/home/ryoga/git/pidgirl/pidgirl-engine/RSA/RSAMethod.cpp`

8.123 SaveEmailGame Class Reference

Save the current EMail Game.

```
#include <SaveEmailGame.h>
```

Inheritance diagram for SaveEmailGame:



Public Member Functions

- **SaveEmailGame** (**MailGameInfo** ***mailGameInfo**, const **QString** &**playerName**, unsigned int **turnCount**, **QObject** *parent=0)
- virtual bool **execute** ()
Executes the command.

Private Attributes

- **FileManager** `fileManager`
- **MailGameInfo** * `mailGameInfo`
- **QString** `playerName`
- unsigned int `turnCount`

Additional Inherited Members

8.123.1 Detailed Description

Save the current EMail Game.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.123.2 Constructor & Destructor Documentation

8.123.2.1 **SaveEMailGame::SaveEMailGame** (**MailGameInfo** * *mailGameInfo*, const **QString** & *playerName*, unsigned int *turnCount*, **QObject** * *parent* = 0)

8.123.3 Member Function Documentation

8.123.3.1 **bool** **SaveEMailGame::execute** () [virtual]

Executes the command.

Returns

True, if successful.

Implements **GameCommand** (p. 133).

8.123.4 Member Data Documentation

8.123.4.1 **FileManager** **SaveEMailGame::fileManager** [private]

8.123.4.2 **MailGameInfo*** **SaveEMailGame::mailGameInfo** [private]

8.123.4.3 **QString** **SaveEMailGame::playerName** [private]

8.123.4.4 unsigned int **SaveEMailGame::turnCount** [private]

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

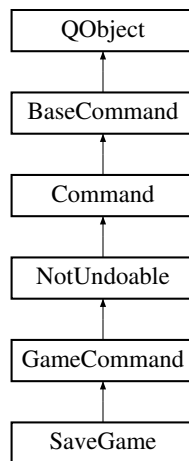
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/**SaveEMailGame.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/**SaveEMailGame.cpp**

8.124 SaveGame Class Reference

Save the current game and world models.

```
#include <SaveGame.h>
```

Inheritance diagram for SaveGame:



Public Member Functions

- **SaveGame** (QObject *parent=0)
- virtual bool **execute** ()
Executes the command.

Private Attributes

- **FileManager** fileManager

Additional Inherited Members

8.124.1 Detailed Description

Save the current game and world models.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.124.2 Constructor & Destructor Documentation

8.124.2.1 SaveGame::SaveGame (QObject * parent = 0)

8.124.3 Member Function Documentation

8.124.3.1 bool SaveGame::execute () [virtual]

Executes the command.

Returns

True, if successful.

Implements **GameCommand** (p. 133).

8.124.4 Member Data Documentation

8.124.4.1 FileManager SaveGame::fileManager [private]

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

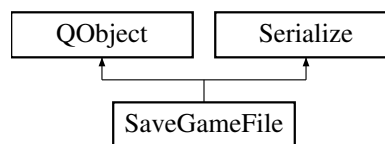
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/**SaveGame.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/**SaveGame.cpp**

8.125 SaveGameFile Class Reference

Represents everything needed to **Serialize** (p. 313) and Deserialize a complete game save file.

```
#include <SaveGameFile.h>
```

Inheritance diagram for SaveGameFile:



Public Member Functions

- **SaveGameFile** (**Processor** *processor, **WorldModel** *worldModel, **GameModel** *gameModel, QObject *parent=0)
- virtual void **serialize** (QIODevice *device) const
- virtual void **deserialize** (QIODevice *device)

Private Member Functions

- **GameModel** * **fromGameType** (**PIDGIRL::GameType** type)
*Creates a **GameModel** (p. 133) with the correct gametype.*

Private Attributes

- **Processor** * processor
- **WorldModel** * worldModel
- **GameModel** * gameModel

8.125.1 Detailed Description

Represents everything needed to **Serialize** (p. 313) and Deserialize a complete game save file.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.125.2 Constructor & Destructor Documentation

8.125.2.1 **SaveGameFile::SaveGameFile** (**Processor** * *processor*, **WorldModel** * *worldModel*, **GameModel** * *gameModel*, **QObject** * *parent* = 0)

8.125.3 Member Function Documentation

8.125.3.1 **void SaveGameFile::deserialize** (**QIODevice** * *device*) [virtual]

Implements **Serialize** (p. 314).

8.125.3.2 **GameModel** * **SaveGameFile::fromGameType** (**PIDGIRL::GameType** *type*) [private]

Creates a **GameModel** (p. 133) with the correct gametype.

Parameters

<i>type</i>	The game type.
-------------	----------------

Returns

A certain game model.

8.125.3.3 **void SaveGameFile::serialize** (**QIODevice** * *device*) const [virtual]

Implements **Serialize** (p. 314).

8.125.4 Member Data Documentation

8.125.4.1 **GameModel*** **SaveGameFile::gameModel** [private]

8.125.4.2 **Processor*** **SaveGameFile::processor** [private]

8.125.4.3 **WorldModel*** **SaveGameFile::worldModel** [private]

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

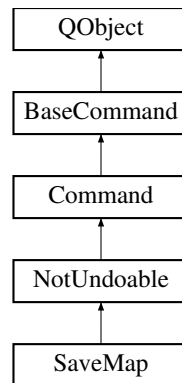
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/**SaveGameFile.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/**SaveGameFile.cpp**

8.126 SaveMap Class Reference

Save a map. If it wasn't saved before, choose a place and file.

```
#include <SaveMap.h>
```

Inheritance diagram for SaveMap:



Public Member Functions

- **SaveMap** (QObject *parent=0)
- virtual bool **execute** ()

Executes the command.

Private Attributes

- **FileManager** **fileManager**

The file manager to save the map with.

Additional Inherited Members

8.126.1 Detailed Description

Save a map. If it wasn't saved before, choose a place and file.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.126.2 Constructor & Destructor Documentation

8.126.2.1 SaveMap::SaveMap (QObject * parent = 0)

8.126.3 Member Function Documentation

8.126.3.1 `bool SaveMap::execute ()` [virtual]

Executes the command.

Returns

True, if successful.

Implements **NotUndoable** (p. 215).

8.126.4 Member Data Documentation

8.126.4.1 `FileManager SaveMap::fileManager` [private]

The file manager to save the map with.

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

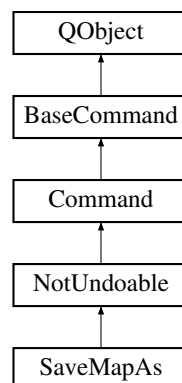
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/SaveMap.h`
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/SaveMap.cpp`

8.127 SaveMapAs Class Reference

Save a map as a speicified file.

```
#include <SaveMapAs.h>
```

Inheritance diagram for SaveMapAs:



Public Member Functions

- **SaveMapAs** (QObject *parent=0)
- virtual bool **execute** ()

Executes the command.

Private Attributes

- **FileManager fileManager**

The file manager to save the map with.

Additional Inherited Members

8.127.1 Detailed Description

Save a map as a specified file.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.127.2 Constructor & Destructor Documentation

8.127.2.1 `SaveMapAs::SaveMapAs (QObject * parent = 0)`

8.127.3 Member Function Documentation

8.127.3.1 `bool SaveMapAs::execute () [virtual]`

Executes the command.

Returns

True, if successful.

Implements **NotUndoable** (p. 215).

8.127.4 Member Data Documentation

8.127.4.1 `FileManager SaveMapAs::fileManager [private]`

The file manager to save the map with.

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

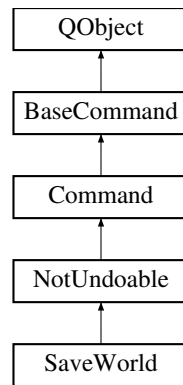
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/SaveMapAs.h`
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/SaveMapAs.cpp`

8.128 SaveWorld Class Reference

Save the world. The world might not exist, so it's simply created anew.

```
#include <SaveWorld.h>
```

Inheritance diagram for SaveWorld:



Public Member Functions

- **SaveWorld** (QObject *parent=0)
- bool **execute** ()
Executes the command.

Private Attributes

- **FileManager** fileManager
The file manager to save the world with.

Additional Inherited Members

8.128.1 Detailed Description

Save the world. The world might not exist, so it's simply created anew.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.128.2 Constructor & Destructor Documentation

8.128.2.1 **SaveWorld::SaveWorld** (QObject * *parent* = 0)

8.128.3 Member Function Documentation

8.128.3.1 bool **SaveWorld::execute** () [virtual]

Executes the command.

Returns

True, if successful.

Implements **NotUndoable** (p. 215).

8.128.4 Member Data Documentation

8.128.4.1 FileManager SaveWorld::fileManager [private]

The file manager to save the world with.

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

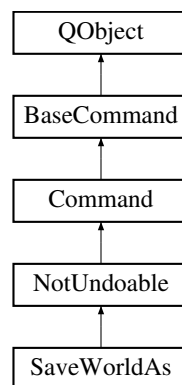
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/**SaveWorld.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/**SaveWorld.cpp**

8.129 SaveWorldAs Class Reference

Save the current world as another world.

```
#include <SaveWorldAs.h>
```

Inheritance diagram for SaveWorldAs:



Public Member Functions

- **SaveWorldAs** (QObject *parent=0)
- bool **execute** ()

Executes the command.

Private Attributes

- **FileManager** fileManager

The file manager to save the world with.

Additional Inherited Members

8.129.1 Detailed Description

Save the current world as another world.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.129.2 Constructor & Destructor Documentation

8.129.2.1 SaveWorldAs::SaveWorldAs (QObject * *parent* = 0)

8.129.3 Member Function Documentation

8.129.3.1 bool SaveWorldAs::execute () [virtual]

Executes the command.

Returns

True, if successful.

Implements **NotUndoable** (p. 215).

8.129.4 Member Data Documentation

8.129.4.1 FileManager SaveWorldAs::fileManager [private]

The file manager to save the world with.

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

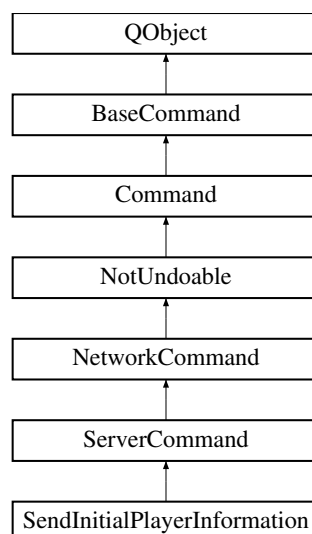
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/**SaveWorldAs.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/**SaveWorldAs.cpp**

8.130 SendInitialPlayerInformation Class Reference

Send the initial player informations to the player that connected.

#include <SendInitialPlayerInformation.h>

Inheritance diagram for SendInitialPlayerInformation:



Public Member Functions

- **SendInitialPlayerInformation** (unsigned int **playerID**, QObject *parent=0)
- virtual bool **execute** ()

Executes the command.

Additional Inherited Members

8.130.1 Detailed Description

Send the initial player informations to the player that connected.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.130.2 Constructor & Destructor Documentation

8.130.2.1 **SendInitialPlayerInformation::SendInitialPlayerInformation** (unsigned int *playerID*, QObject * *parent* = 0)

8.130.3 Member Function Documentation

8.130.3.1 **bool SendInitialPlayerInformation::execute** () [virtual]

Executes the command.

Returns

True, if successful.

Implements **ServerCommand** (p.316).

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

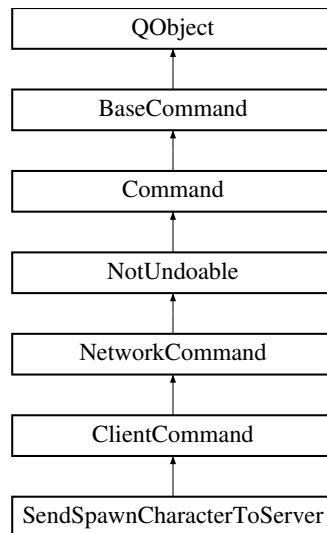
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/**SendInitialPlayerInformation.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/**SendInitialPlayerInformation.-cpp**

8.131 SendSpawnCharacterToServer Class Reference

Send the character that has been created to the server.

```
#include <SendSpawnCharacterToServer.h>
```

Inheritance diagram for SendSpawnCharacterToServer:



Public Member Functions

- **SendSpawnCharacterToServer** (const **ObjectCharacter** &**newCharacter**, `QObject *parent=0`)
- virtual bool **execute** ()
Executes the command.

Private Attributes

- const **ObjectCharacter** & **newCharacter**
The character that will be spawned.

Additional Inherited Members

8.131.1 Detailed Description

Send the character that has been created to the server.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.131.2 Constructor & Destructor Documentation

- 8.131.2.1 **SendSpawnCharacterToServer::SendSpawnCharacterToServer** (const **ObjectCharacter** & *newCharacter*, `QObject *parent = 0`)

8.131.3 Member Function Documentation

- 8.131.3.1 bool **SendSpawnCharacterToServer::execute** () [virtual]

Executes the command.

Returns

True, if successful.

Implements **ClientCommand** (p. 80).

8.131.4 Member Data Documentation**8.131.4.1** `const ObjectCharacter& SendSpawnCharacterToServer::newCharacter` [private]

The character that will be spawned.

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

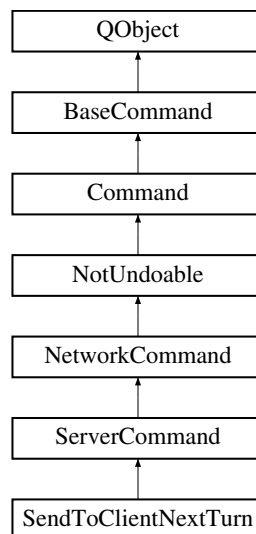
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/**SendSpawnCharacterToServer.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/**SendSpawnCharacterToServer.cpp**

8.132 SendToClientNextTurn Class Reference

Send to the next client the next turn to start the new turn for that player.

```
#include <SendToClientNextTurn.h>
```

Inheritance diagram for SendToClientNextTurn:

**Public Member Functions**

- **SendToClientNextTurn** (unsigned int **playerID**, QObject *parent=0)
- virtual bool **execute** ()
Executes the command.

Additional Inherited Members**8.132.1 Detailed Description**

Send to the next client the next turn to start the new turn for that player.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.132.2 Constructor & Destructor Documentation**8.132.2.1** SendToClientNextTurn::SendToClientNextTurn (unsigned int *playerID*, QObject * *parent* = 0)**8.132.3 Member Function Documentation****8.132.3.1** bool SendToClientNextTurn::execute () [virtual]

Executes the command.

Returns

True, if successful.

Implements **ServerCommand** (p. 316).

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

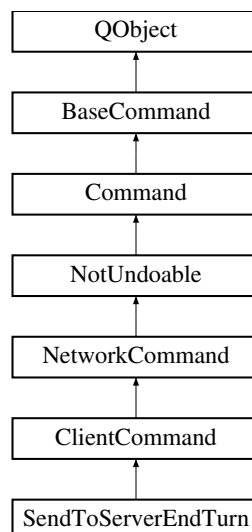
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/**SendToClientNextTurn.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/**SendToClientNextTurn.cpp**

8.133 SendToServerEndTurn Class Reference

Send End Turn to the server.

#include <SendToServerEndTurn.h>

Inheritance diagram for SendToServerEndTurn:



Public Member Functions

- **SendToServerEndTurn** (QObject *parent=0)
- virtual bool **execute** ()

Executes the command.

Additional Inherited Members

8.133.1 Detailed Description

Send End Turn to the server.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.133.2 Constructor & Destructor Documentation

8.133.2.1 **SendToServerEndTurn::SendToServerEndTurn** (QObject * *parent* = 0)

8.133.3 Member Function Documentation

8.133.3.1 bool **SendToServerEndTurn::execute** () [virtual]

Executes the command.

Returns

True, if successful.

Implements **ClientCommand** (p. 80).

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

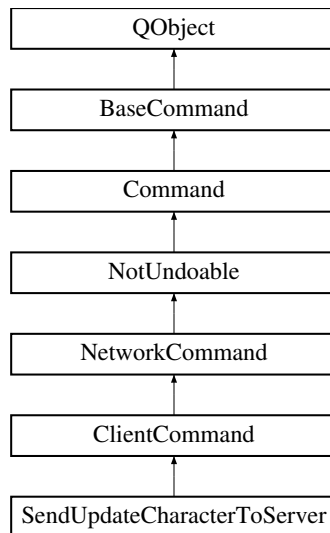
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/**SendToServerEndTurn.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/**SendToServerEndTurn.cpp**

8.134 SendUpdateCharacterToServer Class Reference

Send updated information about a character to the server.

```
#include <SendUpdateCharacterToServer.h>
```

Inheritance diagram for SendUpdateCharacterToServer:



Public Member Functions

- **SendUpdateCharacterToServer** (const **ObjectCharacter** &oldCharacter, const **ObjectCharacter** &newCharacter, QObject *parent=0)
- virtual bool **execute** ()
Executes the command.

Private Attributes

- const **ObjectCharacter** & **oldCharacter**
The old character that will be updated.
- const **ObjectCharacter** & **newCharacter**
The new character which has the informations to replace the old character.

Additional Inherited Members

8.134.1 Detailed Description

Send updated information about a character to the server.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.134.2 Constructor & Destructor Documentation

- 8.134.2.1 **SendUpdateCharacterToServer::SendUpdateCharacterToServer** (const **ObjectCharacter** & *oldCharacter*, const **ObjectCharacter** & *newCharacter*, QObject * *parent* = 0)

8.134.3 Member Function Documentation

8.134.3.1 `bool SendUpdateCharacterToServer::execute () [virtual]`

Executes the command.

Returns

True, if successful.

Implements **ClientCommand** (p. 80).

8.134.4 Member Data Documentation

8.134.4.1 `const ObjectCharacter& SendUpdateCharacterToServer::newCharacter [private]`

The new character which has the informations to replace the old character.

8.134.4.2 `const ObjectCharacter& SendUpdateCharacterToServer::oldCharacter [private]`

The old character that will be updated.

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

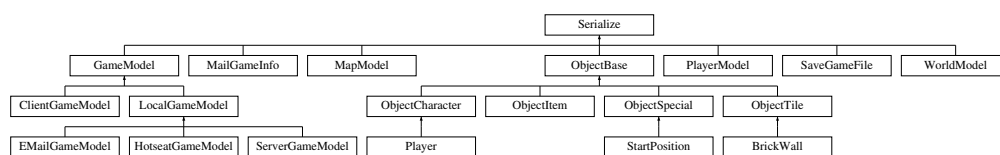
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/SendUpdateCharacterToServer.h`
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/SendUpdateCharacterToServer.cpp`

8.135 Serialize Class Reference

Interface for serializable classes.

```
#include <Serialize.h>
```

Inheritance diagram for Serialize:



Public Member Functions

- **Serialize** ()
- virtual void **serialize** (QIODevice *device) const =0
- virtual void **deserialize** (QIODevice *device)=0
- void **resetFileInfo** ()
- QFileInfo **getFileInfo** () const
- void **setFileInfo** (const QFileInfo &fileInfo)
- void **setFileInfo** (const QString &fileInfo)

Private Attributes

- **QFileInfo fileInfo**

The file information for serializable classes.

8.135.1 Detailed Description

Interface for serializable classes.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.135.2 Constructor & Destructor Documentation

8.135.2.1 Serialize::Serialize ()

8.135.3 Member Function Documentation

8.135.3.1 virtual void Serialize::deserialize (QIODevice * *device*) [pure virtual]

Implemented in **MapModel** (p. 178), **ObjectBase** (p. 220), **ClientGameModel** (p. 85), **MailGameInfo** (p. 163), **ObjectTile** (p. 239), **ServerGameModel** (p. 318), **ObjectSpecial** (p. 236), **WorldModel** (p. 367), **PlayerModel** (p. 253), **EMailGameModel** (p. 121), **ObjectCharacter** (p. 225), **ObjectItem** (p. 232), **HotseatGameModel** (p. 145), **Player** (p. 251), **BrickWall** (p. 67), **StartPosition** (p. 349), and **SaveGameFile** (p. 300).

8.135.3.2 QFileInfo Serialize::getFileInfo () const

8.135.3.3 void Serialize::resetFileInfo ()

8.135.3.4 virtual void Serialize::serialize (QIODevice * *device*) const [pure virtual]

Implemented in **MapModel** (p. 182), **ObjectBase** (p. 222), **ClientGameModel** (p. 86), **MailGameInfo** (p. 164), **ObjectTile** (p. 239), **ServerGameModel** (p. 319), **ObjectSpecial** (p. 236), **WorldModel** (p. 367), **PlayerModel** (p. 253), **EMailGameModel** (p. 122), **ObjectCharacter** (p. 226), **ObjectItem** (p. 233), **HotseatGameModel** (p. 145), **Player** (p. 252), **BrickWall** (p. 68), **StartPosition** (p. 349), and **SaveGameFile** (p. 300).

8.135.3.5 void Serialize::setFileInfo (const QFileInfo & *fileInfo*)

8.135.3.6 void Serialize::setFileInfo (const QString & *fileInfo*)

8.135.4 Member Data Documentation

8.135.4.1 QFileInfo Serialize::fileInfo [private]

The file information for serializable classes.

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

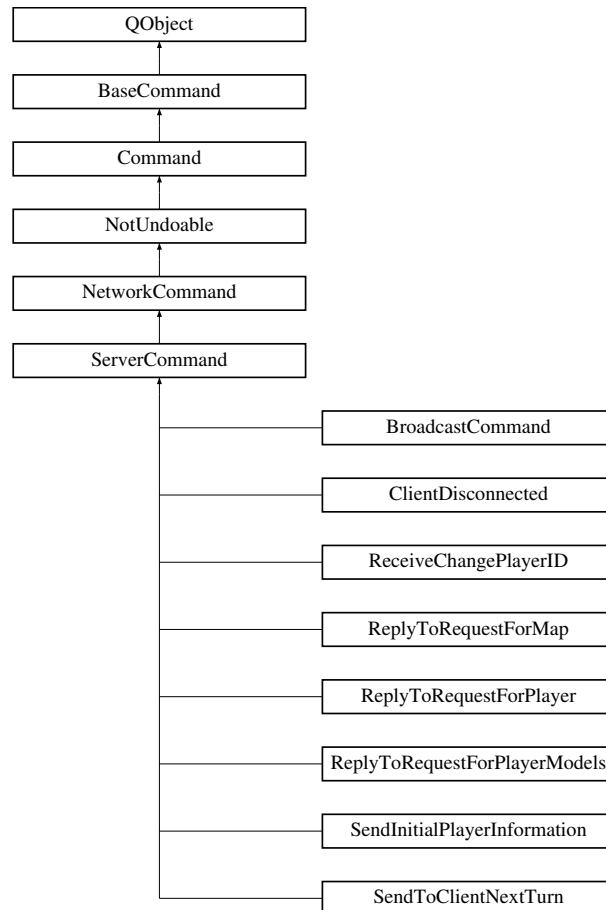
- /home/ryoga/git/pidgirl/pidgirl-engine/Common/**Serialize.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Common/**Serialize.cpp**

8.136 ServerCommand Class Reference

Command (p. 93) executed by a server.

```
#include <ServerCommand.h>
```

Inheritance diagram for ServerCommand:



Public Member Functions

- **ServerCommand** (unsigned int **playerID**, QObject *parent=0)
- virtual bool **isExecutable** () const
Check if the command is executable.
- virtual bool **execute** ()=0
Executes the command.

Protected Attributes

- unsigned int **playerID**
The player ID that this command is either meant for or came from.

8.136.1 Detailed Description

Command (p. 93) executed by a server.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.136.2 Constructor & Destructor Documentation

8.136.2.1 `ServerCommand::ServerCommand (unsigned int playerID, QObject * parent = 0)`

8.136.3 Member Function Documentation

8.136.3.1 `virtual bool ServerCommand::execute () [pure virtual]`

Executes the command.

Returns

True, if successful.

Implements **NetworkCommand** (p. 202).

Implemented in **BroadcastCommand** (p. 69), **BroadcastUpdateCharacter** (p. 73), **BroadcastSpawnCharacter** (p. 71), **ClientDisconnected** (p. 81), **ReceiveChangePlayerID** (p. 263), **ReplyToRequestForMap** (p. 284), **ReplyToRequestForPlayer** (p. 286), **ReplyToRequestForPlayerModels** (p. 287), **SendInitialPlayerInformation** (p. 307), and **SendToClientNextTurn** (p. 310).

8.136.3.2 `bool ServerCommand::isExecutable () const [virtual]`

Check if the command is executable.

Returns

If the command is executable.

Reimplemented from **NetworkCommand** (p. 203).

8.136.4 Member Data Documentation

8.136.4.1 `unsigned int ServerCommand::playerID [protected]`

The player ID that this command is either meant for or came from.

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

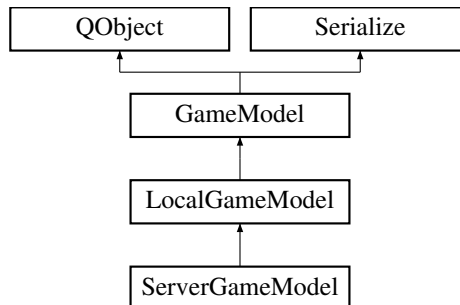
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/**ServerCommand.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/**ServerCommand.cpp**

8.137 ServerGameModel Class Reference

The model representing a game that is also a server.

```
#include <ServerGameModel.h>
```

Inheritance diagram for ServerGameModel:



Public Member Functions

- **ServerGameModel** (**Processor** *processor, QObject *parent)
- virtual bool **nextTurn** ()
Prepare for the next turn.
- virtual bool **beginTurn** ()
Begin the turn.
- virtual bool **firstTurn** ()
Prepare for the first turn.
- virtual bool **endTurn** ()
End your turn.
- virtual bool **spawnCharacter** (unsigned int **playerID**, **MapModel** *map, const **ObjectCharacter** &new-Character)
Spawn a given character on a given map.
- virtual bool **updateCharacter** (unsigned int **playerID**, **MapModel** *map, const **ObjectCharacter** &old-Character, const **ObjectCharacter** &newCharacter)
Update a given character with the information of a new character on a given map.
- virtual void **deactivatePlayer** (unsigned int **playerID**)
Deactivate a player.
- virtual void **convertFromDifferentGameModel** (**GameModel** *gameModel)
Update this game model with a game model of different type.
- virtual bool **isYourTurn** () const
- virtual **PIDGIRL::GameType** **getGameType** () const
- virtual int **getCurrentPlayerID** () const
- virtual **MapModel** * **refCurrentMap** () const
- virtual void **serialize** (QIODevice *device) const
- virtual void **deserialize** (QIODevice *device)

Private Member Functions

- void **nextCurrentPlayer** ()
- void **setCurrentPlayerID** (unsigned int **currentPlayerID**)

Private Attributes

- unsigned int **currentPlayerID**

Additional Inherited Members

8.137.1 Detailed Description

The model representing a game that is also a server.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.137.2 Constructor & Destructor Documentation

8.137.2.1 **ServerGameModel::ServerGameModel** (**Processor** * *processor*, **QObject** * *parent*)

8.137.3 Member Function Documentation

8.137.3.1 **bool** **ServerGameModel::beginTurn** () [virtual]

Begin the turn.

Implements **GameModel** (p. 137).

8.137.3.2 **void** **ServerGameModel::convertFromDifferentGameModel** (**GameModel** * *gameModel*) [virtual]

Update this game model with a game model of different type.

Parameters

<i>gameModel</i>	Game Model with a different type than the one being updated.
------------------	--

Reimplemented from **LocalGameModel** (p. 160).

8.137.3.3 **void** **ServerGameModel::deactivatePlayer** (unsigned int *playerID*) [virtual]

Deactivate a player.

Parameters

<i>playerID</i>	The playerID of which to deactivate the character of.
-----------------	---

Reimplemented from **LocalGameModel** (p. 160).

8.137.3.4 **void** **ServerGameModel::deserialize** (**QIODevice** * *device*) [virtual]

Implements **Serialize** (p. 314).

8.137.3.5 `bool ServerGameModel::endTurn () [virtual]`

End your turn.

Implements **GameModel** (p. 138).

8.137.3.6 `bool ServerGameModel::firstTurn () [virtual]`

Prepare for the first turn.

Implements **GameModel** (p. 138).

8.137.3.7 `int ServerGameModel::getCurrentPlayerID () const [virtual]`

Reimplemented from **GameModel** (p. 138).

8.137.3.8 `PIDGIRL::GameType ServerGameModel::getGameType () const [virtual]`

Implements **GameModel** (p. 138).

8.137.3.9 `bool ServerGameModel::isYourTurn () const [virtual]`

Reimplemented from **LocalGameModel** (p. 161).

8.137.3.10 `void ServerGameModel::nextCurrentPlayer () [private]`

8.137.3.11 `bool ServerGameModel::nextTurn () [virtual]`

Prepare for the next turn.

Implements **GameModel** (p. 140).

8.137.3.12 `MapModel * ServerGameModel::refCurrentMap () const [virtual]`

Reimplemented from **LocalGameModel** (p. 161).

8.137.3.13 `void ServerGameModel::serialize (QIODevice * device) const [virtual]`

Implements **Serialize** (p. 314).

8.137.3.14 `void ServerGameModel::setCurrentPlayerID (unsigned int currentPlayerID) [private]`

8.137.3.15 `bool ServerGameModel::spawnCharacter (unsigned int playerId, MapModel * map, const ObjectCharacter & newCharacter) [virtual]`

Spawn a given character on a given map.

Parameters

<i>playerID</i>	The playerId of the player that wants to spawn a character.
-----------------	---

<i>map</i>	The map where to spawn the character on.
<i>newCharacter</i>	The character to be spawned with given coordinates already set.

Returns

True, if the character has been spawned. Otherwise, false.

Reimplemented from **GameModel** (p. 141).

8.137.3.16 `bool ServerGameModel::updateCharacter (unsigned int playerID, MapModel * map, const ObjectCharacter & oldCharacter, const ObjectCharacter & newCharacter) [virtual]`

Update a given character with the information of a new character on a given myp.

Parameters

<i>playerID</i>	The playerID of the player that wants to update a character.
<i>map</i>	The map where to update the character.
<i>oldCharacter</i>	The old character that will be updated.
<i>newCharacter</i>	The new character with the updated information.

Returns

True, if the character has been updated. Otherwise, false.

Reimplemented from **GameModel** (p. 141).

8.137.4 Member Data Documentation

8.137.4.1 `unsigned int ServerGameModel::currentPlayerID [private]`

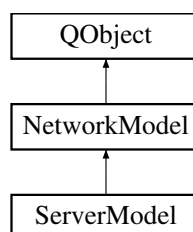
The documentation for this class was generated from the following files:

- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/**ServerGameModel.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/**ServerGameModel.cpp**

8.138 ServerModel Class Reference

```
#include <ServerModel.h>
```

Inheritance diagram for ServerModel:

**Public Member Functions**

- **ServerModel** (int **port**, WorldModel ***worldModel**, Processor ***processor**)

- virtual **~ServerModel** ()
- virtual bool **isClient** ()
- virtual bool **isServer** ()
- virtual QVector< QString > **getConnectionList** ()
- virtual QVector< unsigned int > **getConnectedPlayerIDs** ()
- virtual bool **changeClientID** (unsigned int fromID, unsigned int toID)
Change a client ID.
- bool **listen** ()
- virtual void **sendData** (unsigned int data, unsigned int playerId=0)
Sends the given data to a playerId.
- virtual void **sendData** (int data, unsigned int playerId=0)
- virtual void **sendData** (bool data, unsigned int playerId=0)
- virtual void **sendData** (const **Serialize** &data, unsigned int playerId=0)
- virtual void **sendData** (const QString &data, unsigned int playerId=0)
- virtual void **receiveData** (unsigned int &data, unsigned int playerId=0)
Receive data from a client with the given playerId.
- virtual void **receiveData** (int &data, unsigned int playerId=0)
- virtual void **receiveData** (bool &data, unsigned int playerId=0)
- virtual void **receiveData** (**Serialize** &data, unsigned int playerId=0)
- virtual void **receiveData** (QString &data, unsigned int playerId=0)

Private Slots

- void **receivePendingData** (**ClientSocket** *client)
- void **newConnection** ()
- void **deleteDisconnectedClient** (**ClientSocket** *client)
- void **socketError** (QAbstractSocket::SocketError socketError)

Private Attributes

- **Processor** * **processor**
- **WorldModel** * **worldModel**
- QTcpServer **server**
The Server connecting to all the clients.
- QMap< unsigned int, **ClientSocket** * > **clients**
A map with all clients. The key is the player's ID.

Additional Inherited Members

8.138.1 Constructor & Destructor Documentation

8.138.1.1 **ServerModel::ServerModel** (int *port*, **WorldModel** * *worldModel*, **Processor** * *processor*)

8.138.1.2 **ServerModel::~ServerModel** () [virtual]

8.138.2 Member Function Documentation

8.138.2.1 bool **ServerModel::changeClientID** (unsigned int *fromID*, unsigned int *toID*) [virtual]

Change a client ID.

Parameters

<i>fromID</i>	The client connection ID from which to change.
<i>toID</i>	The client ID to which to change to.

Returns

True, if successfully changed. Otherwise, false.

Implements **NetworkModel** (p. 204).

8.138.2.2 `void ServerModel::deleteDisconnectedClient (ClientSocket * client)` [private],[slot]

8.138.2.3 `QVector< unsigned int > ServerModel::getConnectedPlayerIDs ()` [virtual]

Implements **NetworkModel** (p. 205).

8.138.2.4 `QVector< QString > ServerModel::getConnectionList ()` [virtual]

Implements **NetworkModel** (p. 205).

8.138.2.5 `bool ServerModel::isClient ()` [virtual]

Implements **NetworkModel** (p. 205).

8.138.2.6 `bool ServerModel::isServer ()` [virtual]

Implements **NetworkModel** (p. 205).

8.138.2.7 `bool ServerModel::listen ()`

8.138.2.8 `void ServerModel::newConnection ()` [private],[slot]

8.138.2.9 `void ServerModel::receiveData (unsigned int & data, unsigned int playerID = 0)` [virtual]

Receive data from a client with the given playerID.

Parameters

<i>data</i>	The data to hold the data received.
<i>playerID</i>	The ID of the client data is received from.

Implements **NetworkModel** (p. 205).

8.138.2.10 `void ServerModel::receiveData (int & data, unsigned int playerID = 0)` [virtual]

Implements **NetworkModel** (p. 205).

8.138.2.11 `void ServerModel::receiveData (bool & data, unsigned int playerID = 0)` [virtual]

Implements **NetworkModel** (p. 205).

8.138.2.12 void ServerModel::receiveData (Serialize & *data*, unsigned int *playerID* = 0) [virtual]

Implements **NetworkModel** (p. 205).

8.138.2.13 void ServerModel::receiveData (QString & *data*, unsigned int *playerID* = 0) [virtual]

Implements **NetworkModel** (p. 206).

8.138.2.14 void ServerModel::receivePendingData (ClientSocket * *client*) [private],[slot]

8.138.2.15 void ServerModel::sendData (unsigned int *data*, unsigned int *playerID* = 0) [virtual]

Sends the given data to a playerID.

Parameters

<i>data</i>	The data to be sent.
<i>playerID</i>	The ID the data needs to be sent to.

Implements **NetworkModel** (p. 206).

8.138.2.16 void ServerModel::sendData (int *data*, unsigned int *playerID* = 0) [virtual]

Implements **NetworkModel** (p. 206).

8.138.2.17 void ServerModel::sendData (bool *data*, unsigned int *playerID* = 0) [virtual]

Implements **NetworkModel** (p. 206).

8.138.2.18 void ServerModel::sendData (const Serialize & *data*, unsigned int *playerID* = 0) [virtual]

Implements **NetworkModel** (p. 206).

8.138.2.19 void ServerModel::sendData (const QString & *data*, unsigned int *playerID* = 0) [virtual]

Implements **NetworkModel** (p. 206).

8.138.2.20 void ServerModel::socketError (QAbstractSocket::SocketError *socketError*) [private],[slot]

8.138.3 Member Data Documentation

8.138.3.1 QMap<unsigned int, ClientSocket *> ServerModel::clients [private]

A map with all clients. The key is the player's ID.

8.138.3.2 Processor* ServerModel::processor [private]

8.138.3.3 QTcpServer ServerModel::server [private]

The Server connecting to all the clients.

8.138.3.4 WorldModel* ServerModel::worldModel [private]

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

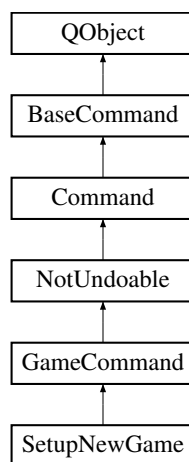
- /home/ryoga/git/pidgirl/pidgirl-engine/Network/**ServerModel.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Network/**ServerModel.cpp**

8.139 SetupNewGame Class Reference

Setup a new game for **PIDGIRL** (p. 37) Engine.

```
#include <SetupNewGame.h>
```

Inheritance diagram for SetupNewGame:



Public Member Functions

- **SetupNewGame** (**GameModel** **gameModelPointer, **QObject** *parent=0)
- virtual bool **execute** ()
Executes the command.

Private Slots

- void **applyGeneralProperties** ()
Apply the General Properties to any game started.
- void **initSoloHotseat** ()
Initialize the Solo / Hotseat Game when triggered.
- void **initNetwork** ()
Initialize the Network when triggered.
- void **initEMailGame** ()
Initialize the E-Mail Game.

Private Member Functions

- bool **initGameSetupDialog** ()
Initialize the Game Setup Dialog.

Private Attributes

- **GameModel ** gameModelPointer**
The Game Model pointer from the Main Window to be set in this.
- QDialog **gameSetupDialog**
- OptionsLayout * **gameSetupLayout**
- GeneralSetupWidget * **generalSetupWidget**
- SoloHotseatSetupWidget * **soloHotseatSetupWidget**
- NetworkSetupWidget * **networkSetupWidget**
- EMailSetupWidget * **emailSetupWidget**

Additional Inherited Members

8.139.1 Detailed Description

Setup a new game for **PIDGIRL** (p. 37) Engine.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.139.2 Constructor & Destructor Documentation

8.139.2.1 SetupNewGame::SetupNewGame (GameModel ** *gameModelPointer*, QObject * *parent* = 0)

8.139.3 Member Function Documentation

8.139.3.1 void SetupNewGame::applyGeneralProperties () [private],[slot]

Apply the General Properties to any game started.

8.139.3.2 bool SetupNewGame::execute () [virtual]

Executes the command.

Returns

True, if successful.

Implements **GameCommand** (p. 133).

8.139.3.3 void SetupNewGame::initEMailGame () [private],[slot]

Initialize the E-Mail Game.

8.139.3.4 `bool SetupNewGame::initGameSetupDialog () [private]`

Initialize the Game Setup Dialog.

Returns

True if initialization was successful, otherwise false.

8.139.3.5 `void SetupNewGame::initNetwork () [private],[slot]`

Initialize the Network when triggered.

8.139.3.6 `void SetupNewGame::initSoloHotseat () [private],[slot]`

Initialize the Solo / Hotseat Game when triggered.

8.139.4 Member Data Documentation

8.139.4.1 `EEmailSetupWidget* SetupNewGame::emailSetupWidget [private]`

8.139.4.2 `GameModel** SetupNewGame::gameModelPointer [private]`

The Game Model pointer from the Main Window to be set in this.

8.139.4.3 `QDialog SetupNewGame::gameSetupDialog [private]`

8.139.4.4 `OptionsLayout* SetupNewGame::gameSetupLayout [private]`

8.139.4.5 `GeneralSetupWidget* SetupNewGame::generalSetupWidget [private]`

8.139.4.6 `NetworkSetupWidget* SetupNewGame::networkSetupWidget [private]`

8.139.4.7 `SoloHotseatSetupWidget* SetupNewGame::soloHotseatSetupWidget [private]`

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

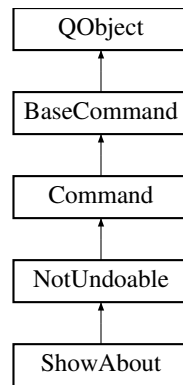
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/**SetupNewGame.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/**SetupNewGame.cpp**

8.140 ShowAbout Class Reference

Show the about dialog.

```
#include <ShowAbout.h>
```

Inheritance diagram for ShowAbout:



Public Member Functions

- **ShowAbout** (`QObject *parent=0`)
- virtual bool **execute** ()
Executes the command.

Additional Inherited Members

8.140.1 Detailed Description

Show the about dialog.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.140.2 Constructor & Destructor Documentation

8.140.2.1 `ShowAbout::ShowAbout (QObject * parent = 0)`

8.140.3 Member Function Documentation

8.140.3.1 `bool ShowAbout::execute () [virtual]`

Executes the command.

Returns

True, if successful.

Implements **NotUndoable** (p. 215).

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

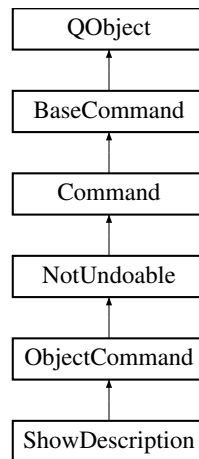
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/ShowAbout.h`
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/ShowAbout.cpp`

8.141 ShowDescription Class Reference

Shows the description of a given object.

```
#include <ShowDescription.h>
```

Inheritance diagram for ShowDescription:



Public Member Functions

- **ShowDescription** (const **ObjectBase** ***object**, `QObject` ***parent**=0)
- virtual bool **execute** ()
Executes the command.

Additional Inherited Members

8.141.1 Detailed Description

Shows the description of a given object.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.141.2 Constructor & Destructor Documentation

8.141.2.1 `ShowDescription::ShowDescription (const ObjectBase * object, QObject * parent = 0)`

8.141.3 Member Function Documentation

8.141.3.1 `bool ShowDescription::execute ()` [virtual]

Executes the command.

Returns

True, if successful.

Implements **ObjectCommand** (p. 227).

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

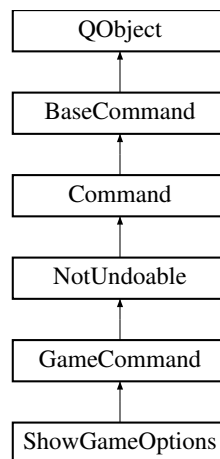
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Object/**ShowDescription.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Object/**ShowDescription.cpp**

8.142 ShowGameOptions Class Reference

Show the options for the game.

```
#include <ShowGameOptions.h>
```

Inheritance diagram for ShowGameOptions:

**Public Member Functions**

- **ShowGameOptions** (QObject *parent=0)
- virtual bool **execute** ()

Executes the command.

Private Slots

- void **saveGame** ()
- void **addHotseatPlayer** ()
- void **leaveGame** ()
- void **choseSaveDir** ()

Private Member Functions

- bool **init** ()

Private Attributes

- QDialog **gameOptionsDialog**
- **OptionsLayout** * **gameOptionsLayout**
- QWidget * **specificGameTypeWidget**
- QGridLayout * **specificGameTypeLayout**
- QPushButton * **saveGameButton**
- QPushButton * **leaveGameButton**
- QPushButton * **addNewPlayerButton**
- QLabel * **saveDirLabel**
- QLineEdit * **saveDirLineEdit**
- QPushButton * **saveDirChooseButton**

Additional Inherited Members

8.142.1 Detailed Description

Show the options for the game.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.142.2 Constructor & Destructor Documentation

8.142.2.1 ShowGameOptions::ShowGameOptions (QObject * *parent* = 0)

8.142.3 Member Function Documentation

8.142.3.1 void ShowGameOptions::addHotseatPlayer () [private],[slot]

8.142.3.2 void ShowGameOptions::choseSaveDir () [private],[slot]

8.142.3.3 bool ShowGameOptions::execute () [virtual]

Executes the command.

Returns

True, if successful.

Implements **GameCommand** (p. 133).

8.142.3.4 bool ShowGameOptions::init () [private]

8.142.3.5 void ShowGameOptions::leaveGame () [private],[slot]

8.142.3.6 void ShowGameOptions::saveGame () [private],[slot]

8.142.4 Member Data Documentation

- 8.142.4.1 `QPushButton*` `ShowGameOptions::addNewPlayerButton` [private]
- 8.142.4.2 `QDialog` `ShowGameOptions::gameOptionsDialog` [private]
- 8.142.4.3 `OptionsLayout*` `ShowGameOptions::gameOptionsLayout` [private]
- 8.142.4.4 `QPushButton*` `ShowGameOptions::leaveGameButton` [private]
- 8.142.4.5 `QPushButton*` `ShowGameOptions::saveDirChooseButton` [private]
- 8.142.4.6 `QLabel*` `ShowGameOptions::saveDirLabel` [private]
- 8.142.4.7 `QLineEdit*` `ShowGameOptions::saveDirLineEdit` [private]
- 8.142.4.8 `QPushButton*` `ShowGameOptions::saveGameButton` [private]
- 8.142.4.9 `QGridLayout*` `ShowGameOptions::specificGameTypeLayout` [private]
- 8.142.4.10 `QWidget*` `ShowGameOptions::specificGameTypeWidget` [private]

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

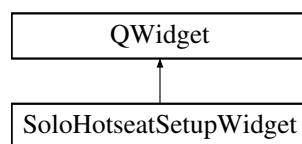
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/ShowGameOptions.h`
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/ShowGameOptions.cpp`

8.143 SoloHotseatSetupWidget Class Reference

The widget to set up an Solo / Hotseat Game.

```
#include <SoloHotseatSetupWidget.h>
```

Inheritance diagram for SoloHotseatSetupWidget:



Public Member Functions

- **SoloHotseatSetupWidget** (`QWidget *parent=0`)

Public Attributes

- `QGridLayout *` **soloHotseatWidgetLayout**
- `QPushButton *` **gameStartButton**

8.143.1 Detailed Description

The widget to set up an Solo / Hotseat Game.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

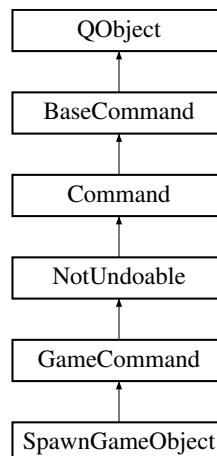
8.143.2 Constructor & Destructor Documentation8.143.2.1 `SoloHotseatSetupWidget::SoloHotseatSetupWidget (QWidget * parent = 0)` `[explicit]`**8.143.3 Member Data Documentation**8.143.3.1 `QPushButton* SoloHotseatSetupWidget::gameStartButton`8.143.3.2 `QGridLayout* SoloHotseatSetupWidget::soloHotseatWidgetLayout`

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

- `/home/ryoga/git/pidgirl/pidgirl-engine/Game/SoloHotseatSetupWidget.h`
- `/home/ryoga/git/pidgirl/pidgirl-engine/Game/SoloHotseatSetupWidget.cpp`

8.144 SpawnGameObject Class ReferenceGeneric **Command** (p. 93) to spawn an object.`#include <SpawnGameObject.h>`

Inheritance diagram for SpawnGameObject:

**Public Member Functions**

- **SpawnGameObject** (**ObjectBase** *spawnableObject, **StackModel** *stackModel, **QObject** *parent=0)
- virtual bool **execute** ()

Executes the command.

Private Attributes

- **ObjectBase * spawnableObject**
Object to be spawned.
- **StackModel * stackModel**
The stack where to spawn the object.

Additional Inherited Members

8.144.1 Detailed Description

Generic **Command** (p. 93) to spawn an object.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.144.2 Constructor & Destructor Documentation

8.144.2.1 **SpawnGameObject::SpawnGameObject (ObjectBase * spawnableObject, StackModel * stackModel, QObject * parent = 0)**

8.144.3 Member Function Documentation

8.144.3.1 **bool SpawnGameObject::execute () [virtual]**

Executes the command.

Returns

True, if successful.

Implements **GameCommand** (p. 133).

8.144.4 Member Data Documentation

8.144.4.1 **ObjectBase* SpawnGameObject::spawnableObject [private]**

Object to be spawned.

8.144.4.2 **StackModel* SpawnGameObject::stackModel [private]**

The stack where to spawn the object.

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

- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/**SpawnGameObject.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/**SpawnGameObject.cpp**

8.145 SpawnPlayer Class Reference

Spawn a player object.

```
#include <SpawnPlayerOnCurrentMap.h>
```

8.145.1 Detailed Description

Spawn a player object.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

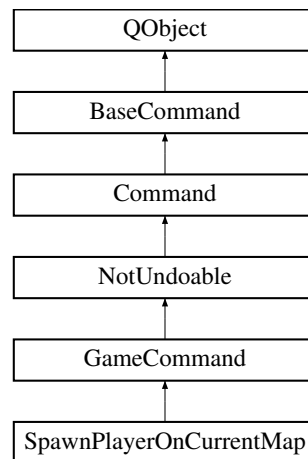
The documentation for this class was generated from the following file:

- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/**SpawnPlayerOnCurrentMap.h**

8.146 SpawnPlayerOnCurrentMap Class Reference

```
#include <SpawnPlayerOnCurrentMap.h>
```

Inheritance diagram for SpawnPlayerOnCurrentMap:



Public Member Functions

- **SpawnPlayerOnCurrentMap** (const QString &**playerName**, QObject *parent=0)
- virtual bool **execute** ()
Executes the command.

Private Attributes

- QString **playerName**
Spawn a player with the given player name.

Additional Inherited Members

8.146.1 Constructor & Destructor Documentation

8.146.1.1 `SpawnPlayerOnCurrentMap::SpawnPlayerOnCurrentMap (const QString & playerName, QObject * parent = 0)`

8.146.2 Member Function Documentation

8.146.2.1 `bool SpawnPlayerOnCurrentMap::execute () [virtual]`

Executes the command.

Returns

True, if successful.

Implements **GameCommand** (p. 133).

8.146.3 Member Data Documentation

8.146.3.1 `QString SpawnPlayerOnCurrentMap::playerName [private]`

Spawn a player with the given player name.

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

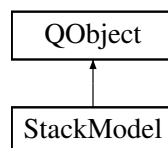
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/SpawnPlayerOnCurrentMap.h`
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/SpawnPlayerOnCurrentMap.cpp`

8.147 StackModel Class Reference

A single stack (square) holding a stack of items and informations about the square which holds them.

```
#include <StackModel.h>
```

Inheritance diagram for StackModel:



Public Types

- enum **PathIndication** {
NONE =0, **N** =1, **S** =2, **W** =4,
NW =5, **SW** =6, **E** =8, **NE** =9,
SE =10, **DESTINATION** }

Indicates the path where the next step is going.

Signals

- void **characterChanged** ()

- void **startPositionSet** (**StackModel** *)
- void **startPositionRemoved** (**StackModel** *)
- void **playerSet** (**StackModel** *)
- void **playerRemoved** (**StackModel** *)
- void **itemListChanged** ()
- void **specialListChanged** ()
- void **pathIndicationChanged** ()
- void **wallChanged** ()
- void **floorChanged** ()

Public Member Functions

- **StackModel** (int xPos, int yPos, int **height**, QObject *parent=0)
- **StackModel** (QPoint pos, int **height**, QObject *parent=0)
- **StackModel** (const **StackModel** &stackModel)
- bool **specialIDExists** (**ObjectID::SpecialID** id) const
Checks if the id is present on this stack.
- void **removeAll** ()
Clear the stack.
- void **deleteAll** ()
Clear the stack and free all the memory.
- void **initStack** (const QPoint &**squarePos**, int **height**)
Initialize Attributes.
- void **resetPathDirection** ()
Delete the path direction.
- void **prependItem** (**ObjectItem** *item)
Prepend an item to the list of items.
- void **removeItem** (int i=0)
Remove i-th element from the list.
- **ObjectItem** * **takeItem** (int i=0)
Take i-th element from the list.
- void **deleteItem** (int i=0)
Delete the i-th element of the list by freeing the space and removing it from the list.
- void **removeAllItems** ()
Remove all items by removing it from the list.
- QList< **ObjectItem** * > **takeAllItems** ()
Take the whole list of items.
- void **deleteAllItems** ()
Delete all items by freeing the space and removing it from the list.
- int **itemListSize** () const
Return the size of the list of items.
- void **prependSpecial** (**ObjectSpecial** *special)
Prepend a special to the list of specials.
- void **removeSpecial** (int i=0)
Remove i-th element from the list.
- **ObjectSpecial** * **takeSpecial** (int i=0)
Take i-th element from the list.
- void **deleteSpecial** (int i=0)
Delete the i-th element of the list by freeing the space and removing it from the list.
- void **removeAllSpecials** ()
Remove all specials by removing it from the list.

- **QList< ObjectSpecial * > takeAllSpecial ()**
Take the whole list of specials.
- **void deleteAllSpecials ()**
Delete all specials by freeing the space and removing it from the list.
- **int specialListSize () const**
Return the size of the list of specials.
- **bool hasSpaceForSpecial (ObjectSpecial *special) const**
Check if special can be fit into the list of specials.
- **ObjectCharacter * replaceCharacter (ObjectCharacter *character)**
Replace the old primary character.
- **ObjectCharacter * takeCharacter ()**
Take the primary character and return it.
- **void removeCharacter ()**
Remove the primary character by setting it to 0.
- **void deleteCharacter ()**
Delete the primary character by freeing the space and setting it to 0.
- **bool hasCharacter () const**
Checks if there is already a character set.
- **bool characterExists (ObjectCharacter *character) const**
Checks if a given character exists.
- **bool characterExists (ObjectID::CharacterID characterID, unsigned int spawnID) const**
- **ObjectTile * replaceWall (ObjectTile *wall)**
Replace the old wall.
- **ObjectTile * takeWall ()**
Take the wall and return it.
- **void removeWall ()**
Remove the wall by setting it to 0.
- **void deleteWall ()**
Delete the wall by freeing the space and setting it to 0.
- **bool hasWall () const**
Checks if there is already a wall set.
- **ObjectTile * replaceFloor (ObjectTile *floor)**
Replace the old floor.
- **ObjectTile * takeFloor ()**
Take the floor and return it.
- **void removeFloor ()**
Remove the floor by setting it to 0.
- **void deleteFloor ()**
Delete the floor by freeing the space and setting it to 0.
- **bool hasFloor () const**
Checks if there is already a floor set.
- **QPoint getSquarePos () const**
- **int getHeight () const**
- **bool isEmpty () const**
- **void setPosition (int x, int y, int z)**
- **void setItemList (const QList< ObjectItem * > &itemList)**
- **void setSpecialList (const QList< ObjectSpecial * > &specialList)**
- **void setPathIndication (StackModel::PathIndication pathIndication)**
- **void setCharacter (ObjectCharacter *character)**
- **void setWall (ObjectTile *wall)**
- **void setFloor (ObjectTile *floor)**

- **StackModel::PathIndication getPathIndication ()** const
- **ObjectCharacter * refCharacter ()** const
- **QList< ObjectItem * > * refItemList ()**
- **ObjectItem * refItem (int i)** const
- **QList< ObjectSpecial * > * refSpecialList ()**
- **ObjectSpecial * refSpecial (int i)** const
- **ObjectTile * refWall ()** const
- **ObjectTile * refFloor ()** const
- void **serialize** (QDataStream &dataStream) const
- void **deserialize** (QDataStream &dataStream)

Private Attributes

- QPoint **squarePos**
- int **height**
- **PathIndication pathIndication**
- **ObjectCharacter * character**
- **QList< ObjectItem * > itemList**
- **QList< ObjectSpecial * > specialList**
- **ObjectTile * wall**
- **ObjectTile * floor**

8.147.1 Detailed Description

A single stack (square) holding a stack of items and informations about the square which holds them.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.147.2 Member Enumeration Documentation

8.147.2.1 enum StackModel::PathIndication

Indicates the path where the next step is going.

Note

It is made, so that you can combine N and S with W and E with the logical OR or addition to result in a combination of both.

Enumerator

NONE

N

S

W

NW

SW***E******NE******SE******DESTINATION***

8.147.3 Constructor & Destructor Documentation

8.147.3.1 **StackModel::StackModel** (int *xPos*, int *yPos*, int *height*, QObject * *parent* = 0) [explicit]

8.147.3.2 **StackModel::StackModel** (QPoint *pos*, int *height*, QObject * *parent* = 0) [explicit]

8.147.3.3 **StackModel::StackModel** (const StackModel & *stackModel*) [explicit]

8.147.4 Member Function Documentation

8.147.4.1 **void StackModel::characterChanged** () [signal]

8.147.4.2 **bool StackModel::characterExists** (ObjectCharacter * *character*) const

Checks if a given character exists.

Parameters

<i>character</i>	The character to be checked.
------------------	------------------------------

Returns

Returns true, if there exists the given character. Otherwise false.

8.147.4.3 **bool StackModel::characterExists** (ObjectID::CharacterID *characterID*, unsigned int *spawnID*) const

8.147.4.4 **void StackModel::deleteAll** ()

Clear the stack and free all the memory.

8.147.4.5 **void StackModel::deleteAllItems** ()

Delete all items by freeing the space and removing it from the list.

8.147.4.6 **void StackModel::deleteAllSpecials** ()

Delete all specials by freeing the space and removing it from the list.

8.147.4.7 **void StackModel::deleteCharacter** ()

Delete the primary character by freeing the space and setting it to 0.

8.147.4.8 **void StackModel::deleteFloor** ()

Delete the floor by freeing the space and setting it to 0.

8.147.4.9 void StackModel::deletetitem (int *i* = 0)

Delete the *i*-th element of the list by freeing the space and removing it from the list.

Parameters

<i>i</i>	The index of the element to be deleted. 0 if no argument given.
----------	---

8.147.4.10 void StackModel::deleteSpecial (int *i* = 0)

Delete the *i*-th element of the list by freeing the space and removing it from the list.

Parameters

<i>i</i>	The index of the element to be deleted. 0 if no argument given.
----------	---

8.147.4.11 void StackModel::deleteWall ()

Delete the wall by freeing the space and setting it to 0.

8.147.4.12 void StackModel::deserialize (QDataStream & *dataStream*)

8.147.4.13 void StackModel::floorChanged () [signal]

8.147.4.14 int StackModel::getHeight () const

8.147.4.15 StackModel::PathIndication StackModel::getPathIndication () const

8.147.4.16 QPoint StackModel::getSquarePos () const

8.147.4.17 bool StackModel::hasCharacter () const

Checks if there is already a character set.

Returns

Returns true, if the stack already has a character. Otherwise, false.

8.147.4.18 bool StackModel::hasFloor () const

Checks if there is already a floor set.

Returns

Returns true, if the stack already has a floor. Otherwise, false.

8.147.4.19 bool StackModel::hasSpaceForSpecial (ObjectSpecial * *special*) const

Check if *special* can be fit into the list of specials.

Parameters

<i>special</i>	The special to be checked if there is still space left to put it in.
----------------	--

Returns

True if it fits, False otherwise.

8.147.4.20 **bool** StackModel::hasWall () const

Checks if there is already a wall set.

Returns

Returns true, if the stack already has a wall. Otherwise, false.

8.147.4.21 **void** StackModel::initStack (const QPoint & *squarePos*, int *height*)

Initialize Attributes.

8.147.4.22 **bool** StackModel::isEmpty () const

8.147.4.23 **void** StackModel::itemListChanged () [signal]

8.147.4.24 **int** StackModel::itemListSize () const

Return the size of the list of items.

Returns

The size of the list of items.

8.147.4.25 **void** StackModel::pathIndicationChanged () [signal]

8.147.4.26 **void** StackModel::playerRemoved (StackModel *) [signal]

8.147.4.27 **void** StackModel::playerSet (StackModel *) [signal]

8.147.4.28 **void** StackModel::prependItem (ObjectItem * *item*)

Prepend an item to the list of items.

Parameters

<i>item</i>	The item to be prepended.
-------------	---------------------------

8.147.4.29 **void** StackModel::prependSpecial (ObjectSpecial * *special*)

Prepend a special to the list of specials.

Parameters

<i>special</i>	The special to be prepended.
----------------	------------------------------

8.147.4.30 **ObjectCharacter *** StackModel::refCharacter () const

8.147.4.31 **ObjectTile *** StackModel::refFloor () const

8.147.4.32 **ObjectItem *** StackModel::refItem (int *i*) const

8.147.4.33 **QList< ObjectItem * > *** StackModel::refItemList ()

8.147.4.34 **ObjectSpecial * StackModel::refSpecial (int *i*) const**

8.147.4.35 **QList< ObjectSpecial * > * StackModel::refSpecialList ()**

8.147.4.36 **ObjectTile * StackModel::refWall () const**

8.147.4.37 **void StackModel::removeAll ()**

Clear the stack.

8.147.4.38 **void StackModel::removeAllItems ()**

Remove all items by removing it from the list.

8.147.4.39 **void StackModel::removeAllSpecials ()**

Remove all specials by removing it from the list.

8.147.4.40 **void StackModel::removeCharacter ()**

Remove the primary character by setting it to 0.

8.147.4.41 **void StackModel::removeFloor ()**

Remove the floor by setting it to 0.

8.147.4.42 **void StackModel::removeItem (int *i* = 0)**

Remove i-th element from the list.

Parameters

<i>i</i>	The index of the element to be removed. 0 if no argument given.
----------	---

8.147.4.43 **void StackModel::removeSpecial (int *i* = 0)**

Remove i-th element from the list.

Parameters

<i>i</i>	The index of the element to be removed. 0 if no argument given.
----------	---

8.147.4.44 **void StackModel::removeWall ()**

Remove the wall by setting it to 0.

8.147.4.45 **ObjectCharacter * StackModel::replaceCharacter (ObjectCharacter * *character*)**

Replace the old primary character.

Parameters

<i>primary-Character</i>	The primary character to be added to this stack.
--------------------------	--

Returns

The old primary character or 0.

8.147.4.46 **ObjectTile * StackModel::replaceFloor (ObjectTile * floor)**

Replace the old floor.

Parameters

<i>wall</i>	The floor to be added to this stack.
-------------	--------------------------------------

Returns

The old floor or 0.

8.147.4.47 **ObjectTile * StackModel::replaceWall (ObjectTile * wall)**

Replace the old wall.

Parameters

<i>wall</i>	The wall to be added to this stack.
-------------	-------------------------------------

Returns

The old wall or 0.

8.147.4.48 **void StackModel::resetPathDirection ()**

Delete the path direction.

8.147.4.49 **void StackModel::serialize (QDataStream & dataStream) const**

8.147.4.50 **void StackModel::setCharacter (ObjectCharacter * character)**

8.147.4.51 **void StackModel::setFloor (ObjectTile * floor)**

8.147.4.52 **void StackModel::setItemList (const QList< ObjectItem * > & itemList)**

8.147.4.53 **void StackModel::setPathIndication (StackModel::PathIndication pathIndication)**

8.147.4.54 **void StackModel::setPosition (int x, int y, int z)**

8.147.4.55 **void StackModel::setSpecialList (const QList< ObjectSpecial * > & specialList)**

8.147.4.56 **void StackModel::setWall (ObjectTile * wall)**

8.147.4.57 **bool StackModel::specialIDExists (ObjectID::SpecialID id) const**

Checks if the id is present on this stack.

Parameters

<i>id</i>	The id to be checked for.
-----------	---------------------------

Returns

True if the stack contains this special id.

8.147.4.58 void StackModel::specialListChanged () [signal]

8.147.4.59 int StackModel::specialListSize () const

Return the size of the list of specials.

Returns

The size of the list of specials.

8.147.4.60 void StackModel::startPositionRemoved (StackModel *) [signal]

8.147.4.61 void StackModel::startPositionSet (StackModel *) [signal]

8.147.4.62 QList< ObjectItem * > StackModel::takeAllItems ()

Take the whole list of items.

Returns

A list with all items.

8.147.4.63 QList< ObjectSpecial * > StackModel::takeAllSpecial ()

Take the whole list of specials.

Returns

A list with all specials.

8.147.4.64 ObjectCharacter * StackModel::takeCharacter ()

Take the primary character and return it.

Returns

The previously used primary character or 0.

8.147.4.65 ObjectTile * StackModel::takeFloor ()

Take the floor and return it.

Returns

The previously used floor or 0.

8.147.4.66 **ObjectItem** * **StackModel::takeItem** (int *i* = 0)

Take i-th element from the list.

Parameters

<i>i</i>	The index of the element to be taken. 0 if no argument given.
----------	---

Returns

The *i*-th element.

8.147.4.67 **ObjectSpecial * StackModel::takeSpecial (int *i* = 0)**

Take *i*-th element from the list.

Parameters

<i>i</i>	The index of the element to be taken. 0 if no argument given.
----------	---

Returns

The *i*-th element.

8.147.4.68 **ObjectTile * StackModel::takeWall ()**

Take the wall and return it.

Returns

The previously used wall or 0.

8.147.4.69 **void StackModel::wallChanged () [signal]**

8.147.5 Member Data Documentation

8.147.5.1 **ObjectCharacter* StackModel::character** [private]8.147.5.2 **ObjectTile* StackModel::floor** [private]8.147.5.3 **int StackModel::height** [private]8.147.5.4 **QList<ObjectItem*> StackModel::itemList** [private]8.147.5.5 **PathIndication StackModel::pathIndication** [private]8.147.5.6 **QList<ObjectSpecial*> StackModel::specialList** [private]8.147.5.7 **QPoint StackModel::squarePos** [private]8.147.5.8 **ObjectTile* StackModel::wall** [private]

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

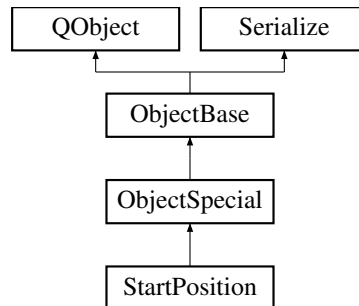
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/**StackModel.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/**StackModel.cpp**

8.148 StartPosition Class Reference

The start position of the player.

```
#include <StartPosition.h>
```

Inheritance diagram for StartPosition:



Public Member Functions

- **StartPosition** (QObject *parent=0)
- **StartPosition** (const **StartPosition** &startPosition)
- virtual QWidget * **createInformationWidget** (QWidget *parent) const
Create an information widget from an object to show informations about it.
- virtual **ObjectSpecial** * **copy** () const
- virtual void **update** (const **ObjectBase** &object)
- virtual **ObjectID::SpecialID** **getObjectID** () const
- virtual QMap< QString, **BaseCommand** * > **getCommandList** () const
- virtual void **serialize** (QDataStream &dataStream) const
- virtual void **serialize** (QIODevice *device) const
- virtual void **deserialize** (QDataStream &dataStream)
- virtual void **deserialize** (QIODevice *device)

Additional Inherited Members

8.148.1 Detailed Description

The start position of the player.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.148.2 Constructor & Destructor Documentation

8.148.2.1 `StartPosition::StartPosition (QObject * parent = 0)`

8.148.2.2 `StartPosition::StartPosition (const StartPosition & startPosition)`

8.148.3 Member Function Documentation

8.148.3.1 `ObjectSpecial * StartPosition::copy () const` [virtual]

Implements **ObjectSpecial** (p. 235).

8.148.3.2 `QWidget * StartPosition::createInformationWidget (QWidget * parent) const` [virtual]

Create an information widget from an object to show informations about it.

Parameters

<i>parent</i>	The parent for the created information widget.
---------------	--

Returns

The widget that represents this object.

Implements **ObjectBase** (p. 220).

8.148.3.3 `void StartPosition::deserialize (QDataStream & dataStream)` [virtual]

Reimplemented from **ObjectSpecial** (p. 236).

8.148.3.4 `void StartPosition::deserialize (QIODevice * device)` [virtual]

Implements **ObjectSpecial** (p. 236).

8.148.3.5 `QMap< QString, BaseCommand * > StartPosition::getCommandList () const` [virtual]

Implements **ObjectBase** (p. 220).

8.148.3.6 `ObjectID::SpecialID StartPosition::getObjectID () const` [virtual]

Implements **ObjectSpecial** (p. 236).

8.148.3.7 `void StartPosition::serialize (QDataStream & dataStream) const` [virtual]

Reimplemented from **ObjectSpecial** (p. 236).

8.148.3.8 `void StartPosition::serialize (QIODevice * device) const` [virtual]

Implements **ObjectSpecial** (p. 236).

8.148.3.9 void StartPosition::update (const ObjectBase & object) [virtual]

Implements **ObjectSpecial** (p. 236).

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

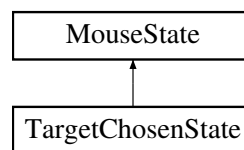
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/Special/**StartPosition.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/Special/**StartPosition.cpp**

8.149 TargetChosenState Class Reference

The target is chosen.

```
#include <TargetChosenState.h>
```

Inheritance diagram for TargetChosenState:



Public Member Functions

- **TargetChosenState** (**Processor** *processor, **StackModel** *target)
- virtual **MouseState** * **nextState** (**MouseState::MouseEventType** type, QMouseEvent *event, **StackModel** *targetStack, **StackModel** *fromStack=0)

Determines which state is next, depending on the input. This method might destroy this state and changes occur.

Additional Inherited Members

8.149.1 Detailed Description

The target is chosen.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.149.2 Constructor & Destructor Documentation

8.149.2.1 TargetChosenState::TargetChosenState (**Processor** * processor, **StackModel** * target)

8.149.3 Member Function Documentation

8.149.3.1 **MouseState** * TargetChosenState::nextState (**MouseState::MouseEventType** type, QMouseEvent * event, **StackModel** * targetStack, **StackModel** * fromStack = 0) [virtual]

Determines which state is next, depending on the input. This method might destroy this state and changes occur.

Parameters

<i>event</i>	Holds the event that has been triggered.
<i>stack</i>	The stack where this method has been invoked from.

Returns

a pointer to the next state.

Implements **MouseState** (p. 197).

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

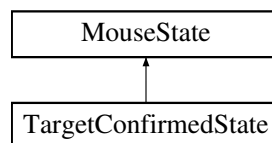
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/TargetChosenState.h
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/TargetChosenState.cpp

8.150 TargetConfirmedState Class Reference

The chosen state is now confirmed.

```
#include <TargetConfirmedState.h>
```

Inheritance diagram for TargetConfirmedState:



Public Member Functions

- **TargetConfirmedState** (**Processor** *processor, **StackModel** *target)
- virtual **MouseState** * **nextState** (**MouseState::MouseEventType** type, QMouseEvent *event, **StackModel** *targetStack, **StackModel** *fromStack=0)

Determines which state is next, depending on the input. This method might destroy this state and changes occur.

Additional Inherited Members

8.150.1 Detailed Description

The chosen state is now confirmed.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.150.2 Constructor & Destructor Documentation

8.150.2.1 `TargetConfirmedState::TargetConfirmedState (Processor * processor, StackModel * target)`

8.150.3 Member Function Documentation

8.150.3.1 `MouseState * TargetConfirmedState::nextState (MouseState::MouseEventType type, QMouseEvent * event, StackModel * targetStack, StackModel * fromStack = 0) [virtual]`

Determines which state is next, depending on the input. This method might destroy this state and changes occur.

Parameters

<i>event</i>	Holds the event that has been triggered.
<i>stack</i>	The stack where this method has been invoked from.

Returns

a pointer to the next state.

Implements **MouseState** (p. 197).

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

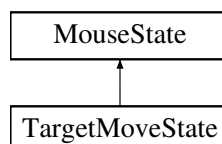
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/TargetConfirmedState.h
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/TargetConfirmedState.cpp

8.151 TargetMoveState Class Reference

Move to the confirmed target.

```
#include <TargetMoveState.h>
```

Inheritance diagram for TargetMoveState:



Public Member Functions

- **TargetMoveState (Processor *processor, StackModel *target)**
- virtual **MouseState * nextState (MouseEventType type, QMouseEvent *event, StackModel *targetStack, StackModel *fromStack=0)**

Determines which state is next, depending on the input. This method might destroy this state and changes occur.

Additional Inherited Members

8.151.1 Detailed Description

Move to the confirmed target.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.151.2 Constructor & Destructor Documentation

8.151.2.1 `TargetMoveState::TargetMoveState (Processor * processor, StackModel * target)`

8.151.3 Member Function Documentation

8.151.3.1 `MouseState * TargetMoveState::nextState (MouseState::MouseEventType type, QMouseEvent * event, StackModel * targetStack, StackModel * fromStack = 0) [virtual]`

Determines which state is next, depending on the input. This method might destroy this state and changes occur.

Parameters

<i>event</i>	Holds the event that has been triggered.
<i>stack</i>	The stack where this method has been invoked from.

Returns

a pointer to the next state.

Implements **MouseState** (p. 197).

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

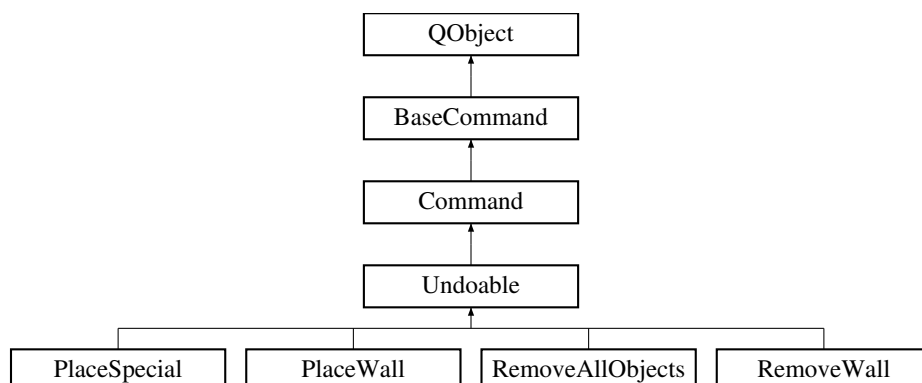
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/TargetMoveState.h
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/TargetMoveState.cpp

8.152 Undoable Class Reference

Abstract class to be inherited by Commands which are undoable.

```
#include <Undoable.h>
```

Inheritance diagram for Undoable:



Public Member Functions

- **Undoable** (QObject *parent=0)
- virtual bool **isExecutable** () const
Check if the command is executable.
- virtual bool **execute** ()=0
Executes the command.
- virtual void **undo** ()=0
Undo changes made by the command.
- virtual void **redo** ()=0
Redo changes made by the undoing of the command.
- virtual bool **isUndoable** ()
Determines undoability.
- virtual bool **isClearingUndoRedo** ()
Determines, if undo and redo should be cleared.

Additional Inherited Members

8.152.1 Detailed Description

Abstract class to be inherited by Commands which are undoable.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.152.2 Constructor & Destructor Documentation

8.152.2.1 **Undoable::Undoable** (QObject * *parent* = 0) [explicit]

8.152.3 Member Function Documentation

8.152.3.1 virtual bool **Undoable::execute** () [pure virtual]

Executes the command.

Returns

True, if successful.

Implements **Command** (p. 94).

Implemented in **RemoveAllObjects** (p. 273), **PlaceWall** (p. 249), **PlaceSpecial** (p. 247), and **RemoveWall** (p. 280).

8.152.3.2 `bool Undoable::isClearingUndoRedo () [virtual]`

Determines, if undo and redo should be cleared.

Returns

True, if undo and redo should be cleared.

Implements **Command** (p. 95).

8.152.3.3 `bool Undoable::isExecutable () const [virtual]`

Check if the command is executable.

Returns

If the command is executable.

Implements **Command** (p. 95).

8.152.3.4 `bool Undoable::isUndoable () [virtual]`

Determines undoability.

Returns

True, if this command can be undone.

Implements **Command** (p. 95).

8.152.3.5 `virtual void Undoable::redo () [pure virtual]`

Redo changes made by the undoing of the command.

Implements **Command** (p. 95).

Implemented in **RemoveAllObjects** (p. 273), **PlaceWall** (p. 249), **PlaceSpecial** (p. 247), and **RemoveWall** (p. 280).

8.152.3.6 `virtual void Undoable::undo () [pure virtual]`

Undo changes made by the command.

Implements **Command** (p. 96).

Implemented in **RemoveAllObjects** (p. 273), **PlaceWall** (p. 249), **PlaceSpecial** (p. 247), and **RemoveWall** (p. 280).

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

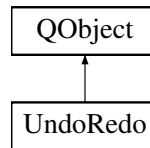
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Undoable.h`
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Undoable.cpp`

8.153 UndoRedo Class Reference

Manage the Undo and Redo functionalities.

```
#include <UndoRedo.h>
```

Inheritance diagram for UndoRedo:



Signals

- void **undoChanged** (bool)
Emited whenever undo list changed.
- void **redoChanged** (bool)
Emited whenever redo list changed.

Public Member Functions

- **UndoRedo** (QObject *parent=0)
- **~UndoRedo** ()
- void **undo** ()
Undo the last command.
- void **redo** ()
Redo the last command.
- void **add** (BaseCommand *command)
Add another command to the undo list.
- void **clear** ()
Clear the undo and redo lists.
- bool **hasOnlySavedMapChanges** ()
- bool **hasOnlySavedWorldChanges** ()
- void **setSavedStates** ()
- void **setOnlySavedWorldChanges** (bool onlySavedWorldChanges)
- void **setOnlySavedMapChanges** (bool onlySavedMapChanges)
- void **setPossibleToReachOriginalState** (bool possibleToReachOriginalState)

Private Member Functions

- void **clearUndoList** ()
- void **clearRedoList** ()

Private Attributes

- QLinkedList< **BaseCommand** * > **undoList**
Undoable (p. 353) command list.
- QLinkedList< **BaseCommand** * > **redoList**
Redoable command list.
- QLinkedList< **BaseCommand** * > **undoListSavedState**
The last undo state.
- QLinkedList< **BaseCommand** * > **redoListSavedState**
The last redo state.

- bool **onlySavedWorldChanges**
True, if the world has been saved since the last changes.
- bool **onlySavedMapChanges**
True, if the map has been saved since the last changed.
- bool **possibleToReachOriginalState**
True, if it's still possible to reach the original state of the map when loaded or created anew.

8.153.1 Detailed Description

Manage the Undo and Redo functionalities.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.153.2 Constructor & Destructor Documentation

8.153.2.1 `UndoRedo::UndoRedo (QObject * parent = 0)` `[explicit]`

8.153.2.2 `UndoRedo::~~UndoRedo ()`

8.153.3 Member Function Documentation

8.153.3.1 `void UndoRedo::add (BaseCommand * command)`

Add another command to the undo list.

Parameters

<i>command</i>	The command to be added to the undo list.
----------------	---

8.153.3.2 `void UndoRedo::clear ()`

Clear the undo and redo lists.

8.153.3.3 `void UndoRedo::clearRedoList ()` `[private]`

8.153.3.4 `void UndoRedo::clearUndoList ()` `[private]`

8.153.3.5 `bool UndoRedo::hasOnlySavedMapChanges ()`

8.153.3.6 `bool UndoRedo::hasOnlySavedWorldChanges ()`

8.153.3.7 `void UndoRedo::redo ()`

Redo the last command.

8.153.3.8 void UndoRedo::redoChanged (bool) [signal]

Emited whenever redo list changed.

8.153.3.9 void UndoRedo::setOnlySavedMapChanges (bool *onlySavedMapChanges*)

8.153.3.10 void UndoRedo::setOnlySavedWorldChanges (bool *onlySavedWorldChanges*)

8.153.3.11 void UndoRedo::setPossibleToReachOriginalState (bool *possibleToReachOriginalState*)

8.153.3.12 void UndoRedo::setSavedStates ()

8.153.3.13 void UndoRedo::undo ()

Undo the last command.

8.153.3.14 void UndoRedo::undoChanged (bool) [signal]

Emited whenever undo list changed.

8.153.4 Member Data Documentation

8.153.4.1 bool UndoRedo::onlySavedMapChanges [private]

True, if the map has been saved since the last changed.

8.153.4.2 bool UndoRedo::onlySavedWorldChanges [private]

True, if the world has been saved since the last changes.

8.153.4.3 bool UndoRedo::possibleToReachOriginalState [private]

True, if it's still possible to reach the original state of the map when loaded or created anew.

8.153.4.4 QList<BaseCommand*> UndoRedo::redoList [private]

Redoable command list.

8.153.4.5 QList<BaseCommand*> UndoRedo::redoListSavedState [private]

The last redo state.

8.153.4.6 QList<BaseCommand*> UndoRedo::undoList [private]

Undoable (p. 353) command list.

8.153.4.7 QList<BaseCommand*> UndoRedo::undoListSavedState [private]

The last undo state.

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

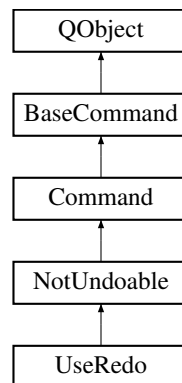
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/UndoRedo.h`
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/UndoRedo.cpp`

8.154 UseRedo Class Reference

Redo the last command.

```
#include <UseRedo.h>
```

Inheritance diagram for UseRedo:



Public Member Functions

- **UseRedo** (`QObject *parent=0`)
- virtual bool **execute** ()
Executes the command.

Additional Inherited Members

8.154.1 Detailed Description

Redo the last command.

Author

RyogaU

Version

0.5

Contact: `RyogaU@googlemail.com`

8.154.2 Constructor & Destructor Documentation

8.154.2.1 `UseRedo::UseRedo (QObject * parent = 0) [explicit]`

8.154.3 Member Function Documentation

8.154.3.1 `bool UseRedo::execute () [virtual]`

Executes the command.

Returns

True, if successful.

Implements **NotUndoable** (p. 215).

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

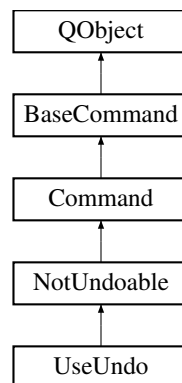
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/**UseRedo.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/**UseRedo.cpp**

8.155 UseUndo Class Reference

Undo the last command.

```
#include <UseUndo.h>
```

Inheritance diagram for UseUndo:



Public Member Functions

- **UseUndo** (QObject *parent=0)
- virtual bool **execute** ()

Executes the command.

Additional Inherited Members

8.155.1 Detailed Description

Undo the last command.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.155.2 Constructor & Destructor Documentation

8.155.2.1 `UseUndo::UseUndo (QObject * parent = 0)` `[explicit]`

8.155.3 Member Function Documentation

8.155.3.1 `bool UseUndo::execute ()` `[virtual]`

Executes the command.

Returns

True, if successful.

Implements **NotUndoable** (p. 215).

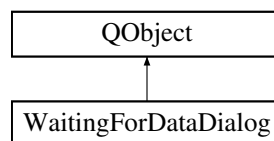
The documentation for this class was generated from the following files:

- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/UseUndo.h`
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/UseUndo.cpp`

8.156 WaitingForDataDialog Class Reference

```
#include <WaitingForDataDialog.h>
```

Inheritance diagram for WaitingForDataDialog:



Public Member Functions

- **WaitingForDataDialog** (**NetworkModel** ***networkModel**, QObject ***parent**=0)
- **WaitingForDataDialog** (**NetworkModel** ***networkModel**, const QString &**title**, const QString &**text**, QObject ***parent**=0)
- bool **waitFor** (**FromClient** **fromClient**)
Wait for a given fromClient enum to be received.
- bool **waitFor** (**FromServer** **fromServer**)
Wait for a given fromServer enum to be received.

Private Slots

- void **dataReceived** (**FromClient** **fromClient**)
- void **dataReceived** (**FromServer** **fromServer**)

Private Attributes

- **NetworkModel** * **networkModel**
- **FromClient** **fromClient**
Wait for a specific fromClient enum from the Client.

- **FromServer fromServer**

Wait for a specific fromServer enum from the Server.

- QMessageBox **messageBox**

The message box informing about what to wait for and giving the possibility to cancel the wait.

- bool **hasBeenReceived**

True, if the waited for enum was received.

8.156.1 Constructor & Destructor Documentation

8.156.1.1 `WaitingForDataDialog::WaitingForDataDialog (NetworkModel * networkModel, QObject * parent = 0)`

8.156.1.2 `WaitingForDataDialog::WaitingForDataDialog (NetworkModel * networkModel, const QString & title, const QString & text, QObject * parent = 0)`

8.156.2 Member Function Documentation

8.156.2.1 `void WaitingForDataDialog::dataReceived (FromClient fromClient) [private],[slot]`

8.156.2.2 `void WaitingForDataDialog::dataReceived (FromServer fromServer) [private],[slot]`

8.156.2.3 `bool WaitingForDataDialog::waitFor (FromClient fromClient)`

Wait for a given fromClient enum to be received.

Parameters

<i>fromClient</i>	The fromClient enum to be received.
-------------------	-------------------------------------

Returns

True, if the waited for fromClient enum was received. Otherwise, false.

8.156.2.4 `bool WaitingForDataDialog::waitFor (FromServer fromServer)`

Wait for a given fromServer enum to be received.

Parameters

<i>fromServer</i>	The fromServer enum to be received.
-------------------	-------------------------------------

Returns

True, if the waited for fromServer enum was received. Otherwise, false.

8.156.3 Member Data Documentation

8.156.3.1 `FromClient WaitingForDataDialog::fromClient [private]`

Wait for a specific fromClient enum from the Client.

8.156.3.2 `FromServer WaitingForDataDialog::fromServer [private]`

Wait for a specific fromServer enum from the Server.

8.156.3.3 `bool` `WaitingForDataDialog::hasBeenReceived` `[private]`

True, if the waited for enum was received.

8.156.3.4 `QMessageBox` `WaitingForDataDialog::messageBox` `[private]`

The message box informing about what to wait for and giving the possibility to cancel the wait.

8.156.3.5 `NetworkModel*` `WaitingForDataDialog::networkModel` `[private]`

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

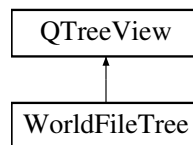
- `/home/ryoga/git/pidgirl/pidgirl-engine/Network/WaitingForDataDialog.h`
- `/home/ryoga/git/pidgirl/pidgirl-engine/Network/WaitingForDataDialog.cpp`

8.157 WorldFileTree Class Reference

Showing the FileTree of an opened world.

```
#include <WorldFileTree.h>
```

Inheritance diagram for WorldFileTree:



Public Member Functions

- `WorldFileTree (Processor *processor, WorldModel *worldModel, QWidget *parent=0)`
- `QMenu * refContextMenu ()`

Protected Member Functions

- virtual void `mouseReleaseEvent` (`QMouseEvent *event`)
The events that occur when the mousebutton is released.
- virtual void `mousePressEvent` (`QMouseEvent *event`)
- virtual void `dragMoveEvent` (`QDragMoveEvent *event`)
- virtual void `dropEvent` (`QDropEvent *`)

Private Slots

- void `indexDoubleClicked` (`QModelIndex index`)
The slot that occurs when someone doubleclicks on an entry.
- void `updateRootIndex` ()
Update the index according to the fileinfo located in the worldWorld.
- void `updateColumnWidth` ()
Update the column width according to whatever is to be shown.
- void `newMap` ()

- Create a new map.*
- void **newFolder** ()
Create a new folder.
- void **rename** ()
Rename a selected entry.
- void **duplicate** ()
Duplicate a selected entry.
- void **remove** ()
Remove a selected entry.

Private Attributes

- **Processor** * **processor**
- **WorldModel** * **worldModel**
- QMenu * **contextMenu**
- QModelIndex **draggedIndex**
- QAction * **newMapAction**
- QAction * **newFolderAction**
- QAction * **renameAction**
- QAction * **duplicateAction**
- QAction * **removeAction**

8.157.1 Detailed Description

Showing the FileTree of an opened world.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.157.2 Constructor & Destructor Documentation

8.157.2.1 **WorldFileTree::WorldFileTree** (**Processor** * *processor*, **WorldModel** * *worldModel*, **QWidget** * *parent* = 0)
[explicit]

8.157.3 Member Function Documentation

8.157.3.1 void **WorldFileTree::dragMoveEvent** (**QDragMoveEvent** * *event*) [protected],[virtual]

8.157.3.2 void **WorldFileTree::dropEvent** (**QDropEvent** *) [protected],[virtual]

8.157.3.3 void **WorldFileTree::duplicate** () [private],[slot]

Duplicate a selected entry.

8.157.3.4 void **WorldFileTree::indexDoubleClicked** (**QModelIndex** *index*) [private],[slot]

The slot that occurs when someone doubleclicks on an entry.

8.157.3.5 void WorldFileTree::mousePressEvent (*QMouseEvent* * *event*) [protected],[virtual]

8.157.3.6 void WorldFileTree::mouseReleaseEvent (*QMouseEvent* * *event*) [protected],[virtual]

The events that occur when the mousebutton is released.

8.157.3.7 void WorldFileTree::newFolder () [private],[slot]

Create a new folder.

8.157.3.8 void WorldFileTree::newMap () [private],[slot]

Create a new map.

8.157.3.9 *QMenu* * WorldFileTree::refContextMenu ()

8.157.3.10 void WorldFileTree::remove () [private],[slot]

Remove a selected entry.

8.157.3.11 void WorldFileTree::rename () [private],[slot]

Rename a selected entry.

8.157.3.12 void WorldFileTree::updateColumnWidth () [private],[slot]

Update the column width according to whatever is to be shown.

8.157.3.13 void WorldFileTree::updateRootIndex () [private],[slot]

Update the index according to the fileinfo located in the worldWorld.

8.157.4 Member Data Documentation

8.157.4.1 *QMenu** WorldFileTree::contextMenu [private]

8.157.4.2 *QModelIndex* WorldFileTree::draggedIndex [private]

8.157.4.3 *QAction** WorldFileTree::duplicateAction [private]

8.157.4.4 *QAction** WorldFileTree::newFolderAction [private]

8.157.4.5 *QAction** WorldFileTree::newMapAction [private]

8.157.4.6 *Processor** WorldFileTree::processor [private]

8.157.4.7 *QAction** WorldFileTree::removeAction [private]

8.157.4.8 *QAction** WorldFileTree::renameAction [private]

8.157.4.9 WorldModel* WorldFileTree::worldModel [private]

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

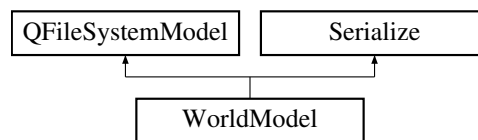
- /home/ryoga/git/pidgirl/pidgirl-engine/Editor/**WorldFileTree.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Editor/**WorldFileTree.cpp**

8.158 WorldModel Class Reference

Represents the world and all of its settings.

```
#include <WorldModel.h>
```

Inheritance diagram for WorldModel:



Signals

- void **fileChanged** ()

Public Member Functions

- **WorldModel** (QObject *parent=0)
- void **setWorldFileInfo** (const QFileInfo &worldFileInfo)
- void **setStartMaps** (const LocalFilePaths &startMaps)
- void **setStartMaps** (const QVector< QString > &pathVector)
- QVector< QString > **getStartMapWorldPaths** () const
- QString **localMapPathToAbsolutePath** (const QString &localMapPath)
- QString **absoluteMapPathToLocalMapPath** (const QString &absoluteMapPath)
- QFileInfo * **refWorldFileInfo** ()
- ObjectID * **refObjectID** ()
- virtual void **serialize** (QIODevice *device) const
- void **serialize** (QDataStream &stream) const
- virtual void **deserialize** (QIODevice *device)
- void **deserialize** (QDataStream &stream)

Private Attributes

- QFileInfo **worldFileInfo**
Information about the world file.
- LocalFilePaths **startMaps**
The list of starting maps.
- ObjectID **objectID**
Information about the next possible ObjectIDs.

8.158.1 Detailed Description

Represents the world and all of its settings.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.158.2 Constructor & Destructor Documentation

8.158.2.1 `WorldModel::WorldModel (QObject * parent = 0)` `[explicit]`

8.158.3 Member Function Documentation

8.158.3.1 `QString WorldModel::absoluteMapPathToLocalMapPath (const QString & absoluteMapPath)`

8.158.3.2 `void WorldModel::deserialize (QIODevice * device)` `[virtual]`

Implements **Serialize** (p. 314).

8.158.3.3 `void WorldModel::deserialize (QDataStream & stream)`

8.158.3.4 `void WorldModel::fileChanged ()` `[signal]`

8.158.3.5 `QVector< QString > WorldModel::getStartMapWorldPaths () const`

8.158.3.6 `QString WorldModel::localMapPathToAbsolutePath (const QString & localMapPath)`

8.158.3.7 `ObjectID * WorldModel::refObjectID ()`

8.158.3.8 `QFileInfo * WorldModel::refWorldFileInfo ()`

8.158.3.9 `void WorldModel::serialize (QIODevice * device) const` `[virtual]`

Implements **Serialize** (p. 314).

8.158.3.10 `void WorldModel::serialize (QDataStream & stream) const`

8.158.3.11 `void WorldModel::setStartMaps (const LocalFilePaths & startMaps)`

8.158.3.12 `void WorldModel::setStartMaps (const QVector< QString > & pathVector)`

8.158.3.13 `void WorldModel::setWorldFileInfo (const QFileInfo & worldFileInfo)`

8.158.4 Member Data Documentation

8.158.4.1 `ObjectID WorldModel::objectID` `[private]`

Information about the next possible ObjectIDs.

8.158.4.2 LocalFilePaths WorldModel::startMaps [private]

The list of starting maps.

8.158.4.3 QFileInfo WorldModel::worldFileInfo [private]

Information about the world file.

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

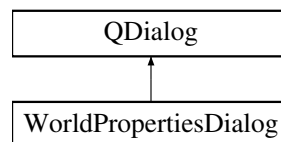
- /home/ryoga/git/pidgirl/pidgirl-engine/Editor/Model/**WorldModel.h**
- /home/ryoga/git/pidgirl/pidgirl-engine/Editor/Model/**WorldModel.cpp**

8.159 WorldPropertiesDialog Class Reference

Shows the properties of a world to change it.

```
#include <WorldPropertiesDialog.h>
```

Inheritance diagram for WorldPropertiesDialog:



Public Slots

- int **exec** ()
- void **addPathToChosen** ()
- void **removePathFromChosen** ()
- void **applyChanges** ()
- void **updateAvailableFileSystemModel** ()

Public Member Functions

- **WorldPropertiesDialog** (Processor *processor, WorldModel *worldModel, QWidget *parent=0)
- void **updateChosenMaps** ()

Private Slots

- void **availableTreeUpdateColumnWidth** ()

Private Attributes

- Processor * **processor**
- WorldModel * **worldModel**
- OptionsLayout * **optionsLayout**
- QWidget * **manageStartingMaps**
- QTreeView * **availableTreeView**
- QFileSystemModel * **availableFileSystemModel**

- `QListView * chosenListView`
- `QStandardItemModel * chosenListModel`
- `QPushButton * addButton`
- `QPushButton * removeButton`
- `QGridLayout * manageStartingMapsLayout`

8.159.1 Detailed Description

Shows the properties of a world to change it.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.159.2 Constructor & Destructor Documentation

8.159.2.1 `WorldPropertiesDialog::WorldPropertiesDialog (Processor * processor, WorldModel * worldModel, QWidget * parent = 0)`

8.159.3 Member Function Documentation

8.159.3.1 `void WorldPropertiesDialog::addPathToChosen () [slot]`

8.159.3.2 `void WorldPropertiesDialog::applyChanges () [slot]`

8.159.3.3 `void WorldPropertiesDialog::availableTreeUpdateColumnWidth () [private],[slot]`

8.159.3.4 `int WorldPropertiesDialog::exec () [slot]`

8.159.3.5 `void WorldPropertiesDialog::removePathFromChosen () [slot]`

8.159.3.6 `void WorldPropertiesDialog::updateAvailableFileSystemModel () [slot]`

8.159.3.7 `void WorldPropertiesDialog::updateChosenMaps ()`

8.159.4 Member Data Documentation

8.159.4.1 `QPushButton* WorldPropertiesDialog::addButton [private]`

8.159.4.2 `QFileSystemModel* WorldPropertiesDialog::availableFileSystemModel [private]`

8.159.4.3 `QTreeView* WorldPropertiesDialog::availableTreeView [private]`

8.159.4.4 `QStandardItemModel* WorldPropertiesDialog::chosenListModel [private]`

8.159.4.5 `QListView* WorldPropertiesDialog::chosenListView [private]`

8.159.4.6 `QWidget* WorldPropertiesDialog::manageStartingMaps [private]`

- 8.159.4.7 `QGridLayout*` `WorldPropertiesDialog::manageStartingMapsLayout` [private]
- 8.159.4.8 `OptionsLayout*` `WorldPropertiesDialog::optionsLayout` [private]
- 8.159.4.9 `Processor*` `WorldPropertiesDialog::processor` [private]
- 8.159.4.10 `QPushButton*` `WorldPropertiesDialog::removeButton` [private]
- 8.159.4.11 `WorldModel*` `WorldPropertiesDialog::worldModel` [private]

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

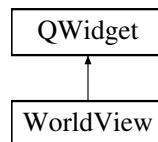
- `/home/ryoga/git/pidgirl/pidgirl-engine/Editor/WorldPropertiesDialog.h`
- `/home/ryoga/git/pidgirl/pidgirl-engine/Editor/WorldPropertiesDialog.cpp`

8.160 WorldView Class Reference

The widget of the world Management.

```
#include <WorldView.h>
```

Inheritance diagram for WorldView:



Public Slots

- void **resizeWorldView** ()

Public Member Functions

- **WorldView** (**Processor** *processor, **WorldModel** *worldModel, QWidget *parent=0)

Private Attributes

- **Processor** * processor
- **WorldModel** * worldModel
- **WorldFileTree** * fileTree

The world file tree representing the world and its maps.

8.160.1 Detailed Description

The widget of the world Management.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.160.2 Constructor & Destructor Documentation

8.160.2.1 `WorldView::WorldView (Processor * processor, WorldModel * worldModel, QWidget * parent = 0)`
`[explicit]`

8.160.3 Member Function Documentation

8.160.3.1 `void WorldView::resizeWorldView ()` `[slot]`

8.160.4 Member Data Documentation

8.160.4.1 `WorldFileTree* WorldView::fileTree` `[private]`

The world file tree representing the world and its maps.

8.160.4.2 `Processor* WorldView::processor` `[private]`

8.160.4.3 `WorldModel* WorldView::worldModel` `[private]`

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

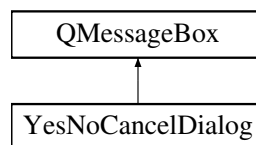
- `/home/ryoga/git/pidgirl/pidgirl-engine/Editor/WorldView.h`
- `/home/ryoga/git/pidgirl/pidgirl-engine/Editor/WorldView.cpp`

8.161 YesNoCancelDialog Class Reference

A wrapper to show yes no cancel dialogs more easily.

```
#include <YesNoCancelDialog.h>
```

Inheritance diagram for YesNoCancelDialog:



Public Types

- enum `ButtonClicked` { `YES_BUTTON`, `NO_BUTTON`, `CANCEL_BUTTON` }

Public Member Functions

- `YesNoCancelDialog ()`
- `void showDialog (const QString &windowTitle, const QString &text)`
Shows a dialog with the given caption and text and saves the button clicked by the user.
- `ButtonClicked getButtonClicked ()`

Private Attributes

- **ButtonClicked** `buttonClicked`
- `QPushButton *` **yesButton**
Represents the button that has been clicked. Standard value is CANCEL.
- `QPushButton *` **noButton**

8.161.1 Detailed Description

A wrapper to show yes no cancel dialogs more easily.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.161.2 Member Enumeration Documentation

8.161.2.1 `enum YesNoCancelDialog::ButtonClicked`

Enumerator

YES_BUTTON

NO_BUTTON

CANCEL_BUTTON

8.161.3 Constructor & Destructor Documentation

8.161.3.1 `YesNoCancelDialog::YesNoCancelDialog ()`

8.161.4 Member Function Documentation

8.161.4.1 `YesNoCancelDialog::ButtonClicked YesNoCancelDialog::getButtonClicked ()`

8.161.4.2 `void YesNoCancelDialog::showDialog (const QString & windowTitle, const QString & text)`

Shows a dialog with the given caption and text and saves the button clicked by the user.

Parameters

<i>windowTitle</i>	The caption shown.
<i>text</i>	The text shown.

8.161.5 Member Data Documentation

8.161.5.1 `ButtonClicked YesNoCancelDialog::buttonClicked` [private]

8.161.5.2 `QPushButton* YesNoCancelDialog::noButton` [private]

8.161.5.3 QPushButton* YesNoCancelDialog::yesButton [private]

Represents the button that has been clicked. Standard value is CANCEL.

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

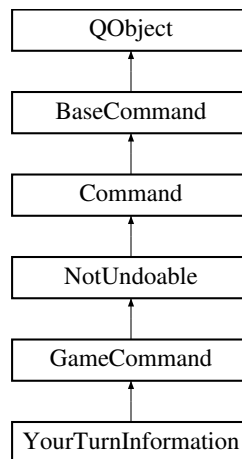
- /home/ryoga/git/pidgirl/pidgirl-engine/Common/YesNoCancelDialog.h
- /home/ryoga/git/pidgirl/pidgirl-engine/Common/YesNoCancelDialog.cpp

8.162 YourTurnInformation Class Reference

Inform with an dialog about it being your turn.

```
#include <YourTurnInformation.h>
```

Inheritance diagram for YourTurnInformation:



Public Member Functions

- **YourTurnInformation** (QObject *parent=0)
- virtual bool **execute** ()

Executes the command.

Additional Inherited Members

8.162.1 Detailed Description

Inform with an dialog about it being your turn.

Author

RyogaU

Version

0.5

Contact: RyogaU@googlemail.com

8.162.2 Constructor & Destructor Documentation

8.162.2.1 `YourTurnInformation::YourTurnInformation (QObject * parent = 0)`

8.162.3 Member Function Documentation

8.162.3.1 `bool YourTurnInformation::execute () [virtual]`

Executes the command.

Returns

True, if successful.

Implements **GameCommand** (p. 133).

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

- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/YourTurnInformation.h`
- `/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/YourTurnInformation.cpp`

Chapter 9

File Documentation

9.1 /home/ryoga/git/pidgirl/pidgirl-engine/AStar/AStarMap.cpp File Reference

```
#include "AStarMap.h"  
#include <QSet>  
#include <math.h>  
#include <iostream>  
#include <limits>
```

9.2 /home/ryoga/git/pidgirl/pidgirl-engine/AStar/AStarMap.h File Reference

```
#include <QVector>  
#include "MainView/Model/MapModel.h"  
#include "AStar/AStarMapEntry.h"
```

Classes

- class **AStarMap**

The representation of the map for the A-Algorithm.*

9.3 /home/ryoga/git/pidgirl/pidgirl-engine/AStar/AStarMapEntry.cpp File Reference

```
#include "AStarMapEntry.h"  
#include <iostream>
```

9.4 /home/ryoga/git/pidgirl/pidgirl-engine/AStar/AStarMapEntry.h File Reference

```
#include "MainView/Model/StackModel.h"  
#include <QHash>  
#include "math.h"  
#include <limits>
```

Classes

- class **AStarMapEntry**

9.5 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/BaseCommand.cpp File Reference

```
#include "BaseCommand.h"
```

9.6 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/BaseCommand.h File Reference

```
#include <QObject>
#include <QIcon>
```

Classes

- class **BaseCommand**
Interface to ensure the least needed functions for the commands.

9.7 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Command.cpp File Reference

```
#include "Command.h"
```

9.8 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Command.h File Reference

```
#include "Command/Backend/BaseCommand.h"
#include "Command/Processor.h"
#include "Command/UndoRedo.h"
#include "Editor/Model/EditorModel.h"
#include "Editor/Model/WorldModel.h"
#include "Game/Model/GameModel.h"
#include "MainView/Model/MapModel.h"
#include "Network/NetworkModel.h"
```

Classes

- class **Command**
The base for all commands.

9.9 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/CommonCommand.cpp File Reference

```
#include "CommonCommand.h"
```

9.10 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/CommonCommand.h File Reference

```
#include "Command/Backend/NotUndoable.h"
```

Classes

- class **CommonCommand**

Interface for commonly used commands.

9.11 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/CopyFolder.cpp File Reference

```
#include "CopyFolder.h"
#include <iostream>
```

9.12 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/CopyFolder.h File Reference

```
#include <QDir>
#include <QList>
#include <QPair>
#include <QFileInfoList>
#include "Command/Backend/Common/CommonCommand.h"
```

Classes

- class **CopyFolder**

Copy a folder recursively to another location.

9.13 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/Quit.cpp File Reference

```
#include "Quit.h"
#include <QDir>
#include <QFileInfo>
#include <QCoreApplication>
#include <QMessageBox>
#include "Command/Backend/Editor/AskForSave.h"
#include "Command/Backend/Common/RemoveFolder.h"
#include "Command/Backend/Game/Ingame/SaveGame.h"
#include "Common/FileNameConventions.h"
#include "Common/Common.h"
#include "MainWindow.h"
```

9.14 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/Quit.h File Reference

```
#include "Command/Backend/Common/CommonCommand.h"
```

Classes

- class **Quit**
Command (p. 93) to quit the running application.

9.15 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/RemoveFolder.cpp File Reference

```
#include "RemoveFolder.h"
#include <QtCore/QFile>
#include <iostream>
```

9.16 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/RemoveFolder.h File Reference

```
#include <QtCore/QDir>
#include <QtCore/QList>
#include <QtCore/QPair>
#include <QtCore/QFileInfoList>
#include "Command/Backend/Common/CommonCommand.h"
```

Classes

- class **RemoveFolder**
Remove the a folder recursively.

9.17 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/AskForSave.cpp File Reference

```
#include "AskForSave.h"
#include "Command/Backend/Editor/Map/AskForSaveMap.h"
#include "Command/Backend/Editor/World/AskForSaveWorld.h"
```

9.18 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/AskForSave.h File Reference

```
#include "Command/Backend/NotUndoable.h"
```

Classes

- class **AskForSave**

Ask if the world or the current map isn't saved, if you want to save.

9.19 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/ApplyMapGeometryChanges.cpp File Reference

```
#include "ApplyMapGeometryChanges.h"
#include <QMessageBox>
#include "Common/Common.h"
```

9.20 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/ApplyMapGeometryChanges.h File Reference

```
#include "Command/Backend/NotUndoable.h"
```

Classes

- class **ApplyMapGeometryChanges**

Apply the geometrical changes to the currently loaded map model.

9.21 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/AskForSaveMap.cpp File Reference

```
#include "AskForSaveMap.h"
#include "Command/Backend/Editor/World/SaveMap.h"
```

9.22 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/AskForSave-Map.h File Reference

```
#include "Command/Backend/NotUndoable.h"  
#include "Common/YesNoCancelDialog.h"
```

Classes

- class **AskForSaveMap**

9.23 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/ChangeZoomFactor.cpp File Reference

```
#include "ChangeZoomFactor.h"
```

9.24 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/ChangeZoomFactor.h File Reference

```
#include "Command/Backend/NotUndoable.h"
```

Classes

- class **ChangeZoomFactor**
Change the zoom factor for a map.

9.25 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/CopyWall.cpp File Reference

```
#include "CopyWall.h"
```

9.26 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/CopyWall.h File Reference

```
#include "Command/Backend/NotUndoable.h"  
#include "MainView/Model/StackModel.h"  
#include "Object/ObjectTile.h"
```

Classes

- class **CopyWall**
Copy the wall to the clipboard.

9.27 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/CutWall.cpp File Reference

```
#include "CutWall.h"
#include "Command/Backend/Editor/Map/CopyWall.h"
#include "Command/Backend/Editor/Map/RemoveWall.h"
```

9.28 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/CutWall.h File Reference

```
#include "Command/Backend/NotUndoable.h"
#include "MainView/Model/StackModel.h"
```

Classes

- class **CutWall**
Copy and delete (Cut) the wall to the clipboard.

9.29 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/LoadMap.cpp File Reference

```
#include "LoadMap.h"
#include <QMessageBox>
#include "Common/Common.h"
#include "Command/Backend/Editor/Map/AskForSaveMap.h"
```

9.30 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/LoadMap.h File Reference

```
#include "Command/Backend/NotUndoable.h"
#include "Common/FileManager.h"
```

Classes

- class **LoadMap**
Load a map from a given path.

9.31 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/LoadMapByIndex.cpp File Reference

```
#include "LoadMapByIndex.h"
#include "Command/Backend/Editor/Map/LoadMap.h"
```

9.32 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/LoadMapByIndex.h File Reference

```
#include <QModelIndex>
#include "Command/Backend/NotUndoable.h"
```

Classes

- class **LoadMapByIndex**
Load a map according to the given path in a model index.

9.33 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/PasteObject.cpp File Reference

```
#include "PasteObject.h"
#include "Command/Backend/Editor/Map/PlaceEditorObject.h"
```

9.34 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/PasteObject.h File Reference

```
#include "Command/Backend/NotUndoable.h"
#include "MainView/Model/StackModel.h"
```

Classes

- class **PasteObject**
Paste an object from the clipboard.

9.35 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/PlaceEditorObject.cpp File Reference

```
#include "PlaceEditorObject.h"
#include "Command/Backend/Editor/Map/PlaceWall.h"
#include "Command/Backend/Editor/Map/PlaceSpecial.h"
#include "Object/Character/Player.h"
#include <iostream>
```

9.36 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/PlaceEditorObject.h File Reference

```
#include "Command/Backend/NotUndoable.h"
#include "Object/ObjectBase.h"
#include "MainView/Model/StackModel.h"
```

Classes

- class **PlaceEditorObject**

9.37 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/PlaceSpecial.cpp File Reference

```
#include "PlaceSpecial.h"
```

9.38 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/PlaceSpecial.h File Reference

```
#include "Command/Backend/Undoable.h"
#include "MainView/Model/StackModel.h"
#include "Object/ObjectSpecial.h"
```

Classes

- class **PlaceSpecial**
Place a special on a chosen stack.

9.39 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/PlaceWall.cpp File Reference

```
#include "PlaceWall.h"
```

9.40 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/PlaceWall.h File Reference

```
#include "Command/Backend/Undoable.h"
#include "MainView/Model/StackModel.h"
#include "Object/ObjectTile.h"
```

Classes

- class **PlaceWall**
Place a wall on a chosen stack.

9.41 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/RemoveAll-Objects.cpp File Reference

```
#include "RemoveAllObjects.h"
```

9.42 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/RemoveAll-Objects.h File Reference

```
#include "Command/Backend/Undoable.h"  
#include "MainView/Model/StackModel.h"  
#include "Object/ObjectCharacter.h"  
#include "Object/ObjectItem.h"  
#include "Object/ObjectTile.h"  
#include "Object/ObjectSpecial.h"
```

Classes

- class **RemoveAllObjects**
Remove all objects from a stackmodel.

9.43 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/Remove-Wall.cpp File Reference

```
#include "RemoveWall.h"
```

9.44 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/Remove-Wall.h File Reference

```
#include "Command/Backend/Undoable.h"  
#include "MainView/Model/StackModel.h"  
#include "Object/ObjectTile.h"
```

Classes

- class **RemoveWall**
Remove a wall from a chosen stack.

9.45 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Tools/Choose-MainTool.cpp File Reference

```
#include "ChooseMainTool.h"  
#include "Object/Tile/BrickWall.h"  
#include "Object/Special/StartPosition.h"
```


9.46 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Tools/ChooseMainTool.h File Reference

```
#include "Command/Backend/NotUndoable.h"
```

Classes

- class **ChooseMainTool**

Choose a Tool and set it in the editorModel.

9.47 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/AddToChosenList.cpp File Reference

```
#include "AddToChosenList.h"  
#include "Common/FileManager.h"  
#include <QMessageBox>
```

9.48 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/AddToChosenList.h File Reference

```
#include <QFileSystemModel>  
#include <QStandardItemModel>  
#include <QModelIndex>  
#include "Command/Backend/NotUndoable.h"  
#include "Common/FileManager.h"
```

Classes

- class **AddToChosenList**

Add the current index to the chosen list.

9.49 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/ApplyWorldPropertiesChanges.cpp File Reference

```
#include "ApplyWorldPropertiesChanges.h"  
#include <QMessageBox>  
#include <Common/Common.h>
```

9.50 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/ApplyWorld-PropertiesChanges.h File Reference

```
#include <QStandardItemModel>
#include "Command/Backend/NotUndoable.h"
```

Classes

- class **ApplyWorldPropertiesChanges**

9.51 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/AskFor-SaveWorld.cpp File Reference

```
#include "AskForSaveWorld.h"
#include "Command/Backend/Editor/World/SaveWorld.h"
```

9.52 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/AskFor-SaveWorld.h File Reference

```
#include "Command/Backend/NotUndoable.h"
#include "Common/YesNoCancelDialog.h"
```

Classes

- class **AskForSaveWorld**
Ask whenever the world isn't saved, if you want to save.

9.53 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/Duplicate-WorldEntry.cpp File Reference

```
#include "DuplicateWorldEntry.h"
#include "Common/Common.h"
```

9.54 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/Duplicate-WorldEntry.h File Reference

```
#include <QModelIndex>
#include "Command/Backend/NotUndoable.h"
```

Classes

- class **DuplicateWorldEntry**

Make a copy of an entry in the world.

9.55 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/LoadWorld.cpp File Reference

```
#include "LoadWorld.h"
#include <QMessageBox>
#include <QFileDialog>
#include <QFileInfo>
#include "Common/Common.h"
#include "Common/FileNameConventions.h"
#include "Common/YesNoCancelDialog.h"
#include "Command/Backend/Editor/World/SaveWorld.h"
#include "Command/Backend/Editor/AskForSave.h"
```

9.56 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/LoadWorld.h File Reference

```
#include "Command/Backend/NotUndoable.h"
#include "Common/FileManager.h"
```

Classes

- class **LoadWorld**

Choose the name of a file and load the world from this file.

9.57 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/MoveFile.cpp File Reference

```
#include "MoveFile.h"
#include <QFileInfo>
#include <QFile>
#include "Common/Common.h"
#include "Command/Backend/Editor/World/RemoveWorldEntry.h"
```

9.58 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/MoveFile.h File Reference

```
#include <QModelIndex>
#include "Command/Backend/NotUndoable.h"
```

Classes

- class **MoveFile**

Move a file from an index to another one.

9.59 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/NewFolder.cpp File Reference

```
#include "NewFolder.h"  
#include "Common/Common.h"  
#include <QInputDialog>
```

9.60 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/NewFolder.h File Reference

```
#include <QModelIndex>  
#include "Command/Backend/NotUndoable.h"
```

Classes

- class **NewFolder**

Create a new folder in the world.

9.61 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/NewMap- File.cpp File Reference

```
#include "NewMapFile.h"  
#include <QtWidgets/QMessageBox>  
#include <QtWidgets/QFileDialog>  
#include <QtWidgets/QInputDialog>  
#include <QtCore/QFileInfo>  
#include "Command/Backend/Editor/AskForSave.h"  
#include "MainView/Model/MapModel.h"  
#include "Common/FileNameConventions.h"  
#include "Common/Common.h"
```

9.62 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/NewMap- File.h File Reference

```
#include <QModelIndexList>  
#include "Command/Backend/NotUndoable.h"  
#include "Common/FileManager.h"
```

Classes

- class **NewMapFile**
Create a new map file.

9.63 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/NewWorld.cpp File Reference

```
#include "NewWorld.h"
#include <QtWidgets/QMessageBox>
#include <QtWidgets/QFileDialog>
#include <QtCore/QFileInfo>
#include "Common/Common.h"
#include "Common/FileNameConventions.h"
#include "Common/YesNoCancelDialog.h"
#include "Command/Backend/Editor/World/SaveWorld.h"
#include "Command/Backend/Editor/AskForSave.h"
```

9.64 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/NewWorld.h File Reference

```
#include "Command/Backend/NotUndoable.h"
#include "Common/FileManager.h"
```

Classes

- class **NewWorld**
Choose the name of a file and create a new world at that location.

9.65 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/RemoveFromChosenList.cpp File Reference

```
#include "RemoveFromChosenList.h"
```

9.66 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/RemoveFromChosenList.h File Reference

```
#include <QStandardItemModel>
#include <QModelIndex>
#include "Command/Backend/NotUndoable.h"
```

Classes

- class **RemoveFromChosenList**

Remove entry of list model.

9.67 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/Remove-WorldEntry.cpp File Reference

```
#include "RemoveWorldEntry.h"  
#include <QMessageBox>  
#include "Common/Common.h"
```

9.68 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/Remove-WorldEntry.h File Reference

```
#include <QModelIndex>  
#include "Command/Backend/NotUndoable.h"
```

Classes

- class **RemoveWorldEntry**

Remove an entry of the world.

9.69 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/Rename-WorldEntry.cpp File Reference

```
#include "RenameWorldEntry.h"  
#include <QInputDialog>  
#include "Common/Common.h"  
#include "Common/FileNameConventions.h"
```

9.70 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/Rename-WorldEntry.h File Reference

```
#include <QModelIndex>  
#include "Command/Backend/NotUndoable.h"
```

Classes

- class **RenameWorldEntry**

Renames an entry of the world.

9.71 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/SaveMap.cpp File Reference

```
#include "SaveMap.h"
#include <QFileDialog>
#include <QMessageBox>
#include "Common/Common.h"
#include "Common/FileNameConventions.h"
#include "Command/Backend/Editor/World/SaveWorld.h"
```

9.72 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/SaveMap.h File Reference

```
#include "Command/Backend/NotUndoable.h"
#include "Common/FileManager.h"
```

Classes

- class **SaveMap**

Save a map. If it wasn't saved before, choose a place and file.

9.73 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/SaveMap- As.cpp File Reference

```
#include "SaveMapAs.h"
#include <QFileDialog>
#include <QMessageBox>
#include "Common/Common.h"
#include "Common/FileNameConventions.h"
#include "Command/Backend/Editor/World/SaveWorld.h"
```

9.74 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/SaveMap- As.h File Reference

```
#include "Command/Backend/NotUndoable.h"
#include "Common/FileManager.h"
```

Classes

- class **SaveMapAs**

Save a map as a speicified file.

9.75 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/SaveWorld.cpp File Reference

```
#include "SaveWorld.h"  
#include <QMessageBox>  
#include <QFileDialog>  
#include <QFileInfo>  
#include "Common/Common.h"
```

9.76 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/SaveWorld.h File Reference

```
#include "Command/Backend/NotUndoable.h"  
#include "Common/FileManager.h"
```

Classes

- class **SaveWorld**

Save the world. The world might not exist, so it's simply created anew.

9.77 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/SaveWorld-As.cpp File Reference

```
#include "SaveWorldAs.h"  
#include <QMessageBox>  
#include <QFileDialog>  
#include <QFileInfo>  
#include "Common/Common.h"  
#include "Common/FileNameConventions.h"
```

9.78 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/World/SaveWorld-As.h File Reference

```
#include "Command/Backend/NotUndoable.h"  
#include "Common/FileManager.h"
```

Classes

- class **SaveWorldAs**

Save the current world as another world.

9.79 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/GameCommand.cpp File Reference

```
#include "GameCommand.h"
#include "Command/Backend/Common/Quit.h"
```

9.80 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/GameCommand.h File Reference

```
#include "Command/Backend/NotUndoable.h"
```

Classes

- class **GameCommand**
Interface for a command done in games.

9.81 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/Create-Character.cpp File Reference

```
#include "CreateCharacter.h"
#include <QMessageBox>
#include <iostream>
#include "Common/Common.h"
#include "Command/Backend/Game/Ingame/LoadStartMapFromIndex.h"
#include "Command/Backend/Game/Ingame/SpawnPlayerOnCurrentMap.h"
#include "Object/Character/Player.h"
#include "Network/WaitingForDataDialog.h"
```

9.82 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/Create-Character.h File Reference

```
#include <QDialog>
#include <QGridLayout>
#include <QStandardItemModel>
#include <QLineEdit>
#include <QVector>
#include <QLabel>
#include "Command/Backend/Game/GameCommand.h"
#include "Common/OptionsLayout.h"
#include "Common/FileManager.h"
```

Classes

- class **CreateCharacter**
Create a character and put his starting information into the game model.

9.83 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/EndTurn.cpp File Reference

```
#include "EndTurn.h"  
#include "Common/Common.h"
```

9.84 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/EndTurn.h File Reference

```
#include "Command/Backend/Game/GameCommand.h"
```

Classes

- class **EndTurn**

Things to do at the end of the turn and prepare to start the next turn.

9.85 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/FirstTurn.cpp File Reference

```
#include "FirstTurn.h"  
#include "Command/Backend/Game/Ingame/CreateCharacter.h"  
#include "Common/Common.h"
```

9.86 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/FirstTurn.h File Reference

```
#include <QDialog>  
#include <QList>  
#include <QGridLayout>  
#include "Command/Backend/Game/GameCommand.h"  
#include "Common/FileManager.h"  
#include "Common/OptionsLayout.h"  
#include "Object/Character/Player.h"
```

Classes

- class **FirstTurn**

Things to do for the very first turn.

9.87 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/LoadStartMapFromIndex.cpp File Reference

```
#include "LoadStartMapFromIndex.h"  
#include <QMessageBox>  
#include "Common/Common.h"
```

9.88 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/LoadStartMapFromIndex.h File Reference

```
#include "Command/Backend/Game/GameCommand.h"
```

Classes

- class **LoadStartMapFromIndex**

Load the map depending on the index given. Can be random as well.

9.89 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/NextTurn.cpp File Reference

```
#include "NextTurn.h"
```

9.90 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/NextTurn.h File Reference

```
#include "Command/Backend/Game/GameCommand.h"
```

Classes

- class **NextTurn**

Begin the next turn.

9.91 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/PreloadAll-Maps.cpp File Reference

```
#include "PreloadAllMaps.h"
#include <QDir>
#include <QMessageBox>
#include <QFileInfoList>
#include <QList>
#include <QFile>
#include <QFileDialog>
#include "Common/Common.h"
#include "Common/FileNameConventions.h"
```

9.92 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/PreloadAll-Maps.h File Reference

```
#include "Command/Backend/Game/GameCommand.h"
#include "Common/FileManager.h"
```

Classes

- class **PreloadAllMaps**

Preload the maps if it's a local game or a server.

9.93 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/SaveEmail-Game.cpp File Reference

```
#include "SaveEmailGame.h"
#include <QDir>
#include <QFileInfo>
#include <QMessageBox>
#include "Common/Common.h"
#include "Common/FileNameConventions.h"
#include "Game/Model/SaveGameFile.h"
```

9.94 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/SaveEmail-Game.h File Reference

```
#include <QString>
#include "Command/Backend/Game/GameCommand.h"
#include "Common/FileManager.h"
#include "Game/Model/MailGameInfo.h"
```

Classes

- class **SaveEMailGame**

Save the current EMail Game.

9.95 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/SaveGame.cpp File Reference

```
#include "SaveGame.h"
#include <QMessageBox>
#include <QDir>
#include <QFileDialog>
#include "Common/Common.h"
#include "Common/FileNameConventions.h"
#include "Game/Model/SaveGameFile.h"
```

9.96 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/SaveGame.h File Reference

```
#include "Command/Backend/Game/GameCommand.h"
#include "Common/FileManager.h"
```

Classes

- class **SaveGame**

Save the current game and world models.

9.97 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/ShowGameOptions.cpp File Reference

```
#include "ShowGameOptions.h"
#include <QMessageBox>
#include <QFileDialog>
#include "Command/Backend/Game/Ingame/SaveGame.h"
#include "Game/Model/EMailGameModel.h"
#include "Game/Model/MailGameInfo.h"
```

9.98 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/Show-GameOptions.h File Reference

```
#include "Command/Backend/Game/GameCommand.h"
#include <QDialog>
#include <QGridLayout>
#include <QPushButton>
#include <QLabel>
#include <QLineEdit>
#include "Common/OptionsLayout.h"
```

Classes

- class **ShowGameOptions**
Show the options for the game.

9.99 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/Spawn-GameObject.cpp File Reference

```
#include "SpawnGameObject.h"
#include <iostream>
```

9.100 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/Spawn-GameObject.h File Reference

```
#include "Command/Backend/Game/GameCommand.h"
#include "Object/ObjectBase.h"
#include "MainView/Model/StackModel.h"
```

Classes

- class **SpawnGameObject**
*Generic **Command** (p. 93) to spawn an object.*

9.101 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/Spawn-PlayerOnCurrentMap.cpp File Reference

```
#include "SpawnPlayerOnCurrentMap.h"
#include <QMessageBox>
#include "Object/ObjectID.h"
```

9.102

/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/SpawnPlayerOnCurrentMap.h File Reference

399

9.102 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/SpawnPlayerOnCurrentMap.h File Reference

```
#include "Command/Backend/Game/GameCommand.h"
#include "Object/Character/Player.h"
#include "MainView/Model/StackModel.h"
```

Classes

- class **SpawnPlayerOnCurrentMap**

9.103 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/YourTurnInformation.cpp File Reference

```
#include "YourTurnInformation.h"
#include <QMessageBox>
```

9.104 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Ingame/YourTurnInformation.h File Reference

```
#include "Command/Backend/Game/GameCommand.h"
```

Classes

- class **YourTurnInformation**
Inform with an dialog about it being your turn.

9.105 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Map/ClearPath.cpp File Reference

```
#include "ClearPath.h"
```

9.106 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Map/ClearPath.h File Reference

```
#include "Command/Backend/Game/GameCommand.h"
```

Classes

- class **ClearPath**
Clear the found path to a destination.

9.107 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Map/FindPath.cpp File Reference

```
#include "FindPath.h"  
#include <iostream>
```

9.108 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Map/FindPath.h File Reference

```
#include "Command/Backend/Game/GameCommand.h"  
#include "MainView/Model/StackModel.h"  
#include "AStar/AStarMap.h"
```

Classes

- class **FindPath**

Creates a path that shows how to get from one stack to another.

9.109 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Map/MoveCharacter.cpp File Reference

```
#include "MoveCharacter.h"
```

9.110 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Map/MoveCharacter.h File Reference

```
#include "Command/Backend/Game/GameCommand.h"  
#include "MainView/Model/StackModel.h"  
#include "Object/ObjectCharacter.h"
```

Classes

- class **MoveCharacter**

Move a character from stack to stack.

9.111 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/ConnectTo.cpp File Reference

```
#include "ConnectTo.h"
#include <QGridLayout>
#include "Command/Backend/Game/Startmenu/PreloadWorld.h"
#include "Common/Common.h"
#include "Network/ClientModel.h"
#include "Network/WaitingForDataDialog.h"
```

9.112 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/ConnectTo.h File Reference

```
#include <QDialog>
#include <QGridLayout>
#include <QPushButton>
#include <QLabel>
#include <QLineEdit>
#include <QRegExpValidator>
#include <QIntValidator>
#include "Command/Backend/Game/GameCommand.h"
#include "Network/ClientModel.h"
```

Classes

- class **ConnectTo**

*Connect to a running **PIDGIRL** (p. 37) Server.*

9.113 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/LoadGame.cpp File Reference

```
#include "LoadGame.h"
#include <QDir>
#include <QFileDialog>
#include "Common/Common.h"
#include "Common/FileNameConventions.h"
#include "Game/Model/HotseatGameModel.h"
#include "Game/Model/ServerGameModel.h"
#include "Network/ServerModel.h"
```

9.114 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/Load-Game.h File Reference

```
#include <QDialog>
#include "Command/Backend/Game/GameCommand.h"
#include "Common/OptionsLayout.h"
#include "Common/FileManager.h"
#include "Game/Model/SaveGameFile.h"
#include "Game/GeneralSetupWidget.h"
#include "Game/SoloHotseatSetupWidget.h"
#include "Game/NetworkSetupWidget.h"
```

Classes

- class **LoadGame**

*Load a game for **PIDGIRL** (p. 37) Engine.*

9.115 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/Preload-World.cpp File Reference

```
#include "PreloadWorld.h"
#include <QFileDialog>
#include <QMessageBox>
#include <QDir>
#include "Common/FileNameConventions.h"
#include "Common/Common.h"
```

9.116 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/Preload-World.h File Reference

```
#include <QFileInfo>
#include "Command/Backend/Game/GameCommand.h"
#include "Common/FileManager.h"
```

Classes

- class **PreloadWorld**

Preload the world.

9.117 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/Resume-EMailGame.cpp File Reference

```
#include "ResumeEMailGame.h"  
#include <QFileDialog>  
#include <QDir>  
#include <QFileInfo>  
#include <QMessageBox>  
#include <QCoreApplication>  
#include "Common/Common.h"  
#include "Common/FileNameConventions.h"  
#include "Game/Model/EMailGameModel.h"  
#include "Game/Model/SaveGameFile.h"
```

9.118 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/Resume-EMailGame.h File Reference

```
#include "Command/Backend/Game/GameCommand.h"  
#include "Game/Model/GameModel.h"  
#include "Common/FileManager.h"
```

Classes

- class **ResumeEMailGame**

Resume an E-Mail Game.

9.119 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/Setup-NewGame.cpp File Reference

```
#include "SetupNewGame.h"  
#include <QPushButton>  
#include <QRegExp>  
#include "Command/Backend/Game/Startmenu/PreloadWorld.h"  
#include "Command/Backend/Game/Ingame/PreloadAllMaps.h"  
#include "Common/Common.h"  
#include "Network/ServerModel.h"  
#include "Game/Model/ServerGameModel.h"  
#include "Game/Model/HotseatGameModel.h"  
#include "Game/Model/EMailGameModel.h"  
#include "Game/Model/PlayerModel.h"  
#include <iostream>
```

9.120 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Game/Startmenu/Setup-NewGame.h File Reference

```
#include <QDialog>
#include <QWidget>
#include <QCheckBox>
#include <QPushButton>
#include <QLabel>
#include <QLineEdit>
#include <QIntValidator>
#include <QRegExpValidator>
#include "Command/Backend/Game/GameCommand.h"
#include "Common/OptionsLayout.h"
#include "Game/GeneralSetupWidget.h"
#include "Game/SoloHotseatSetupWidget.h"
#include "Game/NetworkSetupWidget.h"
#include "Game/EmailSetupWidget.h"
```

Classes

- class **SetupNewGame**

*Setup a new game for **PIDGIRL** (p. 37) Engine.*

9.121 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/Change-PlayerIDOnServer.cpp File Reference

```
#include "ChangePlayerIDOnServer.h"
```

9.122 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/Change-PlayerIDOnServer.h File Reference

```
#include "Command/Backend/Network/Client/ClientCommand.h"
```

Classes

- class **ChangePlayerIDOnServer**

Send the new player ID to the server that will replace the old one.

9.123 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/Client-Command.cpp File Reference

```
#include "ClientCommand.h"
```

9.124 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ClientCommand.h File Reference

```
#include "Command/Backend/Network/NetworkCommand.h"
```

Classes

- class **ClientCommand**
Interface for a command executed by network clients.

9.125 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ReceiveInitialPlayerInformation.cpp File Reference

```
#include "ReceiveInitialPlayerInformation.h"
```

9.126 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ReceiveInitialPlayerInformation.h File Reference

```
#include "Command/Backend/Network/Client/ClientCommand.h"
```

Classes

- class **ReceiveInitialPlayerInformation**
Receive the initial player information.

9.127 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ReceiveMap.cpp File Reference

```
#include "ReceiveMap.h"
```

9.128 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/ReceiveMap.h File Reference

```
#include "Command/Backend/Network/Client/ClientCommand.h"
```

Classes

- class **ReceiveMap**
Receive and load a map from the server.

9.129 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/Receive-Player.cpp File Reference

```
#include "ReceivePlayer.h"
```

9.130 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/Receive-Player.h File Reference

```
#include "Command/Backend/Network/Client/ClientCommand.h"
```

Classes

- class **ReceivePlayer**

Receive a player object from the server.

9.131 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/Receive-PlayerModels.cpp File Reference

```
#include "ReceivePlayerModels.h"
```

9.132 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/Receive-PlayerModels.h File Reference

```
#include "Command/Backend/Network/Client/ClientCommand.h"
```

Classes

- class **ReceivePlayerModels**

Receive the requested player models.

9.133 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/Request-MapFromServer.cpp File Reference

```
#include "RequestMapFromServer.h"  
#include "Network/NetworkModel.h"  
#include "Network/WaitingForDataDialog.h"
```

9.134 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/Request-MapFromServer.h File Reference

```
#include "Command/Backend/Network/Client/ClientCommand.h"
```

Classes

- class **RequestMapFromServer**
Request a specific map from the server.

9.135 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/Request-PlayerFromServer.cpp File Reference

```
#include "RequestPlayerFromServer.h"  
#include "Network/WaitingForDataDialog.h"
```

9.136 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/Request-PlayerFromServer.h File Reference

```
#include "Command/Backend/Network/Client/ClientCommand.h"
```

Classes

- class **RequestPlayerFromServer**
Request a specific player from the server.

9.137 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/Request-PlayerModelsFromServer.cpp File Reference

```
#include "RequestPlayerModelsFromServer.h"  
#include "Network/WaitingForDataDialog.h"
```

9.138 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/Request-PlayerModelsFromServer.h File Reference

```
#include "Command/Backend/Network/Client/ClientCommand.h"
```

Classes

- class **RequestPlayerModelsFromServer**
Request all player models from the server.

9.139 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/Send-SpawnCharacterToServer.cpp File Reference

```
#include "SendSpawnCharacterToServer.h"
```

9.140 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/Send-SpawnCharacterToServer.h File Reference

```
#include "Command/Backend/Network/Client/ClientCommand.h"
```

Classes

- class **SendSpawnCharacterToServer**
Send the character that has been created to the server.

9.141 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/SendTo-ServerEndTurn.cpp File Reference

```
#include "SendToServerEndTurn.h"
```

9.142 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/SendTo-ServerEndTurn.h File Reference

```
#include "Command/Backend/Network/Client/ClientCommand.h"
```

Classes

- class **SendToServerEndTurn**
Send End Turn to the server.

9.143 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/Send-UpdateCharacterToServer.cpp File Reference

```
#include "SendUpdateCharacterToServer.h"
```

9.144 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Client/Send-UpdateCharacterToServer.h File Reference

```
#include "Command/Backend/Network/Client/ClientCommand.h"
```


Classes

- class **SendUpdateCharacterToServer**
Send updated information about a character to the server.

9.145 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/NetworkCommand.cpp File Reference

```
#include "NetworkCommand.h"
```

9.146 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/NetworkCommand.h File Reference

```
#include "Command/Backend/NotUndoable.h"
```

Classes

- class **NetworkCommand**
***Command** (p. 93) sent by Server or Client alike.*

9.147 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/ReceiveSpawn-CharacterFrom.cpp File Reference

```
#include "ReceiveSpawnCharacterFrom.h"  
#include "Object/ObjectFromID.h"
```

9.148 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/ReceiveSpawn-CharacterFrom.h File Reference

```
#include "Command/Backend/Network/NetworkCommand.h"
```

Classes

- class **ReceiveSpawnCharacterFrom**
Receive the character that has been spawned.

9.149 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/ReceiveUpdate-CharacterFrom.cpp File Reference

```
#include "ReceiveUpdateCharacterFrom.h"  
#include "Object/ObjectFromID.h"
```

9.150 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/ReceiveUpdate-CharacterFrom.h File Reference

```
#include "Command/Backend/Network/NetworkCommand.h"
```

Classes

- class **ReceiveUpdateCharacterFrom**
Receive the Updated Character from any given sender ID.

9.151 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Broadcast-/BroadcastCommand.cpp File Reference

```
#include "BroadcastCommand.h"  
#include "Common/Common.h"
```

9.152 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Broadcast-/BroadcastCommand.h File Reference

```
#include "Command/Backend/Network/Server/ServerCommand.h"
```

Classes

- class **BroadcastCommand**
Broadcast information to all clients on the same map the information was sent from.

9.153 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Broadcast-/BroadcastSpawnCharacter.cpp File Reference

```
#include "BroadcastSpawnCharacter.h"
```

9.154 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Broadcast-/BroadcastSpawnCharacter.h File Reference

```
#include "Command/Backend/Network/Server/Broadcast/BroadcastCommand.h"
```

Classes

- class **BroadcastSpawnCharacter**
Broadcast the new character to be spawned.

9.155 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Broadcast/Broadcast-UpdateCharacter.cpp File

Reference

411

9.155 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Broadcast/BroadcastUpdateCharacter.cpp File Reference

```
#include "BroadcastUpdateCharacter.h"
```

9.156 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Broadcast/BroadcastUpdateCharacter.h File Reference

```
#include "Command/Backend/Network/Server/Broadcast/BroadcastCommand.h"
```

Classes

- class **BroadcastUpdateCharacter**
Broadcast the character that has been changed.

9.157 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Client-Disconnected.cpp File Reference

```
#include "ClientDisconnected.h"
```

9.158 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Client-Disconnected.h File Reference

```
#include "Command/Backend/Network/Server/ServerCommand.h"
```

Classes

- class **ClientDisconnected**
***Command** (p. 93) is triggered whenever a client is disconnected.*

9.159 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Receive-ChangePlayerID.cpp File Reference

```
#include "ReceiveChangePlayerID.h"
```

9.160 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Receive-ChangePlayerID.h File Reference

```
#include "Command/Backend/Network/Server/ServerCommand.h"
```

Classes

- class **ReceiveChangePlayerID**

Receive the player ID to which a player connection should change.

9.161 **/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Reply-ToRequestForMap.cpp File Reference**

```
#include "ReplyToRequestForMap.h"
```

9.162 **/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Reply-ToRequestForMap.h File Reference**

```
#include "Command/Backend/Network/Server/ServerCommand.h"
```

Classes

- class **ReplyToRequestForMap**

Reply a requested map to a player.

9.163 **/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Reply-ToRequestForPlayer.cpp File Reference**

```
#include "ReplyToRequestForPlayer.h"
```

9.164 **/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Reply-ToRequestForPlayer.h File Reference**

```
#include "Command/Backend/Network/Server/ServerCommand.h"
```

Classes

- class **ReplyToRequestForPlayer**

Reply a requested player object to a given player.

9.165 **/home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Reply-ToRequestForPlayerModels.cpp File Reference**

```
#include "ReplyToRequestForPlayerModels.h"
```

9.166 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Reply- ToRequestForPlayerModels.h File Reference

```
#include "Command/Backend/Network/Server/ServerCommand.h"
```

Classes

- class **ReplyToRequestForPlayerModels**
Reply requested player models to a player.

9.167 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Send- InitialPlayerInformation.cpp File Reference

```
#include "SendInitialPlayerInformation.h"  
#include <QVector>  
#include "Game/Model/PlayerModel.h"
```

9.168 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Send- InitialPlayerInformation.h File Reference

```
#include "Command/Backend/Network/Server/ServerCommand.h"
```

Classes

- class **SendInitialPlayerInformation**
Send the initial player informations to the player that connected.

9.169 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/SendTo- ClientNextTurn.cpp File Reference

```
#include "SendToClientNextTurn.h"
```

9.170 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/SendTo- ClientNextTurn.h File Reference

```
#include "Command/Backend/Network/Server/ServerCommand.h"
```

Classes

- class **SendToClientNextTurn**
Send to the next client the next turn to start the new turn for that player.

9.171 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Server-Command.cpp File Reference

```
#include "ServerCommand.h"
```

9.172 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/Server-Command.h File Reference

```
#include "Command/Backend/Network/NetworkCommand.h"
```

Classes

- class **ServerCommand**
Command (p. 93) executed by a server.

9.173 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/NotUndoable.cpp File Reference

```
#include "Command/Backend/NotUndoable.h"
```

9.174 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/NotUndoable.h File Reference

```
#include "Command/Backend/Command.h"
```

Classes

- class **NotUndoable**
Abstract class to be inherited by Commands which are not undoable.

9.175 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Object/ObjectCommand.cpp File Reference

```
#include "ObjectCommand.h"
```

9.176 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Object/ObjectCommand.h File Reference

```
#include "Command/Backend/NotUndoable.h"  
#include <QString>
```

Classes

- class **ObjectCommand**

A command corresponding and holding to a given object.

9.177 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Object/ShowDescription.cpp File Reference

```
#include "ShowDescription.h"
#include <QMessageBox>
```

9.178 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Object/ShowDescription.h File Reference

```
#include "Command/Backend/Object/ObjectCommand.h"
```

Classes

- class **ShowDescription**

Shows the description of a given object.

9.179 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/ShowAbout.cpp File Reference

```
#include "ShowAbout.h"
#include <QtWidgets/QMessageBox>
#include "Common/Common.h"
```

9.180 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/ShowAbout.h File Reference

```
#include "Command/Backend/NotUndoable.h"
```

Classes

- class **ShowAbout**

Show the about dialog.

9.181 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Undoable.cpp File Reference

```
#include "Undoable.h"
```

9.182 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Undoable.h File Reference

```
#include "Command/Backend/Command.h"
```

Classes

- class **Undoable**

Abstract class to be inherited by Commands which are undoable.

9.183 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/UseRedo.cpp File Reference

```
#include "UseRedo.h"
```

9.184 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/UseRedo.h File Reference

```
#include "Command/Backend/NotUndoable.h"  
#include "Command/Processor.h"
```

Classes

- class **UseRedo**

Redo the last command.

9.185 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/UseUndo.cpp File Reference

```
#include "UseUndo.h"
```


9.186 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/UseUndo.h File Reference

```
#include "Command/Backend/NotUndoable.h"
#include "Command/Processor.h"
```

Classes

- class **UseUndo**

Undo the last command.

9.187 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Processor.cpp File Reference

```
#include "Command/Processor.h"
#include <iostream>
```

9.188 /home/ryoga/git/pidgirl/pidgirl-engine/Command/Processor.h File Reference

```
#include <QObject>
#include "Object/ObjectBase.h"
#include "Command/UndoRedo.h"
#include "Command/Backend/BaseCommand.h"
#include "Editor/Model/EditorModel.h"
#include "Editor/Model/WorldModel.h"
#include "Game/Model/GameModel.h"
#include "MainView/Model/MapModel.h"
#include "Network/NetworkModel.h"
```

Classes

- class **Processor**

Process commands and manages them accordingly.

9.189 /home/ryoga/git/pidgirl/pidgirl-engine/Command/UndoRedo.cpp File Reference

```
#include "UndoRedo.h"
#include <iostream>
```

9.190 /home/ryoga/git/pidgirl/pidgirl-engine/Command/UndoRedo.h File Reference

```
#include <QObject>
#include <QLinkedList>
#include "Command/Backend/BaseCommand.h"
```

Classes

- class **UndoRedo**

Manage the Undo and Redo functionalities.

Namespaces

- **PIDGIRL**

*The **PIDGIRL** (p. 37) namespace.*

Constant Groups

- **PIDGIRL**

*The **PIDGIRL** (p. 37) namespace.*

Variables

- const int **PIDGIRL::MAX_UNDO_REDO_SIZE** = 5

9.191 /home/ryoga/git/pidgirl/pidgirl-engine/Common/Common.cpp File Reference

```
#include "Common.h"
```

9.192 /home/ryoga/git/pidgirl/pidgirl-engine/Common/Common.h File Reference

```
#include <QCursor>
#include <QSize>
#include <QString>
#include <QtGlobal>
#include <QTime>
#include <iostream>
#include <climits>
```

Namespaces

- **PIDGIRL**

*The **PIDGIRL** (p. 37) namespace.*

Constant Groups

- **PIDGIRL**

*The **PIDGIRL** (p. 37) namespace.*

Functions

- void **PIDGIRL::showUsage** ()
*Show the usage of **PIDGIRL** (p. 37).*
- void **PIDGIRL::writeLog** (const QString &s)
Write a log output in the outputbuffer, so 6 characters are left to be written.
- void **PIDGIRL::writeLogDone** (const QString &s="[DONE]")
Write a certain string.
- void **PIDGIRL::writeLogFail** (const QString &error="")
Something failed and a given error is written after [FAIL] is written.
- void **PIDGIRL::newSeed** ()
Create a new seed for random numbers.
- unsigned int **PIDGIRL::nextID** (unsigned int id, const unsigned int lowestID=0)
Get the next higher possible ID.

Variables

- const QString **PIDGIRL::PIDGIRL_VERSION** = "0.5"
Version Number.
- const int **PIDGIRL::SECONDS_UNTIL_RETRYING_FILE_ACCESS** = 1
Specifies the milliseconds until a file access should be retried. 1 sec = 1 millisecond.
- const int **PIDGIRL::MAX_FILE_ACCESS_RETRIES** = 5
Specifies how many times a fileaccess should be retried until it's canceled.
- const int **PIDGIRL::MIN_HORIZONTAL_SQUARE_COUNT** = 3
The minimum horizontal square count.
- const int **PIDGIRL::MAX_HORIZONTAL_SQUARE_COUNT** = 30
The maximum horizontal square count.
- const int **PIDGIRL::STANDARD_HORIZONTAL_SQUARE_COUNT** = 10
The standard horizontal square count.
- const int **PIDGIRL::MIN_VERTICAL_SQUARE_COUNT** = 3
The minimum vertical square count.
- const int **PIDGIRL::MAX_VERTICAL_SQUARE_COUNT** = 20
The maximum vertical square count.
- const int **PIDGIRL::STANDARD_VERTICAL_SQUARE_COUNT** = 10
The standard vertical square count.
- const int **PIDGIRL::MIN_HEIGHT** = 1
The minimum height.
- const int **PIDGIRL::MAX_HEIGHT** = 30
The maximum height.
- const int **PIDGIRL::STANDARD_HEIGHT** = 1
The standard height.
- const int **PIDGIRL::MIN_ZOOM_FACTOR** = 1
The minimum zoomfactor for the square size.
- const int **PIDGIRL::MAX_ZOOM_FACTOR** = 3
The maximum zoomfactor for the square size.
- const int **PIDGIRL::STANDARD_ZOOM_FACTOR** = 1
The standard zoomfactor for the square size.
- const int **PIDGIRL::MIN_SQUARE_SIZE** = MIN_ZOOM_FACTOR*30
The minimum size of a square in SIZE x SIZE pixel.
- const int **PIDGIRL::MAX_SQUARE_SIZE** = MAX_ZOOM_FACTOR*MIN_SQUARE_SIZE
The maximum size of a square in SIZE x SIZE pixel.

- const int **PIDGIRL::STANDARD_SQUARE_SIZE** = MIN_SQUARE_SIZE
The standard size of a square in SIZE x SIZE pixel.
- const int **PIDGIRL::MIN_GRID_LINE_SIZE** = 1
The minimum size of a grid line.
- const int **PIDGIRL::MAX_GRID_LINE_SIZE** = 8
The maximum size of a grid line.
- const int **PIDGIRL::STANDARD_GRID_LINE_SIZE** = 1
The standard size of a grid line.
- const int **PIDGIRL::MIN_PLAYER_NAME_LENGTH** = 4
The minimum of characters that has to be used when naming a character.
- const int **PIDGIRL::MAX_PLAYER_NAME_LENGTH** = 30
The maximum of characters that can be used when naming a character.
- const QString **PIDGIRL::NAME_REG_EXP_STRING** = QString("(\\s\\w){%1,%2}").arg(PIDGIRL::MIN_PLAYER_NAME_LENGTH).arg(PIDGIRL::MAX_PLAYER_NAME_LENGTH)
Regular expression for names the player can enter.
- const QString **PIDGIRL::IP_ADDRESS_REG_EXP_STRING** = QString("[0-9][a-z][A-Z]([0-9]|-|.|[a-z][A-Z])*")
Regular experssion for IP Addresses. According to RFC952 and RFC1123.
- const int **PIDGIRL::SEND_DATA_RETRIES** = 3
The amount of retries before canceling the sending process.
- const unsigned int **PIDGIRL::SERVER_PLAYER_ID** = 0
Represents the player ID used by the server.
- const int **PIDGIRL::MAX_EMAIL_PLAYER_COUNT** = 128
The initial maximum amount of players in an E-Mail Game.

9.193 /home/ryoga/git/pidgirl/pidgirl-engine/Common/FileManager.cpp File Reference

```
#include "FileManager.h"
#include <QDir>
#include <QByteArray>
#include <QBuffer>
#include "Common/Common.h"
```

9.194 /home/ryoga/git/pidgirl/pidgirl-engine/Common/FileManager.h File Reference

```
#include <QFile>
#include <QFileInfo>
#include <QObject>
#include "Common/Serialize.h"
```

Classes

- class **FileManager**
Manages files to open and close accordingly and other things.

9.195 /home/ryoga/git/pidgirl/pidgirl-engine/Common/FileNameConventions.cpp File Reference

```
#include "FileNameConventions.h"
```

9.196 /home/ryoga/git/pidgirl/pidgirl-engine/Common/FileNameConventions.h File Reference

```
#include <QString>
```

Namespaces

- **PIDGIRL**

*The **PIDGIRL** (p. 37) namespace.*

Constant Groups

- **PIDGIRL**

*The **PIDGIRL** (p. 37) namespace.*

Variables

- const QString **PIDGIRL::MAP_SUFFIX** = ".tmap"
The map suffix.
- const QString **PIDGIRL::WORLD_FOLDER_NAME** = "worlds"
The World Folder Name.
- const QString **PIDGIRL::WORLD_SUFFIX** = ".tworld"
The world suffix.
- const QString **PIDGIRL::GAME_INFO_SUFFIX** = ".tgameinfo"
The game information suffix.
- const QString **PIDGIRL::SAVE_SUFFIX** = ".tsave"
The save suffix.
- const QString **PIDGIRL::SAVE_FOLDER_NAME** = "saves"
The Save Folder Name.
- const QString **PIDGIRL::EMAIL_SUFFIX** = ".tmail"
The email file suffix.
- const QString **PIDGIRL::EMAIL_FOLDER_NAME** = "email"
The EMail Folder Name.
- const QString **PIDGIRL::RSA_FOLDER_NAME** = "RSA-Encryption"
The RSA Folder Name.
- const QString **PIDGIRL::RSA_PRIVATE_KEY_FILE_NAME** = "private.key"
The RSA Private Key File Name.
- const QString **PIDGIRL::RSA_PUBLIC_KEY_FILE_NAME** = "public.key"
The RSA Public Key File Name.
- const QString **PIDGIRL::RSA_PLAINTEXT_FILE_NAME** = "plain.text"
The Plaintext File Name.

9.197 /home/ryoga/git/pidgirl/pidgirl-engine/Common/OptionsLayout.cpp File Reference

```
#include "OptionsLayout.h"  
#include <iostream>
```

9.198 /home/ryoga/git/pidgirl/pidgirl-engine/Common/OptionsLayout.h File Reference

```
#include <QListWidget>  
#include <QStackedLayout>  
#include <QGridLayout>  
#include <QPushButton>  
#include <QDragMoveEvent>
```

Classes

- class **List**
The list for widgets with reimplemented mouse events.
- class **OptionsLayout**
The layout for options to be chosen.

9.199 /home/ryoga/git/pidgirl/pidgirl-engine/Common/Serialize.cpp File Reference

```
#include "Serialize.h"
```

9.200 /home/ryoga/git/pidgirl/pidgirl-engine/Common/Serialize.h File Reference

```
#include <QIODevice>  
#include <QFileInfo>
```

Classes

- class **Serialize**
Interface for serializable classes.

9.201 /home/ryoga/git/pidgirl/pidgirl-engine/Common/YesNoCancelDialog.cpp File Reference

```
#include "YesNoCancelDialog.h"  
#include "Common/Common.h"
```

9.202 /home/ryoga/git/pidgirl/pidgirl-engine/Common/YesNoCancelDialog.h File Reference

```
#include <QMessageBox>
#include <QPushButton>
```

Classes

- class **YesNoCancelDialog**
A wrapper to show yes no cancel dialogs more easily.

9.203 /home/ryoga/git/pidgirl/pidgirl-engine/Editor/EditorCommon.h File Reference

Namespaces

- **PIDGIRL**
*The **PIDGIRL** (p. 37) namespace.*

Constant Groups

- **PIDGIRL**
*The **PIDGIRL** (p. 37) namespace.*

Variables

- const int **PIDGIRL::EDITOR_VIEW_WIDTH** = 200
The width of the Editor View.
- const int **PIDGIRL::EDITOR_VIEW_HEIGHT** = 250
The minimal height of the Editor View.
- const int **PIDGIRL::TOOL_BUTTON_COUNT_IN_ROW** = 4
The amount of tool buttons per row.

9.204 /home/ryoga/git/pidgirl/pidgirl-engine/Editor/EditorMenu.cpp File Reference

```
#include "Editor/EditorMenu.h"
#include <QKeySequence>
#include "Editor/EditorCommon.h"
#include "Command/Backend/UseRedo.h"
#include "Command/Backend/UseUndo.h"
#include "Command/Backend/Editor/World/NewWorld.h"
#include "Command/Backend/Editor/World/SaveWorld.h"
#include "Command/Backend/Editor/World/SaveWorldAs.h"
#include "Command/Backend/Editor/World/LoadWorld.h"
#include "Command/Backend/Common/Quit.h"
#include "Command/Backend/Editor/World/SaveMap.h"
#include "Command/Backend/Editor/World/SaveMapAs.h"
#include "Command/Backend/Editor/Map/ChangeZoomFactor.h"
#include "Command/Backend/ShowAbout.h"
```

9.205 /home/ryoga/git/pidgirl/pidgirl-engine/Editor/EditorMenu.h File Reference

```
#include <QEvent>
#include <QMenuBar>
#include <QMenu>
#include "Editor/Model/EditorModel.h"
#include "Editor/Model/WorldModel.h"
#include "Editor/MapPropertiesDialog.h"
#include "Editor/WorldPropertiesDialog.h"
#include "Command/Processor.h"
```

Classes

- class **EditorMenu**

Represents the menubar of the editor.

9.206 /home/ryoga/git/pidgirl/pidgirl-engine/Editor/EditorTools.cpp File Reference

```
#include "Editor/EditorTools.h"
#include "Editor/EditorCommon.h"
#include "Command/Backend/Editor/Tools/ChooseMainTool.h"
#include <QPoint>
```

9.207 /home/ryoga/git/pidgirl/pidgirl-engine/Editor/EditorTools.h File Reference

```
#include <QObject>
#include <QVector>
#include <QWidget>
#include <QGridLayout>
#include <QButtonGroup>
#include <QToolButton>
#include "Command/Processor.h"
#include "Editor/Model/EditorModel.h"
#include "Editor/EditorCommon.h"
```

Classes

- class **EditorTools**

The tools to be used in the editor.

9.208 /home/ryoga/git/pidgirl/pidgirl-engine/Editor/MapPropertiesDialog.cpp File Reference

```
#include "MapPropertiesDialog.h"
#include <QLabel>
#include "Common/Common.h"
#include "Command/Backend/Editor/Map/ApplyMapGeometryChanges.h"
```


9.209 /home/ryoga/git/pidgirl/pidgirl-engine/Editor/MapPropertiesDialog.h File Reference

```
#include <QDialog>
#include <QAction>
#include <QSpinBox>
#include <QComboBox>
#include <QPushButton>
#include "Command/Processor.h"
#include "Common/OptionsLayout.h"
```

Classes

- class **MapPropertiesDialog**

9.210 /home/ryoga/git/pidgirl/pidgirl-engine/Editor/Model/EditorModel.cpp File Reference

```
#include <QToolButton>
#include "Editor/Model/EditorModel.h"
```

9.211 /home/ryoga/git/pidgirl/pidgirl-engine/Editor/Model/EditorModel.h File Reference

```
#include <QObject>
#include "Object/ObjectBase.h"
```

Classes

- class **EditorModel**
Represents the editor state.

Namespaces

- **PIDGIRL**
*The **PIDGIRL** (p. 37) namespace.*

Constant Groups

- **PIDGIRL**
*The **PIDGIRL** (p. 37) namespace.*

Enumerations

- enum **PIDGIRL::MainToolButton** { **PIDGIRL::START_POSITION** =0, **PIDGIRL::BRICKWALL** }
Represents every maintool button.

9.212 /home/ryoga/git/pidgirl/pidgirl-engine/Editor/Model/LocalFilePaths.cpp File Reference

```
#include "LocalFilePaths.h"
```

9.213 /home/ryoga/git/pidgirl/pidgirl-engine/Editor/Model/LocalFilePaths.h File Reference

```
#include <QString>
#include <QVector>
```

Classes

- class **LocalFilePaths**

Locally in world folder represented file paths.

9.214 /home/ryoga/git/pidgirl/pidgirl-engine/Editor/Model/WorldModel.cpp File Reference

```
#include "Editor/Model/WorldModel.h"
```

9.215 /home/ryoga/git/pidgirl/pidgirl-engine/Editor/Model/WorldModel.h File Reference

```
#include <QFileSystemModel>
#include <QString>
#include <QFile>
#include <QDataStream>
#include "Editor/Model/LocalFilePaths.h"
#include "Common/Serialize.h"
#include "Object/ObjectID.h"
```

Classes

- class **WorldModel**

Represents the world and all of its settings.

9.216 /home/ryoga/git/pidgirl/pidgirl-engine/Editor/WorldFileTree.cpp File Reference

```
#include "WorldFileTree.h"
#include <QMouseEvent>
#include <QModelIndexList>
#include <QApplication>
#include <QPoint>
#include <QSize>
#include <QHeaderView>
#include "Command/Backend/Editor/Map/LoadMapByIndex.h"
#include "Command/Backend/Editor/World/NewMapFile.h"
#include "Command/Backend/Editor/World/NewFolder.h"
#include "Command/Backend/Editor/World/RenameWorldEntry.h"
#include "Command/Backend/Editor/World/DuplicateWorldEntry.h"
#include "Command/Backend/Editor/World/RemoveWorldEntry.h"
#include "Command/Backend/Editor/World/MoveFile.h"
```

9.217 /home/ryoga/git/pidgirl/pidgirl-engine/Editor/WorldFileTree.h File Reference

```
#include <QTreeView>
#include <QMenu>
#include <QAction>
#include "Command/Processor.h"
#include "Editor/Model/WorldModel.h"
```

Classes

- class **WorldFileTree**

Showing the FileTree of an opened world.

9.218 /home/ryoga/git/pidgirl/pidgirl-engine/Editor/WorldPropertiesDialog.cpp File Reference

```
#include "WorldPropertiesDialog.h"
#include <QLabel>
#include <QHeaderView>
#include "Common/FileNameConventions.h"
#include "Command/Backend/Editor/World/AddToChosenList.h"
#include "Command/Backend/Editor/World/RemoveFromChosenList.h"
#include "Command/Backend/Editor/World/ApplyWorldPropertiesChanges.h"
```

9.219 /home/ryoga/git/pidgirl/pidgirl-engine/Editor/WorldPropertiesDialog.h File Reference

```
#include <QDialog>
#include <QTreeView>
#include <QListView>
#include <QPushButton>
#include <QFileSystemModel>
#include <QStandardItemModel>
#include "Command/Processor.h"
#include "Editor/Model/WorldModel.h"
#include "Common/OptionsLayout.h"
```

Classes

- class **WorldPropertiesDialog**

Shows the properties of a world to change it.

9.220 /home/ryoga/git/pidgirl/pidgirl-engine/Editor/WorldView.cpp File Reference

```
#include "WorldView.h"
#include <QStringList>
#include <QMouseEvent>
#include <QBoxLayout>
#include "Editor/EditorCommon.h"
#include "Common/FileNameConventions.h"
```

9.221 /home/ryoga/git/pidgirl/pidgirl-engine/Editor/WorldView.h File Reference

```
#include <QWidget>
#include "Command/Processor.h"
#include "Editor/WorldFileTree.h"
#include "Editor/Model/WorldModel.h"
```

Classes

- class **WorldView**

The widget of the world Management.

9.222 /home/ryoga/git/pidgirl/pidgirl-engine/Game/EMailSetupWidget.cpp File Reference

```
#include "EMailSetupWidget.h"
#include "Common/Common.h"
```

9.223 /home/ryoga/git/pidgirl/pidgirl-engine/Game/EMailSetupWidget.h File Reference

```
#include <QWidget>
#include <QGridLayout>
#include <QPushButton>
#include <QSpinBox>
#include <QLabel>
```

Classes

- class **EMailSetupWidget**

The widget to set up an E-Mail Game.

9.224 /home/ryoga/git/pidgirl/pidgirl-engine/Game/GeneralSetupWidget.cpp File Reference

```
#include "GeneralSetupWidget.h"
```

9.225 /home/ryoga/git/pidgirl/pidgirl-engine/Game/GeneralSetupWidget.h File Reference

```
#include <QWidget>
#include <QGridLayout>
#include <QCheckBox>
#include "Editor/Model/WorldModel.h"
```

Classes

- class **GeneralSetupWidget**

The widget to set up general options for every game before it begins.

9.226 /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/ClientGameModel.cpp File Reference

```
#include "ClientGameModel.h"
#include "Common/Common.h"
#include "Command/Backend/Game/Ingame/FirstTurn.h"
#include "Command/Backend/Game/Ingame/YourTurnInformation.h"
#include "Command/Backend/Network/Client/RequestMapFromServer.h"
#include "Command/Backend/Network/Client/SendSpawnCharacterToServer.h"
#include "Command/Backend/Network/Client/SendUpdateCharacterToServer.h"
#include "Command/Backend/Network/Client/SendToServerEndTurn.h"
#include "Command/Backend/Network/Client/RequestPlayerModelsFromServer.h"
#include "Command/Backend/Network/Client/RequestPlayerFromServer.h"
#include "Command/Backend/Network/Client/ChangePlayerIDOnServer.h"
```

9.227 /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/ClientGameModel.h File Reference

```
#include <QVariant>
#include "Game/Model/GameModel.h"
#include "Command/Processor.h"
#include "Network/ClientModel.h"
```

Classes

- class **ClientGameModel**

The model representing a game ran as a client where most information is accessible remotely.

9.228 /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/EMailGameModel.cpp File Reference

```
#include "EMailGameModel.h"
#include <QMessageBox>
#include "Command/Backend/Game/Ingame/SaveEMailGame.h"
#include "Command/Backend/Game/Ingame/YourTurnInformation.h"
#include "Command/Backend/Game/Ingame/FirstTurn.h"
#include "Command/Backend/Common/Quit.h"
```

9.229 /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/EMailGameModel.h File Reference

```
#include "Game/Model/LocalGameModel.h"
#include "Game/Model/MailGameInfo.h"
```

Classes

- class **EMailGameModel**

The model representing an E-Mail game.

9.230 /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/GameModel.cpp File Reference

```
#include "GameModel.h"
#include <QDebug>
#include "Object/ObjectFromID.h"
```

9.231 /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/GameModel.h File Reference

```
#include <QObject>
#include "Game/Model/PlayerModel.h"
#include "Common/Serialize.h"
#include "MainView/Model/MapModel.h"
#include "Object/Character/Player.h"
```

Classes

- class **GameModel**
The model representing the game preferences and players.

Namespaces

- **PIDGIRL**
*The **PIDGIRL** (p. 37) namespace.*

Constant Groups

- **PIDGIRL**
*The **PIDGIRL** (p. 37) namespace.*

Enumerations

- enum **PIDGIRL::GameType** { **PIDGIRL::HOTSEAT** = 0, **PIDGIRL::NETWORK**, **PIDGIRL::EMAIL** }

9.232 /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/HotseatGameModel.cpp File Reference

```
#include "HotseatGameModel.h"
#include <QMessageBox>
#include "Command/Backend/Game/Ingame/FirstTurn.h"
#include "Command/Backend/Game/Ingame/YourTurnInformation.h"
#include "Command/Backend/Common/Quit.h"
#include "Game/Model/ServerGameModel.h"
```

9.233 /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/HotseatGameModel.h File Reference

```
#include "Game/Model/LocalGameModel.h"
```

Classes

- class **HotseatGameModel**
The model representing a hotseat game with one or more players.

9.234 /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/LocalGameModel.cpp File Reference

```
#include "LocalGameModel.h"
```

9.235 /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/LocalGameModel.h File Reference

```
#include <QMap>
#include "Command/Processor.h"
#include "Game/Model/GameModel.h"
```

Classes

- class **LocalGameModel**

The model representing a local game where all the information is held locally.

9.236 /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/MailGameInfo.cpp File Reference

```
#include "MailGameInfo.h"
#include <QFile>
#include <QTextStream>
#include <QDebug>
#include "Common/FileNameConventions.h"
#include "Common/Common.h"
#include "RSA/RSAMethod.h"
```

9.237 /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/MailGameInfo.h File Reference

```
#include <QObject>
#include <QDir>
#include <QDataStream>
#include <string>
#include "Common/Serialize.h"
```

Classes

- class **MailGameInfo**

Holds all the informations about your E-Mail Game.

9.238 /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/PlayerModel.cpp File Reference

```
#include "PlayerModel.h"  
#include <QDataStream>
```

9.239 /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/PlayerModel.h File Reference

```
#include <QObject>  
#include <QString>  
#include <QDataStream>  
#include "Common/Serialize.h"
```

Classes

- class **PlayerModel**

This model represents the informations about a specific player.

9.240 /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/SaveGameFile.cpp File Reference

```
#include "SaveGameFile.h"  
#include <QDataStream>  
#include "Command/Backend/Common/Quit.h"  
#include "Game/Model/ServerGameModel.h"  
#include "Game/Model/HotseatGameModel.h"
```

9.241 /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/SaveGameFile.h File Reference

```
#include <QObject>  
#include "Common/Serialize.h"  
#include "Command/Processor.h"  
#include "Editor/Model/WorldModel.h"  
#include "Game/Model/GameModel.h"
```

Classes

- class **SaveGameFile**

*Represents everything needed to **Serialize** (p. 313) and **Deserialize** a complete game save file.*

9.242 /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/ServerGameModel.cpp File Reference

```
#include "ServerGameModel.h"
#include <QMessageBox>
#include "Command/Backend/Game/Ingame/FirstTurn.h"
#include "Command/Backend/Game/Ingame/YourTurnInformation.h"
#include "Command/Backend/Network/Server/SendToClientNextTurn.h"
#include "Command/Backend/Network/Server/Broadcast/BroadcastSpawnCharacter.h"
#include "Command/Backend/Network/Server/Broadcast/BroadcastUpdateCharacter.h"
#include "Command/Backend/Common/Quit.h"
```

9.243 /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/ServerGameModel.h File Reference

```
#include "Game/Model/LocalGameModel.h"
```

Classes

- class **ServerGameModel**

The model representing a game that is also a server.

9.244 /home/ryoga/git/pidgirl/pidgirl-engine/Game/NetworkSetupWidget.cpp File Reference

```
#include "NetworkSetupWidget.h"
```

9.245 /home/ryoga/git/pidgirl/pidgirl-engine/Game/NetworkSetupWidget.h File Reference

```
#include <QWidget>
#include <QGridLayout>
#include <QLabel>
#include <QLineEdit>
#include <QIntValidator>
#include <QPushButton>
```

Classes

- class **NetworkSetupWidget**

The widget to set up an Network Game.

9.246 /home/ryoga/git/pidgirl/pidgirl-engine/Game/SoloHotseatSetupWidget.cpp File Reference

```
#include "SoloHotseatSetupWidget.h"
```

9.247 /home/ryoga/git/pidgirl/pidgirl-engine/Game/SoloHotseatSetupWidget.h File Reference

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

Classes

- class **SoloHotseatSetupWidget**

The widget to set up an Solo / Hotseat Game.

9.248 /home/ryoga/git/pidgirl/pidgirl-engine/main.cpp File Reference

```
#include <QApplication>
#include <QString>
#include <QTranslator>
#include <QCoreApplication>
#include <QLocale>
#include "MainWindow.h"
#include "Common/Common.h"
```

Functions

- const QString & **checkLocale** (const QString &locale)
- int **main** (int argc, char *argv[])

9.248.1 Function Documentation

9.248.1.1 const QString& **checkLocale** (const QString & *locale*)

9.248.1.2 int **main** (int *argc*, char * *argv*[])

9.249 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/CommandAction.cpp File Reference

```
#include "CommandAction.h"
```

9.250 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/CommandAction.h File Reference

```
#include <QAction>
#include <QIcon>
```

Classes

- class **CommandAction**

An action that will trigger a command.

9.251 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/MainViewMap.cpp File Reference

```
#include "MainViewMap.h"
#include <QPainter>
#include <QMessageBox>
#include "MainView/Model/MouseInput/MouseGameInput.h"
#include "MainView/Model/MouseInput/MouseEditorInput.h"
#include <iostream>
```

9.252 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/MainViewMap.h File Reference

```
#include <QWidget>
#include "Command/Processor.h"
#include "Editor/Model/EditorModel.h"
#include "MainView/MainViewStack.h"
#include "MainView/Model/StackModel.h"
#include "MainView/Model/MapModel.h"
#include "MainView/Model/MouseInput/MouseInput.h"
#include "Game/Model/GameModel.h"
```

Classes

- class **MainViewMap**

The main view showing every stack on an grid.

9.253 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/MainViewStack.cpp File Reference

```
#include "MainViewStack.h"
#include <QPainter>
#include <QPoint>
#include <QApplication>
#include <iostream>
#include "MainView/ObjectMenu.h"
#include "Game/Model/GameModel.h"
```

9.254 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/MainViewStack.h File Reference

```
#include <QWidget>
#include <QMenu>
#include "MainView/Model/StackModel.h"
#include "MainView/Model/MapModel.h"
#include "MainView/Model/MouseInput/MouseInput.h"
#include "Editor/Model/EditorModel.h"
#include "Command/Processor.h"
```

Classes

- class **MainViewStack**

Graphically represents a stack of a square in the game.

9.255 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MapModel.cpp File Reference

```
#include "MainView/Model/MapModel.h"
#include <QDataStream>
#include <QDebug>
#include "Common/Common.h"
#include "Object/Character/Player.h"
#include <iostream>
```

9.256 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MapModel.h File Reference

```
#include <QSize>
#include <QObject>
#include <QPoint>
#include <QFileInfo>
#include "MainView/Model/StackModel.h"
#include "Object/ObjectBase.h"
#include "Common/Serialize.h"
#include "AStar/AStarMapEntry.h"
```

Classes

- class **MapModel**

Represents the map.

9.257 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ContextEditorMenuState.cpp File Reference

```
#include "ContextEditorMenuState.h"
#include "MainView/Model/MouseInput/PlaceObjectsState.h"
```

9.258 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ContextEditor-MenuState.h File Reference

```
#include "MainView/Model/MouseInput/MouseState.h"
```

Classes

- class **ContextEditorMenuState**

The state in which to show the editor context menu.

9.259 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ContextGame-MenuState.cpp File Reference

```
#include "ContextGameMenuState.h"  
#include "Command/Backend/Game/Map/FindPath.h"  
#include "MainView/Model/MouseInput/IdleGameMouseState.h"  
#include "MainView/Model/MouseInput/TargetChosenState.h"
```

9.260 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ContextGame-MenuState.h File Reference

```
#include "MainView/Model/MouseInput/MouseState.h"
```

Classes

- class **ContextGameMenuState**

The state in which to show the game menu.

9.261 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/IdleEditor-MouseState.cpp File Reference

```
#include "IdleEditorMouseState.h"  
#include "MainView/Model/MouseInput/PlaceObjectsState.h"  
#include "MainView/Model/MouseInput/RemoveObjectsState.h"  
#include "MainView/Model/MouseInput/ContextEditorMenuState.h"
```

9.262 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/IdleEditor-MouseState.h File Reference

```
#include "MainView/Model/MouseInput/MouseState.h"
```

Classes

- class **IdleEditorMouseState**

The editor mouse state when nothing is being done.

9.263 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/IdleGame-MouseState.cpp File Reference

```
#include "IdleGameMouseState.h"
#include "Command/Backend/Game/Map/FindPath.h"
#include "MainView/Model/MouseInput/TargetChosenState.h"
#include "MainView/Model/MouseInput/ContextGameMenuState.h"
```

9.264 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/IdleGame-MouseState.h File Reference

```
#include <QMouseEvent>
#include "MainView/Model/MouseInput/MouseState.h"
#include "MainView/Model/StackModel.h"
```

Classes

- class **IdleGameMouseState**

The game mouse state when nothing is being done.

9.265 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/MouseEditor-Input.cpp File Reference

```
#include "MouseEditorInput.h"
#include "MainView/Model/MouseInput/IdleEditorMouseState.h"
#include "Command/Backend/Editor/Map/CopyWall.h"
#include "Command/Backend/Editor/Map/CutWall.h"
#include "Command/Backend/Editor/Map/RemoveWall.h"
#include "Command/Backend/Editor/Map/PasteObject.h"
#include "Command/Backend/Editor/Map/RemoveAllObjects.h"
```

9.266 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/MouseEditor-Input.h File Reference

```
#include <QMenu>
#include <QAction>
#include "MainView/Model/MouseInput/MouseInput.h"
```

Classes

- class **MouseEditorInput**

The mouse input manager for the editor.

9.267 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/MouseGame-Input.cpp File Reference

```
#include "MouseGameInput.h"
#include "MainView/Model/MouseInput/IdleGameMouseState.h"
#include "MainView/ObjectMenu.h"
#include "Command/Backend/Game/Ingame/EndTurn.h"
#include "Command/Backend/Game/Ingame/ShowGameOptions.h"
```

9.268 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/MouseGame-Input.h File Reference

```
#include "MainView/Model/MouseInput/MouseInput.h"
```

Classes

- class **MouseGameInput**

The mouse input manager for the game.

9.269 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/MouseInput.cpp File Reference

```
#include "MouseInput.h"
```

9.270 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/MouseInput.h File Reference

```
#include <QObject>
#include <QMouseEvent>
#include <QMenu>
#include "Editor/Model/EditorModel.h"
#include "Command/Processor.h"
#include "MainView/Model/MouseInput/MouseState.h"
#include "MainView/Model/StackModel.h"
#include "MainView/Model/MapModel.h"
#include "Game/Model/GameModel.h"
```


Classes

- class **MouseInput**

The base class that manages the flow of the events to be processed next.

9.271 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/MouseState.cpp File Reference

```
#include "MouseState.h"
```

9.272 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/MouseState.h File Reference

```
#include <QMouseEvent>
#include "MainView/Model/StackModel.h"
#include "Command/Processor.h"
```

Classes

- class **MouseState**

The interface used for any state concerning mouse inputs.

9.273 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/PlaceObjects-State.cpp File Reference

```
#include "PlaceObjectsState.h"
#include "MainView/Model/MouseInput/IdleEditorMouseState.h"
#include "Command/Backend/Editor/Map/PlaceEditorObject.h"
```

9.274 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/PlaceObjects-State.h File Reference

```
#include <QSet>
#include "MainView/Model/MouseInput/MouseState.h"
```

Classes

- class **PlaceObjectsState**

The state in which objects will be placed.

9.275 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/RemoveObjects-State.cpp File Reference

```
#include "RemoveObjectsState.h"
#include "MainView/Model/MouseInput/IdleEditorMouseState.h"
#include "Command/Backend/Editor/Map/RemoveAllObjects.h"
```

9.276 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/RemoveObjects-State.h File Reference

```
#include <QSet>
#include "MainView/Model/MouseInput/MouseState.h"
```

Classes

- class **RemoveObjectsState**
Remove objects from stacks.

9.277 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/TargetChosen-State.cpp File Reference

```
#include "TargetChosenState.h"
#include <QWidget>
#include <QApplication>
#include "Command/Backend/Game/Map/FindPath.h"
#include "MainView/Model/MouseInput/TargetConfirmedState.h"
```

9.278 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/TargetChosen-State.h File Reference

```
#include "MainView/Model/MouseInput/MouseState.h"
#include "MainView/Model/StackModel.h"
```

Classes

- class **TargetChosenState**
The target is chosen.

9.279 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/TargetConfirmedState.cpp File Reference

```
#include "TargetConfirmedState.h"
#include "Command/Backend/Game/Map/ClearPath.h"
#include "Command/Backend/Game/Map/FindPath.h"
#include "MainView/Model/MouseInput/TargetChosenState.h"
#include "MainView/Model/MouseInput/TargetMoveState.h"
#include "MainView/Model/MouseInput/ContextGameMenuState.h"
#include "MainView/Model/MouseInput/IdleGameMouseState.h"
```

9.280 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/TargetConfirmedState.h File Reference

```
#include "MainView/Model/MouseInput/MouseState.h"
#include "MainView/Model/StackModel.h"
```

Classes

- class **TargetConfirmedState**

The chosen state is now confirmed.

9.281 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/TargetMoveState.cpp File Reference

```
#include "TargetMoveState.h"
#include "MainView/Model/MouseInput/IdleGameMouseState.h"
#include "MainView/Model/MouseInput/TargetConfirmedState.h"
#include "Command/Backend/Game/Map/MoveCharacter.h"
```

9.282 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/TargetMoveState.h File Reference

```
#include "MainView/Model/MouseInput/MouseState.h"
#include "MainView/Model/StackModel.h"
```

Classes

- class **TargetMoveState**

Move to the confirmed target.

9.283 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/StackModel.cpp File Reference

```
#include "StackModel.h"
#include <iostream>
#include "Object/Character/Player.h"
#include "Object/ObjectFromID.h"
```

9.284 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/StackModel.h File Reference

```
#include <QObject>
#include <QVector>
#include <QList>
#include <QDataStream>
#include <QAction>
#include <QMouseEvent>
#include "Object/ObjectItem.h"
#include "Object/ObjectCharacter.h"
#include "Object/ObjectTile.h"
#include "Object/ObjectSpecial.h"
```

Classes

- class **StackModel**

A single stack (square) holding a stack of items and informations about the square which holds them.

9.285 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/ObjectMenu.cpp File Reference

```
#include "ObjectMenu.h"
#include <QAction>
#include "Command/Backend/BaseCommand.h"
#include "MainView/CommandAction.h"
```

9.286 /home/ryoga/git/pidgirl/pidgirl-engine/MainView/ObjectMenu.h File Reference

```
#include <QMenu>
#include "Command/Processor.h"
#include "Object/ObjectBase.h"
```

Classes

- class **ObjectMenu**

The menu which is created by the command actions created through the given object.

9.287 /home/ryoga/git/pidgirl/pidgirl-engine/MainWindow.cpp File Reference

```
#include "MainWindow.h"
#include <QCoreApplication>
#include <QFileInfo>
#include <QDockWidget>
#include <QMessageBox>
#include <QPushButton>
#include <QFileDialog>
#include <QString>
#include "Command/Backend/Editor/World/NewWorld.h"
#include "Command/Backend/Editor/World/LoadWorld.h"
#include "Command/Backend/Editor/World/SaveWorld.h"
#include "Command/Backend/Common/Quit.h"
#include "Command/Backend/Game/Startmenu/LoadGame.h"
#include "Command/Backend/Game/Startmenu/SetupNewGame.h"
#include "Command/Backend/Game/Startmenu/ResumeEMailGame.h"
#include "Command/Backend/Game/Startmenu/ConnectTo.h"
#include "Game/Model/ClientGameModel.h"
#include "Game/Model/EMailGameModel.h"
#include "Common/YesNoCancelDialog.h"
#include "Common/Common.h"
#include <iostream>
```

Namespaces

- **PIDGIRL**

The **PIDGIRL** (p. 37) namespace.

Constant Groups

- **PIDGIRL**

The **PIDGIRL** (p. 37) namespace.

Variables

- bool **PIDGIRL::done** = false

Needed for canceling the program before the application has been started yet (Before `app.exec()` in **main()** (p. 435)).
Only set with the **Quit** (p. 261) **Command** (p. 93).

9.288 /home/ryoga/git/pidgirl/pidgirl-engine/MainWindow.h File Reference

```
#include <QMainWindow>
#include "Editor/EditorTools.h"
#include "Editor/EditorMenu.h"
#include "Editor/WorldView.h"
#include "Editor/Model/EditorModel.h"
#include "Editor/Model/WorldModel.h"
#include "Game/Model/GameModel.h"
#include "MainView/MainViewMap.h"
#include "MainView/Model/MapModel.h"
#include "Command/Processor.h"
#include "Common/Common.h"
```

Classes

- class **MainWindow**

Holds all the models and GUIs and initializes them.

Namespaces

- **PIDGIRL**

*The **PIDGIRL** (p. 37) namespace.*

Constant Groups

- **PIDGIRL**

*The **PIDGIRL** (p. 37) namespace.*

9.289 /home/ryoga/git/pidgirl/pidgirl-engine/Network/ClientModel.cpp File Reference

```
#include "ClientModel.h"
#include <QMessageBox>
#include <QDataStream>
#include <QHostAddress>
#include <QBuffer>
#include "Common/Common.h"
#include "Command/Backend/Network/Client/ReceiveInitialPlayerInformation.h"
#include "Command/Backend/Network/Client/ReceiveMap.h"
#include "Command/Backend/Network/Client/ReceivePlayer.h"
#include "Command/Backend/Network/Client/ReceivePlayerModels.h"
#include "Command/Backend/Network/ReceiveSpawnCharacterFrom.h"
#include "Command/Backend/Network/ReceiveUpdateCharacterFrom.h"
#include "Command/Backend/Game/Ingame/NextTurn.h"
#include "Command/Backend/Common/Quit.h"
```

9.290 /home/ryoga/git/pidgirl/pidgirl-engine/Network/ClientModel.h File Reference

```
#include <QTcpSocket>
#include <QString>
#include <QDataStream>
#include "Command/Processor.h"
#include "Network/NetworkModel.h"
```

Classes

- class **ClientModel**

The client side implementation of the network model.

9.291 /home/ryoga/git/pidgirl/pidgirl-engine/Network/ClientSocket.cpp File Reference

```
#include "ClientSocket.h"
```

9.292 /home/ryoga/git/pidgirl/pidgirl-engine/Network/ClientSocket.h File Reference

```
#include <QTcpSocket>
```

Classes

- class **ClientSocket**

9.293 /home/ryoga/git/pidgirl/pidgirl-engine/Network/NetworkModel.cpp File Reference

```
#include "NetworkModel.h"
```

9.294 /home/ryoga/git/pidgirl/pidgirl-engine/Network/NetworkModel.h File Reference

```
#include <QObject>
#include <QString>
#include <QTcpSocket>
#include "Common/Serialize.h"
```

Classes

- class **NetworkModel**

The network model managing connections and interpretations of incoming messages to be converted to commands.

Enumerations

- enum **FromServer** {
RECEIVE_INITIAL_PLAYER_INFORMATION = 0, SEND_SPECIFIC_MAP, SEND_SPECIFIC_PLAYER,
SEND_PLAYER_MODELS,
FROM_SERVER_NEXT_TURN, BROADCAST_SPAWN_CHARACTER, BROADCAST_UPDATE_CHARACTER }
The Enum for messages received from the server.
- enum **FromClient** {
RECEIVE_REQUEST_FOR_SPECIFIC_MAP = 0, RECEIVE_REQUEST_FOR_PLAYER_MODELS, RECEIVE_REQUEST_FOR_PLAYER, END_TURN,
FROM_CLIENT_NEXT_TURN, SEND_SPAWN_CHARACTER, SEND_UPDATE_CHARACTER, CHANGE_PLAYER_ID }
The Enum for messages received from the client.

9.294.1 Enumeration Type Documentation

9.294.1.1 enum FromClient

The Enum for messages received from the client.

Enumerator

RECEIVE_REQUEST_FOR_SPECIFIC_MAP
RECEIVE_REQUEST_FOR_PLAYER_MODELS
RECEIVE_REQUEST_FOR_PLAYER
END_TURN
FROM_CLIENT_NEXT_TURN
SEND_SPAWN_CHARACTER
SEND_UPDATE_CHARACTER
CHANGE_PLAYER_ID

9.294.1.2 enum FromServer

The Enum for messages received from the server.

Enumerator

RECEIVE_INITIAL_PLAYER_INFORMATION
SEND_SPECIFIC_MAP
SEND_SPECIFIC_PLAYER
SEND_PLAYER_MODELS
FROM_SERVER_NEXT_TURN
BROADCAST_SPAWN_CHARACTER
BROADCAST_UPDATE_CHARACTER

9.295 /home/ryoga/git/pidgirl/pidgirl-engine/Network/ServerModel.cpp File Reference

```
#include "ServerModel.h"
#include <QMessageBox>
#include <QList>
#include <QDataStream>
#include <QBuffer>
#include <climits>
#include "Common/Common.h"
#include "Command/Backend/Network/Server/ClientDisconnected.h"
#include "Command/Backend/Network/Server/SendInitialPlayerInformation.h"
#include "Command/Backend/Network/Server/ReplyToRequestForMap.h"
#include "Command/Backend/Network/Server/ReplyToRequestForPlayer.h"
#include "Command/Backend/Network/Server/ReplyToRequestForPlayerModels.h"
#include "Command/Backend/Network/Server/ReceiveChangePlayerID.h"
#include "Command/Backend/Network/ReceiveSpawnCharacterFrom.h"
#include "Command/Backend/Network/ReceiveUpdateCharacterFrom.h"
#include "Command/Backend/Game/Ingame/NextTurn.h"
```


9.296 /home/ryoga/git/pidgirl/pidgirl-engine/Network/ServerModel.h File Reference

```
#include <QTcpServer>
#include <QMap>
#include "Network/NetworkModel.h"
#include "Network/ClientSocket.h"
#include "Command/Processor.h"
#include "Editor/Model/WorldModel.h"
```

Classes

- class **ServerModel**

9.297 /home/ryoga/git/pidgirl/pidgirl-engine/Network/WaitingForDataDialog.cpp File Reference

```
#include "WaitingForDataDialog.h"
```

9.298 /home/ryoga/git/pidgirl/pidgirl-engine/Network/WaitingForDataDialog.h File Reference

```
#include <QObject>
#include <QMessageBox>
#include "Network/NetworkModel.h"
```

Classes

- class **WaitingForDataDialog**

9.299 /home/ryoga/git/pidgirl/pidgirl-engine/Object/Character/Player.cpp File Reference

```
#include "Player.h"
#include <QGridLayout>
#include <QLabel>
#include "Command/Backend/Object/ShowDescription.h"
```

9.300 /home/ryoga/git/pidgirl/pidgirl-engine/Object/Character/Player.h File Reference

```
#include "Object/ObjectCharacter.h"
```

Classes

- class **Player**

The player object representing the player.

9.301 /home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectBase.cpp File Reference

```
#include <iostream>
#include "Object/ObjectBase.h"
#include <climits>
```

9.302 /home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectBase.h File Reference

```
#include <QObject>
#include <QString>
#include <QVector>
#include <QDataStream>
#include <QImage>
#include "Object/ObjectImage.h"
#include "Object/ObjectID.h"
#include "Command/Backend/BaseCommand.h"
#include "Common/Serialize.h"
```

Classes

- class **ObjectBase**

The baseclass for the Objects to be placed into the game's mainView.

9.303 /home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectCharacter.cpp File Reference

```
#include "Object/ObjectCharacter.h"
```

9.304 /home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectCharacter.h File Reference

```
#include "Object/ObjectBase.h"
#include "Object/ObjectID.h"
```

Classes

- class **ObjectCharacter**

The baseclass for the Characters to be placed into the game's mainView.

9.305 /home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectFromID.cpp File Reference

```
#include "ObjectFromID.h"
#include "Object/Character/Player.h"
#include "Object/Special/StartPosition.h"
#include "Object/Tile/BrickWall.h"
```

9.306 /home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectFromID.h File Reference

```
#include "Object/ObjectID.h"
#include "Object/ObjectCharacter.h"
#include "Object/ObjectItem.h"
#include "Object/ObjectTile.h"
#include "Object/ObjectSpecial.h"
```

Namespaces

- **ObjectFromID**

Constant Groups

- **ObjectFromID**

Functions

- **ObjectCharacter * ObjectFromID::objectFrom (ObjectID::CharacterID id, QObject *parent=0)**
Create object from an enum.
- **ObjectItem * ObjectFromID::objectFrom (ObjectID::ItemID id, QObject *parent=0)**
- **ObjectSpecial * ObjectFromID::objectFrom (ObjectID::SpecialID id, QObject *parent=0)**
- **ObjectTile * ObjectFromID::objectFrom (ObjectID::TileID id, QObject *parent=0)**

9.307 /home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectID.cpp File Reference

```
#include "ObjectID.h"
#include "Common/Common.h"
#include <QDataStream>
```

9.308 /home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectID.h File Reference

```
#include <QMap>
```

Classes

- class **ObjectID**

Information and management of spawnIDs for the corresponding ObjectIDs.

9.309 /home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectImage.cpp File Reference

```
#include "Object/ObjectImage.h"
```

9.310 /home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectImage.h File Reference

```
#include <QImage>
#include <QString>
```

Classes

- class **ObjectImage**

Representing and QImage which knows the path it has been loaded from.

9.311 /home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectItem.cpp File Reference

```
#include "Object/ObjectItem.h"
```

9.312 /home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectItem.h File Reference

```
#include "Object/ObjectBase.h"
#include "Object/ObjectID.h"
```

Classes

- class **ObjectItem**

The baseclass for the Items to be placed into the game's mainView.

9.313 /home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectSpecial.cpp File Reference

```
#include "ObjectSpecial.h"
```

9.314 /home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectSpecial.h File Reference

```
#include "Object/ObjectID.h"
#include "Object/ObjectBase.h"
```

Classes

- class **ObjectSpecial**

9.315 /home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectTile.cpp File Reference

```
#include "Object/ObjectTile.h"  
#include "Object/ObjectID.h"
```

9.316 /home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectTile.h File Reference

```
#include "Object/ObjectBase.h"  
#include "Object/ObjectID.h"
```

Classes

- class **ObjectTile**

Represents a tile to be put into the mainView.

9.317 /home/ryoga/git/pidgirl/pidgirl-engine/Object/Special/StartPosition.cpp File Reference

```
#include "StartPosition.h"  
#include <QGridLayout>  
#include <QLabel>  
#include "Command/Backend/Object/ShowDescription.h"
```

9.318 /home/ryoga/git/pidgirl/pidgirl-engine/Object/Special/StartPosition.h File Reference

```
#include "Object/ObjectSpecial.h"  
#include "Object/ObjectID.h"
```

Classes

- class **StartPosition**

The start position of the player.

9.319 /home/ryoga/git/pidgirl/pidgirl-engine/Object/Tile/BrickWall.cpp File Reference

```
#include "BrickWall.h"  
#include <QGridLayout>  
#include <QLabel>  
#include "Command/Backend/Object/ShowDescription.h"
```

9.320 /home/ryoga/git/pidgirl/pidgirl-engine/Object/Tile/BrickWall.h File Reference

```
#include "Object/ObjectTile.h"  
#include "Object/ObjectID.h"
```

Classes

- class **BrickWall**
A wall of bricks.

9.321 /home/ryoga/git/pidgirl/pidgirl-engine/RSA/RSAMethod.cpp File Reference

```
#include "RSAMethod.h"  
#include <cryptopp/dsa.h>  
#include <cryptopp/base64.h>  
#include <cryptopp/files.h>  
#include <cryptopp/pssr.h>  
#include <cryptopp/sha.h>  
#include <cryptopp/oaep.h>  
#include <cryptopp/pubkey.h>  
#include <cryptopp/queue.h>  
#include "Common/FileNameConventions.h"
```

9.322 /home/ryoga/git/pidgirl/pidgirl-engine/RSA/RSAMethod.h File Reference

```
#include <cryptopp/rsa.h>  
#include <cryptopp/osrng.h>  
#include <string>  
#include <iostream>
```

Classes

- class **RSAMethod**
The RSA Method implemented wrapped around the Crypto++ library. This is written to be completely independent of QT.

Index

- ~AStarMap
 - AStarMap, 57
- ~ClientModel
 - ClientModel, 89
- ~ClientSocket
 - ClientSocket, 92
- ~CreateCharacter
 - CreateCharacter, 108
- ~MouseInput
 - MouseInput, 195
- ~MouseState
 - MouseState, 197
- ~NetworkModel
 - NetworkModel, 204
- ~ServerModel
 - ServerModel, 321
- ~UndoRedo
 - UndoRedo, 357
- /home/ryoga/git/pidgirl/pidgirl-engine/AStar/AStarMap.cpp, 375
- /home/ryoga/git/pidgirl/pidgirl-engine/AStar/AStarMap.h, 375
- /home/ryoga/git/pidgirl/pidgirl-engine/AStar/AStarMapEntry.cpp, 375
- /home/ryoga/git/pidgirl/pidgirl-engine/AStar/AStarMapEntry.h, 375
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/BaseCommand.cpp, 376
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/BaseCommand.h, 376
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Command.cpp, 376
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Command.h, 376
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/CommonCommand.cpp, 377
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/CommonCommand.h, 377
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/CopyFolder.cpp, 377
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/CopyFolder.h, 377
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/Quit.cpp, 378
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/Quit.h, 378
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/RemoveFolder.cpp, 378
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Common/RemoveFolder.h, 378
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/AskForSave.cpp, 379
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/AskForSave.h, 379
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/ApplyMapGeometryChanges.cpp, 379
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/ApplyMapGeometryChanges.h, 379
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/AskForSaveMap.cpp, 379
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/AskForSaveMap.h, 380
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/ChangeZoomFactor.cpp, 380
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/ChangeZoomFactor.h, 380
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/CopyWall.cpp, 380
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/CopyWall.h, 380
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/CutWall.cpp, 381
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/CutWall.h, 381
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/LoadMap.cpp, 381
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/LoadMap.h, 381
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/LoadMapByIndex.cpp, 381
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/LoadMapByIndex.h, 382
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/PasteObject.cpp, 382
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/PasteObject.h, 382
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/PlaceEditorObject.cpp, 382
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Editor/Map/PlaceEditorObject.h, 382

- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/Map/PlaceSpecial.cpp, 383
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/Map/PlaceSpecial.h, 383
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/Map/PlaceWall.cpp, 383
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/Map/PlaceWall.h, 383
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/Map/RemoveAllObjects.cpp, 384
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/Map/RemoveAllObjects.h, 384
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/Map/RemoveWall.cpp, 384
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/Map/RemoveWall.h, 384
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/Tools/ChooseMainTool.cpp, 384
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/Tools/ChooseMainTool.h, 385
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/World/AddToChosenList.cpp, 385
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/World/AddToChosenList.h, 385
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/World/ApplyWorldPropertiesChanges.cpp, 385
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/World/ApplyWorldPropertiesChanges.h, 386
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/World/AskForSaveWorld.cpp, 386
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/World/AskForSaveWorld.h, 386
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/World/DuplicateWorldEntry.cpp, 386
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/World/DuplicateWorldEntry.h, 386
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/World/LoadWorld.cpp, 387
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/World/LoadWorld.h, 387
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/World/MoveFile.cpp, 387
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/World/MoveFile.h, 387
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/World/NewFolder.cpp, 388
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/World/NewFolder.h, 388
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/World/NewMapFile.cpp, 388
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/World/NewMapFile.h, 388
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/World/NewWorld.cpp, 389
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/World/NewWorld.h, 389
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/World/RemoveFromChosenList.cpp, 389
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/World/RemoveFromChosenList.h, 389
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/World/RemoveWorldEntry.cpp, 390
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/World/RemoveWorldEntry.h, 390
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/World/RenameWorldEntry.cpp, 390
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/World/RenameWorldEntry.h, 390
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/World/SaveMap.cpp, 391
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/World/SaveMap.h, 391
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/World/SaveMapAs.cpp, 391
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/World/SaveMapAs.h, 391
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/World/SaveWorld.cpp, 392
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/World/SaveWorld.h, 392
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/World/SaveWorldAs.cpp, 392
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Editor/World/SaveWorldAs.h, 392
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/GameCommand.cpp, 393
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/GameCommand.h, 393
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Ingame/CreateCharacter.cpp, 393
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Ingame/CreateCharacter.h, 393
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Ingame/EndTurn.cpp, 394
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Ingame/EndTurn.h, 394

- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Ingame/FirstTurn.cpp, 394
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Ingame/FirstTurn.h, 394
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Ingame/LoadStartMapFromIndex.cpp, 395
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Ingame/LoadStartMapFromIndex.h, 395
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Ingame/NextTurn.cpp, 395
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Ingame/NextTurn.h, 395
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Ingame/PreloadAllMaps.cpp, 396
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Ingame/PreloadAllMaps.h, 396
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Ingame/SaveEMailGame.cpp, 396
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Ingame/SaveEMailGame.h, 396
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Ingame/SaveGame.cpp, 397
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Ingame/SaveGame.h, 397
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Ingame/ShowGameOptions.cpp, 397
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Ingame/ShowGameOptions.h, 398
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Ingame/SpawnGameObject.cpp, 398
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Ingame/SpawnGameObject.h, 398
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Ingame/SpawnPlayerOnCurrentMap.cpp, 398
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Ingame/SpawnPlayerOnCurrentMap.h, 399
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Ingame/YourTurnInformation.cpp, 399
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Ingame/YourTurnInformation.h, 399
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Map/ClearPath.cpp, 399
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Map/ClearPath.h, 399
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Map/FindPath.cpp, 400
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Map/FindPath.h, 400
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Map/MoveCharacter.cpp, 400
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Map/MoveCharacter.h, 400
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Startmenu/ConnectTo.cpp, 401
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Startmenu/ConnectTo.h, 401
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Startmenu/LoadGame.cpp, 401
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Startmenu/LoadGame.h, 402
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Startmenu/PreloadWorld.cpp, 402
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Startmenu/PreloadWorld.h, 402
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Startmenu/ResumeEMailGame.cpp, 403
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Startmenu/ResumeEMailGame.h, 403
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Startmenu/SetupNewGame.cpp, 403
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Game/Startmenu/SetupNewGame.h, 404
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Client/ChangePlayerIDOnServer.cpp, 404
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Client/ChangePlayerIDOnServer.h, 404
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Client/ClientCommand.cpp, 404
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Client/ClientCommand.h, 405
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Client/ReceiveInitialPlayerInformation.cpp, 405
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Client/ReceiveInitialPlayerInformation.h, 405
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Client/ReceiveMap.cpp, 405

- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Client/ReceiveMap.h, 405
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Client/ReceivePlayer.cpp, 406
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Client/ReceivePlayer.h, 406
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Client/ReceivePlayer-Models.cpp, 406
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Client/ReceivePlayer-Models.h, 406
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Client/RequestMapFrom-Server.cpp, 406
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Client/RequestMapFrom-Server.h, 407
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Client/RequestPlayerFrom-Server.cpp, 407
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Client/RequestPlayerFrom-Server.h, 407
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Client/RequestPlayer-ModelsFromServer.cpp, 407
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Client/RequestPlayer-ModelsFromServer.h, 407
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Client/SendSpawnCharacter-ToServer.cpp, 408
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Client/SendSpawnCharacter-ToServer.h, 408
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Client/SendToServerEnd-Turn.cpp, 408
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Client/SendToServerEnd-Turn.h, 408
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Client/SendUpdateCharacter-ToServer.cpp, 408
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Client/SendUpdateCharacter-ToServer.h, 408
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/NetworkCommand.cpp, 409
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/NetworkCommand.h, 409
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/ReceiveSpawnCharacter-From.cpp, 409
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/ReceiveSpawnCharacter-From.h, 409
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/ReceiveUpdateCharacter-From.cpp, 409
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/ReceiveUpdateCharacter-From.h, 410
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Server/Broadcast/Broadcast-Command.cpp, 410
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Server/Broadcast/Broadcast-Command.h, 410
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Server/Broadcast/Broadcast-SpawnCharacter.cpp, 410
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Server/Broadcast/Broadcast-SpawnCharacter.h, 410
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Server/Broadcast/Broadcast-UpdateCharacter.cpp, 411
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Server/Broadcast/Broadcast-UpdateCharacter.h, 411
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Server/ClientDisconnected.-cpp, 411
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Server/ClientDisconnected.-h, 411
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Server/ReceiveChange-PlayerID.cpp, 411
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Server/ReceiveChange-PlayerID.h, 411
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Server/ReplyToRequest-ForMap.cpp, 412
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Server/ReplyToRequest-ForMap.h, 412
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Server/ReplyToRequest-ForPlayer.cpp, 412
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Server/ReplyToRequest-ForPlayer.h, 412
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Server/ReplyToRequest-ForPlayerModels.cpp, 412
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/-Backend/Network/Server/ReplyToRequest-ForPlayerModels.h, 413

- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/SendInitialPlayerInformation.cpp, 413
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/SendInitialPlayerInformation.h, 413
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/SendToClientNextTurn.cpp, 413
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/SendToClientNextTurn.h, 413
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/ServerCommand.cpp, 414
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Network/Server/ServerCommand.h, 414
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/NotUndoable.cpp, 414
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/NotUndoable.h, 414
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Object/ObjectCommand.cpp, 414
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Object/ObjectCommand.h, 414
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Object/ShowDescription.cpp, 415
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Object/ShowDescription.h, 415
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/ShowAbout.cpp, 415
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/ShowAbout.h, 415
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Undoable.cpp, 416
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/Undoable.h, 416
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/UseRedo.cpp, 416
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/UseRedo.h, 416
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/UseUndo.cpp, 416
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Backend/UseUndo.h, 417
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Processor.cpp, 417
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/Processor.h, 417
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/UndoRedo.cpp, 417
- /home/ryoga/git/pidgirl/pidgirl-engine/Command/UndoRedo.h, 417
- /home/ryoga/git/pidgirl/pidgirl-engine/Common/Common.cpp, 418
- /home/ryoga/git/pidgirl/pidgirl-engine/Common/Common.h, 418
- /home/ryoga/git/pidgirl/pidgirl-engine/Common/FileManager.cpp, 420
- /home/ryoga/git/pidgirl/pidgirl-engine/Common/FileManager.h, 420
- /home/ryoga/git/pidgirl/pidgirl-engine/Common/FileNameConventions.cpp, 421
- /home/ryoga/git/pidgirl/pidgirl-engine/Common/FileNameConventions.h, 421
- /home/ryoga/git/pidgirl/pidgirl-engine/Common/OptionsLayout.cpp, 422
- /home/ryoga/git/pidgirl/pidgirl-engine/Common/OptionsLayout.h, 422
- /home/ryoga/git/pidgirl/pidgirl-engine/Common/Serialize.cpp, 422
- /home/ryoga/git/pidgirl/pidgirl-engine/Common/Serialize.h, 422
- /home/ryoga/git/pidgirl/pidgirl-engine/Common/YesNoCancelDialog.cpp, 422
- /home/ryoga/git/pidgirl/pidgirl-engine/Common/YesNoCancelDialog.h, 423
- /home/ryoga/git/pidgirl/pidgirl-engine/Editor/EditorCommon.h, 423
- /home/ryoga/git/pidgirl/pidgirl-engine/Editor/EditorMenu.cpp, 423
- /home/ryoga/git/pidgirl/pidgirl-engine/Editor/EditorMenu.h, 424
- /home/ryoga/git/pidgirl/pidgirl-engine/Editor/EditorTools.cpp, 424
- /home/ryoga/git/pidgirl/pidgirl-engine/Editor/EditorTools.h, 424
- /home/ryoga/git/pidgirl/pidgirl-engine/Editor/MapPropertiesDialog.cpp, 424
- /home/ryoga/git/pidgirl/pidgirl-engine/Editor/MapPropertiesDialog.h, 425
- /home/ryoga/git/pidgirl/pidgirl-engine/Editor/ModelEditorModel.cpp, 425
- /home/ryoga/git/pidgirl/pidgirl-engine/Editor/ModelEditorModel.h, 425
- /home/ryoga/git/pidgirl/pidgirl-engine/Editor/ModelLocalFilePaths.cpp, 426
- /home/ryoga/git/pidgirl/pidgirl-engine/Editor/ModelLocalFilePaths.h, 426
- /home/ryoga/git/pidgirl/pidgirl-engine/Editor/ModelWorldModel.cpp, 426
- /home/ryoga/git/pidgirl/pidgirl-engine/Editor/ModelWorldModel.h, 426
- /home/ryoga/git/pidgirl/pidgirl-engine/Editor/WorldFileTree.cpp, 427
- /home/ryoga/git/pidgirl/pidgirl-engine/Editor/WorldFileTree.h, 427
- /home/ryoga/git/pidgirl/pidgirl-engine/Editor/WorldPropertiesDialog.cpp, 427
- /home/ryoga/git/pidgirl/pidgirl-engine/Editor/WorldPropertiesDialog.h, 428
- /home/ryoga/git/pidgirl/pidgirl-engine/Editor/WorldView.cpp, 428
- /home/ryoga/git/pidgirl/pidgirl-engine/Editor/WorldView.h, 428

- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Email-SetupWidget.cpp, 428
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Email-SetupWidget.h, 429
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/General-SetupWidget.cpp, 429
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/General-SetupWidget.h, 429
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/ClientGameModel.cpp, 429
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/ClientGameModel.h, 430
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/EmailGameModel.cpp, 430
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/EmailGameModel.h, 430
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/GameModel.cpp, 430
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/GameModel.h, 431
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/HotseatGameModel.cpp, 431
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/HotseatGameModel.h, 431
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/LocalGameModel.cpp, 432
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/LocalGameModel.h, 432
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/MailGameInfo.cpp, 432
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/MailGameInfo.h, 432
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/PlayerModel.cpp, 433
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/PlayerModel.h, 433
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/SaveGameFile.cpp, 433
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/SaveGameFile.h, 433
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/ServerGameModel.cpp, 434
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Model/ServerGameModel.h, 434
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Network-SetupWidget.cpp, 434
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Network-SetupWidget.h, 434
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Solo-HotseatSetupWidget.cpp, 435
- /home/ryoga/git/pidgirl/pidgirl-engine/Game/Solo-HotseatSetupWidget.h, 435
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Command-Action.cpp, 435
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Command-Action.h, 436
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Main-ViewMap.cpp, 436
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Main-ViewMap.h, 436
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Main-ViewStack.cpp, 436
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Main-ViewStack.h, 437
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MapModel.cpp, 437
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MapModel.h, 437
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ContextMenuMenuState.cpp, 437
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ContextMenuMenuState.h, 438
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ContextGameMenuState.cpp, 438
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/ContextGameMenuState.h, 438
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/IdleEditorMouseState.cpp, 438
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/IdleEditorMouseState.h, 438
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/IdleGameMouseState.cpp, 439
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/IdleGameMouseState.h, 439
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/MouseEditorInput.cpp, 439
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/MouseEditorInput.h, 439
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/MouseGameInput.cpp, 440
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/MouseGameInput.h, 440
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/MouseInput.cpp, 440
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/MouseInput.h, 440
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/MouseState.cpp, 441
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/MouseState.h, 441
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/PlaceObjectsState.cpp, 441
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/PlaceObjectsState.h, 441
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/RemoveObjectsState.cpp, 442
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/RemoveObjectsState.h, 442
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/TargetChosenState.cpp, 442
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/TargetChosenState.h, 442
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/MouseInput/TargetConfirmedState.cpp, 443

- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/-
 MouseButton/TargetConfirmedState.h, 443
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/-
 MouseButton/TargetMoveState.cpp, 443
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/-
 MouseButton/TargetMoveState.h, 443
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/-
 StackModel.cpp, 444
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Model/-
 StackModel.h, 444
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Object-
 Menu.cpp, 444
- /home/ryoga/git/pidgirl/pidgirl-engine/MainView/Object-
 Menu.h, 444
- /home/ryoga/git/pidgirl/pidgirl-engine/MainWindow.cpp,
 445
- /home/ryoga/git/pidgirl/pidgirl-engine/MainWindow.h,
 445
- /home/ryoga/git/pidgirl/pidgirl-engine/Network/Client-
 Model.cpp, 446
- /home/ryoga/git/pidgirl/pidgirl-engine/Network/Client-
 Model.h, 446
- /home/ryoga/git/pidgirl/pidgirl-engine/Network/Client-
 Socket.cpp, 447
- /home/ryoga/git/pidgirl/pidgirl-engine/Network/Client-
 Socket.h, 447
- /home/ryoga/git/pidgirl/pidgirl-engine/Network/Network-
 Model.cpp, 447
- /home/ryoga/git/pidgirl/pidgirl-engine/Network/Network-
 Model.h, 447
- /home/ryoga/git/pidgirl/pidgirl-engine/Network/Server-
 Model.cpp, 448
- /home/ryoga/git/pidgirl/pidgirl-engine/Network/Server-
 Model.h, 449
- /home/ryoga/git/pidgirl/pidgirl-engine/Network/Waiting-
 ForDataDialog.cpp, 449
- /home/ryoga/git/pidgirl/pidgirl-engine/Network/Waiting-
 ForDataDialog.h, 449
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/Character/-
 Player.cpp, 449
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/Character/-
 Player.h, 449
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/Object-
 Base.cpp, 450
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/Object-
 Base.h, 450
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/Object-
 Character.cpp, 450
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/Object-
 Character.h, 450
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/Object-
 FromID.cpp, 451
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/Object-
 FromID.h, 451
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectID.-
 cpp, 451
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectID.h,
 451
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/Object-
 Image.cpp, 452
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/Object-
 Image.h, 452
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/Object-
 Item.cpp, 452
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/Object-
 Item.h, 452
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/Object-
 Special.cpp, 452
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/Object-
 Special.h, 452
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectTile.-
 cpp, 453
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/ObjectTile.-
 h, 453
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/Special/-
 StartPosition.cpp, 453
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/Special/-
 StartPosition.h, 453
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/Tile/Brick-
 Wall.cpp, 453
- /home/ryoga/git/pidgirl/pidgirl-engine/Object/Tile/Brick-
 Wall.h, 454
- /home/ryoga/git/pidgirl/pidgirl-engine/RSA/RSAMethod.-
 cpp, 454
- /home/ryoga/git/pidgirl/pidgirl-engine/RSA/RSAMethod.-
 h, 454
- /home/ryoga/git/pidgirl/pidgirl-engine/main.cpp, 435
- AStar, 23
- AStarMap, 56
 - ~AStarMap, 57
 - AStarMap, 57
 - AStarMap, 57
 - calculatePath, 57
 - fromMapModel, 57
 - getMapHeight, 58
 - getMapWidth, 58
 - getPath, 58
 - getPossibleNeighborsOf, 58
 - map, 58
 - path, 58
 - refEntry, 58
 - retracePath, 58
- aStarMap
 - FindPath, 129
- AStarMapEntry, 58
 - AStarMapEntry, 60
 - AStarMapEntry, 60
 - calculateF, 60
 - cameFrom, 62
 - copy, 60
 - f, 62
 - fromStackModel, 60
 - g, 62
 - getF, 60
 - getG, 60
 - getPosition, 60

- getWeight, 60
- hasInfinityWeight, 60
- heuristicDistanceTo, 60
- init, 60
- operator=, 62
- operator==, 62
- position, 62
- refCameFrom, 62
- setCameFrom, 62
- setF, 62
- setG, 62
- setWeight, 62
- setWeightToInfinity, 62
- weight, 62
- about
 - EditorMenu, 114
- aboutAction
 - EditorMenu, 116
- absoluteMapPathToLocalMapPath
 - WorldModel, 367
- activePlayer
 - PlayerModel, 254
- add
 - UndoRedo, 357
- addButton
 - EditorTools, 119
 - WorldPropertiesDialog, 369
- addHotseatPlayer
 - ShowGameOptions, 330
- addNewPlayer
 - ClientGameModel, 83
 - GameModel, 136
 - LocalGameModel, 160
- addNewPlayerButton
 - ShowGameOptions, 331
- addPathToChosen
 - WorldPropertiesDialog, 369
- addPlayerStack
 - MapModel, 177
- addPropertyWidget
 - OptionsLayout, 241
- addStartPositionStack
 - MapModel, 177
- AddToChosenList, 47
 - AddToChosenList, 48
 - AddToChosenList, 48
 - availableFileSystemModel, 48
 - chosenListModel, 48
 - currentIndex, 48
 - execute, 48
- addtoPreloadedMaps
 - ClientGameModel, 84
 - GameModel, 137
 - LocalGameModel, 160
- allDataReceivedFromClient
 - NetworkModel, 204
- allDataReceivedFromServer
 - NetworkModel, 204
- alreadyPlaced
 - PlaceObjectsState, 246
 - RemoveObjectsState, 278
- ApplyWorldPropertiesChanges, 49
- appendHeightLeft
 - MapModel, 177
- appendHeightRight
 - MapModel, 177
- appendHorizontalLeft
 - MapModel, 177
- appendHorizontalRight
 - MapModel, 177
- appendImage
 - ObjectBase, 219
- appendImages
 - ObjectBase, 219
- appendVerticalLeft
 - MapModel, 177
- appendVerticalRight
 - MapModel, 178
- applyButton
 - OptionsLayout, 241
- applyChanges
 - MapPropertiesDialog, 186
 - WorldPropertiesDialog, 369
- applyChosenStarterMaps
 - ApplyWorldPropertiesChanges, 52
- applyGeneralProperties
 - SetupNewGame, 325
- applyGeometryChanges
 - ApplyMapGeometryChanges, 50
- ApplyMapGeometryChanges, 49
 - applyGeometryChanges, 50
 - ApplyMapGeometryChanges, 50
 - ApplyMapGeometryChanges, 50
 - execute, 50
 - horizontalDirection, 51
 - horizontalNewValue, 51
 - maxHeightDirection, 51
 - maxHeightNewValue, 51
 - verticalDirection, 51
 - verticalNewValue, 51
- ApplyWorldPropertiesChanges, 51
 - applyChosenStarterMaps, 52
 - ApplyWorldPropertiesChanges, 52
 - ApplyWorldPropertiesChanges, 52
 - chosenListModel, 52
 - execute, 52
- ask
 - LoadMap, 152
 - LoadMapByIndex, 154
- AskForSave, 52
 - AskForSave, 53
 - AskForSave, 53
 - execute, 53
- AskForSaveMap, 54
 - AskForSaveMap, 54
 - AskForSaveMap, 54

- execute, 54
- yncDialog, 55
- AskForSaveWorld, 55
 - AskForSaveWorld, 56
 - AskForSaveWorld, 56
 - execute, 56
 - yncDialog, 56
- askToSaveGame
 - Quit, 262
- authenticatePlayer
 - MailGameInfo, 163
- availableFileSystemModel
 - AddToChosenList, 48
 - WorldPropertiesDialog, 369
- availableTreeUpdateColumnWidth
 - WorldPropertiesDialog, 369
- availableTreeView
 - WorldPropertiesDialog, 369
- BRICK_WALL
 - ObjectID, 229
- BRICKWALL
 - PIDGIRL, 40
- BROADCAST_SPAWN_CHARACTER
 - NetworkModel.h, 448
- BROADCAST_UPDATE_CHARACTER
 - NetworkModel.h, 448
- BaseCommand, 63
 - BaseCommand, 64
 - BaseCommand, 64
 - errorEnabled, 66
 - execute, 64
 - getCommandIcon, 65
 - isClearingUndoRedo, 65
 - isErrorEnabled, 65
 - isExecutable, 65
 - isUndoable, 65
 - redo, 65
 - setErrorEnabled, 65
 - setModels, 65
 - undo, 65
- beginTurn
 - ClientGameModel, 84
 - EmailGameModel, 121
 - GameModel, 137
 - HotseatGameModel, 144
 - ServerGameModel, 318
- BrickWall, 66
 - BrickWall, 67
 - BrickWall, 67
 - copy, 67
 - createInformationWidget, 67
 - deserialize, 67
 - getCommandList, 67
 - getObjectID, 67
 - serialize, 68
 - update, 68
- BroadcastCommand, 68
 - BroadcastCommand, 69
 - broadcastPlayerIDs, 69
 - BroadcastCommand, 69
 - execute, 69
- broadcastPlayerIDs
 - BroadcastCommand, 69
- BroadcastSpawnCharacter, 70
 - BroadcastSpawnCharacter, 71
 - BroadcastSpawnCharacter, 71
 - execute, 71
 - newCharacter, 71
- BroadcastUpdateCharacter, 71
 - BroadcastUpdateCharacter, 73
 - BroadcastUpdateCharacter, 73
 - execute, 73
 - newCharacter, 73
 - oldCharacter, 73
- button
 - ChooseMainTool, 77
- ButtonClicked
 - YesNoCancelDialog, 372
- buttonClicked
 - YesNoCancelDialog, 372
- buttonVector
 - EditorTools, 119
- CANCEL_BUTTON
 - YesNoCancelDialog, 372
- CHANGE_PLAYER_ID
 - NetworkModel.h, 448
- CHARACTER
 - ObjectBase, 219
- calculateF
 - AStarMapEntry, 60
- calculatePath
 - AStarMap, 57
- cameFrom
 - AStarMapEntry, 62
- changeClientID
 - ClientModel, 89
 - NetworkModel, 204
 - ServerModel, 321
- changePlayerID
 - ClientGameModel, 84
 - GameModel, 137
 - HotseatGameModel, 144
- ChangePlayerIDOnServer, 73
 - ChangePlayerIDOnServer, 74
 - ChangePlayerIDOnServer, 74
 - execute, 74
 - newID, 75
- ChangeZoomFactor, 75
 - ChangeZoomFactor, 76
 - ChangeZoomFactor, 76
 - execute, 76
 - newFactor, 76
- character
 - StackModel, 347
- characterChanged
 - StackModel, 339

- characterCreationDialog
 - CreateCharacter, 109
- characterCreationLayout
 - CreateCharacter, 109
- characterDone
 - CreateCharacter, 108
- characterExists
 - StackModel, 339
- CharacterID
 - ObjectID, 228
- characterMenu
 - MouseEditorInput, 192
- characterSpawnIDMap
 - ObjectID, 229
- characterSpawned
 - PlayerModel, 254
- checkCharacterActions
 - MouseEditorInput, 189
- checkFloorActions
 - MouseEditorInput, 189
- checkItemsActions
 - MouseEditorInput, 189
- checkLocale
 - main.cpp, 435
- checkPasteAction
 - MouseEditorInput, 189
- checkRemoveAllAction
 - MouseEditorInput, 189
- checkSpecialsActions
 - MouseEditorInput, 189
- checkWallActions
 - MouseEditorInput, 189
- choosableCharacterWidgets
 - FirstTurn, 131
- chooseCharacterDialog
 - FirstTurn, 131
- chooseCharacterLayout
 - FirstTurn, 131
- chooseListItems
 - CreateCharacter, 109
- chooseListModel
 - CreateCharacter, 109
- chooseListView
 - CreateCharacter, 109
- ChooseMainTool, 76
 - button, 77
 - ChooseMainTool, 77
 - ChooseMainTool, 77
 - execute, 77
- chooseStartMap
 - CreateCharacter, 109
- chooseStartMapLayout
 - CreateCharacter, 109
- chosedSaveDir
 - ShowGameOptions, 330
- chosenListModel
 - AddToChosenList, 48
 - ApplyWorldPropertiesChanges, 52
 - RemoveFromChosenList, 277
 - WorldPropertiesDialog, 369
- chosenListView
 - WorldPropertiesDialog, 369
- cipherText
 - MailGameInfo, 164
- clear
 - UndoRedo, 357
- clearImageList
 - ObjectBase, 220
- ClearPath, 78
 - ClearPath, 78
 - ClearPath, 78
 - execute, 78
- clearPath
 - MapModel, 178
- clearRedoList
 - UndoRedo, 357
- clearStackModelGrid
 - MapModel, 178
- clearUndoList
 - UndoRedo, 357
- clearsUndoRedo
 - NotUndoable, 216
- ClientCommand, 79
 - ClientCommand, 80
 - ClientCommand, 80
 - execute, 80
 - isExecutable, 80
- clientDisconnect
 - NetworkModel, 205
- ClientDisconnected, 80
 - ClientDisconnected, 81
 - ClientDisconnected, 81
 - execute, 81
- ClientGameModel, 82
 - addNewPlayer, 83
 - addtoPreloadedMaps, 84
 - beginTurn, 84
 - changePlayerID, 84
 - ClientGameModel, 83
 - ClientGameModel, 83
 - convertFromDifferentGameModel, 84
 - currentMap, 87
 - deactivatePlayer, 84
 - deserialize, 84
 - endTurn, 85
 - firstTurn, 85
 - getGameType, 85
 - getPlayerModels, 85
 - getPreloadedMaps, 85
 - isYourTurn, 85
 - loadMap, 85
 - nextTurn, 86
 - processor, 87
 - refCurrentMap, 86
 - refPlayer, 86
 - refPlayerModel, 86

- refPreloadedMap, 86
- serialize, 86
- setCurrentMap, 86
- setYourTurn, 86
- spawnCharacter, 86
- updateCharacter, 87
- yourTurn, 87
- ClientModel, 87
 - ~ClientModel, 89
 - changeClientID, 89
 - ClientModel, 89
 - ClientModel, 89
 - connectTo, 89
 - connection, 91
 - disconnectedFromServer, 89
 - getConnectedPlayerIDs, 89
 - getConnectionList, 89
 - ipAddress, 91
 - isClient, 90
 - isServer, 90
 - processor, 91
 - receiveData, 90
 - receivePendingData, 90
 - sendData, 90, 91
 - socketError, 91
- clientModel
 - ConnectTo, 101
- ClientSocket, 91
 - ~ClientSocket, 92
 - ClientSocket, 92
 - ClientSocket, 92
 - disconnected, 92
 - getPlayerID, 92
 - gotDisconnected, 92
 - gotReadyRead, 92
 - playerID, 92
 - readyRead, 92
 - refSocket, 92
 - setPlayerID, 92
 - socket, 92
- clients
 - ServerModel, 323
- clipboardObject
 - MapModel, 184
- clipboardObjectSet
 - MapModel, 178
- closeEvent
 - MainWindow, 171
- Command, 93
 - Command, 94
 - editorModel, 96
 - execute, 94
 - gameModel, 96
 - isClearingUndoRedo, 95
 - isExecutable, 95
 - isUndoable, 95
 - mapModel, 96
 - networkModel, 96
 - processor, 96
 - redo, 95
 - setModels, 95
 - undo, 96
 - undoRedo, 96
 - worldModel, 96
- CommandAction, 96
 - CommandAction, 97
 - commandActionTriggered, 97
 - CommandAction, 97
 - triggerWithName, 97
- commandActionTriggered
 - CommandAction, 97
- commandProcessed
 - Processor, 258
- CommandProcessor, 28
- Commands, 24
- Common, 29
- CommonCommand, 97
 - CommonCommand, 98
 - CommonCommand, 98
 - execute, 98
 - isExecutable, 98
- ConnectTo, 99
 - clientModel, 101
 - ConnectTo, 100
 - connectToDialog, 101
 - connectToLayout, 101
 - connectToServer, 100
 - ConnectTo, 100
 - execute, 100
 - IPAddressEdit, 101
 - IPAddressLabel, 101
 - IPAddressValidator, 101
 - initToConnectDialog, 100
 - portEdit, 101
 - portLabel, 101
 - portValidator, 101
 - startNetworkButton, 101
- connectTo
 - ClientModel, 89
- connectToDialog
 - ConnectTo, 101
- connectToLayout
 - ConnectTo, 101
- connectToServer
 - ConnectTo, 100
- connection
 - ClientModel, 91
- connectionEstablished
 - NetworkModel, 205
- connectionLabel
 - CreateCharacter, 109
- connectionListItems
 - CreateCharacter, 109
- connectionListModel
 - CreateCharacter, 109
- connectionListView

- CreateCharacter, 109
- connectionsChanged
 - CreateCharacter, 108
- ContextEditorMenuState, 101
 - ContextEditorMenuState, 102
 - ContextEditorMenuState, 102
 - nextState, 102
 - showContextMenu, 102
- ContextGameMenuState, 102
 - ContextGameMenuState, 103
 - ContextGameMenuState, 103
 - nextState, 103
 - showContextMenu, 103
- contextMenu
 - MouseInput, 195
 - WorldFileTree, 365
- contextMenuStack
 - MouseEditorInput, 192
- convertFromDifferentGameModel
 - ClientGameModel, 84
 - EMailGameModel, 121
 - GameModel, 137
 - HotseatGameModel, 144
 - LocalGameModel, 160
 - ServerGameModel, 318
- convertToFileName
 - LocalFilePaths, 158
- convertToFileNameVector
 - LocalFilePaths, 158
- copy
 - AStarMapEntry, 60
 - BrickWall, 67
 - CopyFolder, 105
 - ObjectCharacter, 225
 - ObjectItem, 232
 - ObjectSpecial, 235
 - ObjectTile, 238
 - Player, 251
 - StartPosition, 349
- copyBase
 - ObjectBase, 220
 - ObjectCharacter, 225
 - ObjectItem, 232
 - ObjectSpecial, 235
 - ObjectTile, 238
- copyCharacter
 - MouseEditorInput, 190
- copyCharacterAction
 - MouseEditorInput, 192
- copyFloor
 - MouseEditorInput, 190
- copyFloorAction
 - MouseEditorInput, 192
- CopyFolder, 104
 - copy, 105
 - CopyFolder, 105
 - CopyFolder, 105
 - execute, 105
 - from, 105
 - to, 105
- copyItems
 - MouseEditorInput, 190
- copyItemsAction
 - MouseEditorInput, 192
- copySpecials
 - MouseEditorInput, 190
- copySpecialsAction
 - MouseEditorInput, 192
- CopyWall, 105
 - CopyWall, 106
 - CopyWall, 106
 - execute, 106
 - wall, 107
- copyWall
 - MouseEditorInput, 190
- copyWallAction
 - MouseEditorInput, 192
- correctPositionEntries
 - MapModel, 178
- CreateCharacter, 107
 - ~CreateCharacter, 108
 - characterCreationDialog, 109
 - characterCreationLayout, 109
 - characterDone, 108
 - chooseListItems, 109
 - chooseListModel, 109
 - chooseListView, 109
 - chooseStartMap, 109
 - chooseStartMapLayout, 109
 - connectionLabel, 109
 - connectionListItems, 109
 - connectionListModel, 109
 - connectionListView, 109
 - connectionsChanged, 108
 - CreateCharacter, 108
 - CreateCharacter, 108
 - deleteConnectionListItems, 109
 - execute, 109
 - informationWidget, 109
 - informationWidgetLayout, 109
 - init, 109
 - playerNameEdit, 109
 - playerNameValidator, 109
 - serverOverview, 110
 - serverOverviewLayout, 110
- createInformationWidget
 - BrickWall, 67
 - ObjectBase, 220
 - Player, 251
 - StartPosition, 349
- createMainViewMap
 - MainWindow, 171
- createNewWorld
 - EditorMenu, 114
- createStackGrid
 - MainViewMap, 165

- createStackModelAt
 - MapModel, 178
- currentHeight
 - MapModel, 184
- currentIndex
 - AddToChosenList, 48
 - RemoveFromChosenList, 277
- currentIndexChangedTo
 - OptionsLayout, 241
- currentMap
 - ClientGameModel, 87
- currentMouseState
 - MouseInput, 195
- currentPath
 - MapModel, 184
- currentPlayerID
 - ServerGameModel, 320
- cutCharacter
 - MouseEditorInput, 190
- cutCharacterAction
 - MouseEditorInput, 192
- cutFloor
 - MouseEditorInput, 190
- cutFloorAction
 - MouseEditorInput, 192
- cutItems
 - MouseEditorInput, 190
- cutItemsAction
 - MouseEditorInput, 192
- cutSpecials
 - MouseEditorInput, 190
- cutSpecialsAction
 - MouseEditorInput, 192
- CutWall, 110
 - CutWall, 111
 - CutWall, 111
 - execute, 111
 - stackModel, 111
- cutWall
 - MouseEditorInput, 190
- cutWallAction
 - MouseEditorInput, 192
- DESTINATION
 - StackModel, 339
- dataReceived
 - WaitingForDataDialog, 362
- deactivatePlayer
 - ClientGameModel, 84
 - GameModel, 137
 - LocalGameModel, 160
 - ServerGameModel, 318
- decryptString
 - RSAMethod, 295
- deleteAll
 - StackModel, 339
- deleteAllItems
 - StackModel, 339
- deleteAllSpecials
 - StackModel, 339
- deleteCharacter
 - StackModel, 339
- deleteConnectionListItems
 - CreateCharacter, 109
- deleteContextMenu
 - MouseInput, 195
- deleteDisconnectedClient
 - ServerModel, 322
- deleteFloor
 - StackModel, 339
- deleteItem
 - StackModel, 339
- deletePlayerModels
 - GameModel, 138
- deleteSpecial
 - StackModel, 341
- deleteStackGrid
 - MainViewMap, 166
- deleteStackModelAt
 - MapModel, 178
- deleteWall
 - StackModel, 341
- description
 - ObjectBase, 223
- deserialize
 - BrickWall, 67
 - ClientGameModel, 84
 - EMailGameModel, 121
 - FileManager, 126
 - GameModel, 138
 - HotseatGameModel, 145
 - LocalGameModel, 160
 - MailGameInfo, 163
 - MapModel, 178
 - ObjectBase, 220
 - ObjectCharacter, 225
 - ObjectID, 229
 - ObjectItem, 232
 - ObjectSpecial, 235, 236
 - ObjectTile, 239
 - Player, 251
 - PlayerModel, 253
 - SaveGameFile, 300
 - Serialize, 314
 - ServerGameModel, 318
 - StackModel, 341
 - StartPosition, 349
 - WorldModel, 367
- dir
 - NewMapFile, 210
- disconnected
 - ClientSocket, 92
- disconnectedFromServer
 - ClientModel, 89
- done
 - FirstTurn, 130
 - MainWindow, 172

- PIDGIRL, 41
- dragMoveEvent
 - WorldFileTree, 364
- draggedIndex
 - WorldFileTree, 365
- drawDirectionIndicator
 - MainViewStack, 168
- drawObject
 - MainViewStack, 168
- dropEvent
 - WorldFileTree, 364
- duplicate
 - WorldFileTree, 364
- duplicateAction
 - WorldFileTree, 365
- DuplicateWorldEntry, 111
 - DuplicateWorldEntry, 112
 - DuplicateWorldEntry, 112
- execute, 112
- index, 112
- E
 - StackModel, 339
- EDITOR_VIEW_HEIGHT
 - PIDGIRL, 41
- EDITOR_VIEW_WIDTH
 - PIDGIRL, 41
- EMAIL
 - PIDGIRL, 40
- EMAIL_FOLDER_NAME
 - PIDGIRL, 41
- EMAIL_SUFFIX
 - PIDGIRL, 41
- EMailGameModel, 120
 - beginTurn, 121
 - convertFromDifferentGameModel, 121
 - deserialize, 121
 - EMailGameModel, 121
 - EMailGameModel, 121
 - endTurn, 122
 - firstTurn, 122
 - getGameType, 122
 - getTurnCount, 122
 - mailGameInfos, 122
 - nextPlayerID, 122
 - nextTurn, 122
 - refMailGameInfo, 122
 - serialize, 122
 - setTurnCount, 122
 - turnCount, 122
- EMailSetupWidget, 123
 - EMailSetupWidget, 123
 - emailWidgetLayout, 123
 - EMailSetupWidget, 123
 - gameStartButton, 123
 - playerCountLabel, 123
 - playerCountSpinBox, 123
- END_TURN
 - NetworkModel.h, 448
- editCharacter
 - MouseEditorInput, 190
- editCharacterAction
 - MouseEditorInput, 192
- editFloor
 - MouseEditorInput, 190
- editFloorAction
 - MouseEditorInput, 192
- editItems
 - MouseEditorInput, 191
- editItemsAction
 - MouseEditorInput, 192
- editMenu
 - EditorMenu, 116
- editSpecials
 - MouseEditorInput, 191
- editSpecialsAction
 - MouseEditorInput, 192
- editWall
 - MouseEditorInput, 191
- editWallAction
 - MouseEditorInput, 192
- EditorMenu, 113
 - about, 114
 - aboutAction, 116
 - createNewWorld, 114
 - editMenu, 116
 - EditorMenu, 114
 - editorModel, 116
 - EditorMenu, 114
 - fileMenu, 116
 - helpMenu, 116
 - loadWorld, 114
 - loadWorldAction, 116
 - mapPropertiesAction, 116
 - mapPropertiesDialog, 116
 - newWorldAction, 116
 - processor, 116
 - quit, 115
 - quitAction, 116
 - redoAction, 116
 - saveMap, 115
 - saveMapAction, 116
 - saveMapAs, 115
 - saveMapAsAction, 116
 - saveWorld, 115
 - saveWorldAction, 116
 - saveWorldAs, 115
 - saveWorldAsAction, 116
 - showMapProperties, 115
 - showWorldProperties, 115
 - undoAction, 116
 - updateSize, 115
 - useRedo, 115
 - useUndo, 115
 - viewMenu, 116
 - worldModel, 116
 - worldPropertiesAction, 116

- worldPropertiesDialog, 116
- zoom, 115
- zoomActionGroup, 116
- zoomMenu, 116
- editorMenu
 - MainWindow, 172
- EditorModel, 117
 - EditorModel, 117
 - EditorModel, 117
 - getIsEditor, 117
 - isEditor, 118
 - mainToolObject, 118
 - refMainObject, 117
 - setMainToolObject, 117
- editorModel
 - Command, 96
 - EditorMenu, 116
 - EditorTools, 119
 - MainViewMap, 166
 - MainViewStack, 169
 - MainWindow, 172
 - Processor, 260
- EditorTools, 118
 - addButton, 119
 - buttonVector, 119
 - editorModel, 119
 - EditorTools, 119
 - EditorTools, 119
 - gridLayout, 120
 - initializeToolButtons, 119
 - processor, 120
 - setMainTool, 119
 - toolButtons, 120
- editorTools
 - MainWindow, 172
- editorToolsDock
 - MainWindow, 172
- emailGameDir
 - MailGameInfo, 164
- emailGameDirSet
 - MailGameInfo, 164
- emailSetupWidget
 - SetupNewGame, 326
- emailWidgetLayout
 - EMailSetupWidget, 123
- encryptString
 - RSAMethod, 295
- EndTurn, 124
 - EndTurn, 124
 - EndTurn, 124
 - execute, 125
- endTurn
 - ClientGameModel, 85
 - EMailGameModel, 122
 - GameModel, 138
 - HotseatGameModel, 145
 - MouseGameInput, 194
 - ServerGameModel, 318
- errorEnabled
 - BaseCommand, 66
- exec
 - WorldPropertiesDialog, 369
- execute
 - AddToChosenList, 48
 - ApplyMapGeometryChanges, 50
 - ApplyWorldPropertiesChanges, 52
 - AskForSave, 53
 - AskForSaveMap, 54
 - AskForSaveWorld, 56
 - BaseCommand, 64
 - BroadcastCommand, 69
 - BroadcastSpawnCharacter, 71
 - BroadcastUpdateCharacter, 73
 - ChangePlayerIDOnServer, 74
 - ChangeZoomFactor, 76
 - ChooseMainTool, 77
 - ClearPath, 78
 - ClientCommand, 80
 - ClientDisconnected, 81
 - Command, 94
 - CommonCommand, 98
 - ConnectTo, 100
 - CopyFolder, 105
 - CopyWall, 106
 - CreateCharacter, 109
 - CutWall, 111
 - DuplicateWorldEntry, 112
 - EndTurn, 125
 - FindPath, 128
 - FirstTurn, 130
 - GameCommand, 133
 - LoadGame, 150
 - LoadMap, 152
 - LoadMapByIndex, 153
 - LoadStartMapFromIndex, 155
 - LoadWorld, 157
 - MoveCharacter, 199
 - MoveFile, 201
 - NetworkCommand, 202
 - NewFolder, 208
 - NewMapFile, 210
 - NewWorld, 211
 - NextTurn, 213
 - NotUndoable, 215
 - ObjectCommand, 227
 - PasteObject, 242
 - PlaceEditorObject, 244
 - PlaceSpecial, 247
 - PlaceWall, 249
 - PreloadAllMaps, 255
 - PreloadWorld, 257
 - Processor, 258
 - Quit, 262
 - ReceiveChangePlayerID, 263
 - ReceiveInitialPlayerInformation, 264
 - ReceiveMap, 266

- ReceivePlayer, 267
- ReceivePlayerModels, 268
- ReceiveSpawnCharacterFrom, 270
- ReceiveUpdateCharacterFrom, 271
- RemoveAllObjects, 273
- RemoveFolder, 275
- RemoveFromChosenList, 277
- RemoveWall, 280
- RemoveWorldEntry, 281
- RenameWorldEntry, 283
- ReplyToRequestForMap, 284
- ReplyToRequestForPlayer, 286
- ReplyToRequestForPlayerModels, 287
- RequestMapFromServer, 288
- RequestPlayerFromServer, 290
- RequestPlayerModelsFromServer, 291
- ResumeEMailGame, 293
- SaveEMailGame, 297
- SaveGame, 299
- SaveMap, 301
- SaveMapAs, 303
- SaveWorld, 304
- SaveWorldAs, 306
- SendInitialPlayerInformation, 307
- SendSpawnCharacterToServer, 308
- SendToClientNextTurn, 310
- SendToServerEndTurn, 311
- SendUpdateCharacterToServer, 313
- ServerCommand, 316
- SetupNewGame, 325
- ShowAbout, 327
- ShowDescription, 328
- ShowGameOptions, 330
- SpawnGameObject, 333
- SpawnPlayerOnCurrentMap, 335
- Undoable, 354
- UseRedo, 359
- UseUndo, 361
- YourTurnInformation, 374
- f
 - AStarMapEntry, 62
- FLOOR
 - ObjectTile, 238
- FROM_CLIENT_NEXT_TURN
 - NetworkModel.h, 448
- FROM_SERVER_NEXT_TURN
 - NetworkModel.h, 448
- file
 - FileManager, 127
 - NewMapFile, 210
- fileChanged
 - WorldModel, 367
- fileClose
 - FileManager, 126
- fileExists
 - FileManager, 126
- fileInfo
 - Serialize, 314
- FileManager, 125
 - deserialize, 126
 - file, 127
 - fileClose, 126
 - fileExists, 126
 - FileManager, 126
 - fileOpen, 126
 - fileRemove, 126
 - FileManager, 126
 - refFile, 127
 - serialize, 127
 - setFilePath, 127
- fileManager
 - LoadGame, 150
 - LoadMap, 152
 - LoadWorld, 157
 - NewMapFile, 210
 - NewWorld, 212
 - PreloadAllMaps, 256
 - PreloadWorld, 257
 - ResumeEMailGame, 293
 - SaveEMailGame, 297
 - SaveGame, 299
 - SaveMap, 302
 - SaveMapAs, 303
 - SaveWorld, 305
 - SaveWorldAs, 306
- fileMenu
 - EditorMenu, 116
- fileOpen
 - FileManager, 126
- fileRemove
 - FileManager, 126
- fileTree
 - WorldView, 371
- FindPath, 127
 - aStarMap, 129
 - execute, 128
 - FindPath, 128
 - FindPath, 128
 - from, 129
 - to, 129
- firstTimeInit
 - MapModel, 179
- FirstTurn, 129
 - choosableCharacterWidgets, 131
 - chooseCharacterDialog, 131
 - chooseCharacterLayout, 131
 - done, 130
 - execute, 130
 - FirstTurn, 130
 - FirstTurn, 130
 - inactivePlayerChosen, 131
 - inactivePlayers, 131
 - indexChanged, 131
 - init, 131
 - newCharacterLayout, 131
 - newCharacterWidget, 131

- players, 131
- firstTurn
 - ClientGameModel, 85
 - EmailGameModel, 122
 - GameModel, 138
 - HotseatGameModel, 145
 - ServerGameModel, 319
- floor
 - StackModel, 347
- floorChanged
 - StackModel, 341
- floorMenu
 - MouseEditorInput, 192
- folder
 - RemoveFolder, 275
- from
 - CopyFolder, 105
 - FindPath, 129
 - MoveCharacter, 199
- FromClient
 - NetworkModel.h, 448
- fromClient
 - WaitingForDataDialog, 362
- fromGameType
 - SaveGameFile, 300
- fromIndex
 - MoveFile, 201
- fromMapModel
 - AStarMap, 57
- FromServer
 - NetworkModel.h, 448
- fromServer
 - WaitingForDataDialog, 362
- fromStackModel
 - AStarMapEntry, 60
- g
 - AStarMapEntry, 62
- GAME_INFO_SUFFIX
 - PIDGIRL, 41
- GUI, 30
- Game, 32
- GameCommand, 131
 - execute, 133
 - GameCommand, 133
 - GameCommand, 133
 - isExecutable, 133
 - quit, 133
- GameModel, 133
 - addNewPlayer, 136
 - addtoPreloadedMaps, 137
 - beginTurn, 137
 - changePlayerID, 137
 - convertFromDifferentGameModel, 137
 - deactivatePlayer, 137
 - deletePlayerModels, 138
 - deserialize, 138
 - endTurn, 138
 - firstTurn, 138
 - GameModel, 136
 - GameModel, 136
 - getCurrentPlayerID, 138
 - getGameType, 138
 - getPlayerChoosesStartLocation, 138
 - getPlayerID, 138
 - getPlayerModels, 138
 - getPreloadedMaps, 138
 - isYourTurn, 138
 - loadMap, 138, 140
 - nextTurn, 140
 - playerChoosesStartLocation, 141
 - playerID, 141
 - playerModels, 142
 - refCurrentMap, 140
 - refPlayer, 140
 - refPlayerModel, 140
 - refPreloadedMap, 140
 - requestedData, 142
 - requestedObject, 142
 - serialize, 140
 - setCurrentMap, 140
 - setNewMapSignal, 140
 - setPlayerChoosesStartLocation, 141
 - setPlayerID, 141
 - setPlayerModels, 141
 - setRequestedData, 141
 - setRequestedObject, 141
 - spawnCharacter, 141
 - updateCharacter, 141
- gameModel
 - Command, 96
 - MainViewMap, 166
 - MainViewStack, 169
 - MainWindow, 172
 - MouseInput, 196
 - Processor, 260
 - SaveGameFile, 300
- gameModelPointer
 - LoadGame, 150
 - SetupNewGame, 326
- gameOptionsDialog
 - ShowGameOptions, 331
- gameOptionsLayout
 - ShowGameOptions, 331
- gameSetupDialog
 - SetupNewGame, 326
- gameSetupLayout
 - SetupNewGame, 326
- gameStartButton
 - EmailSetupWidget, 123
 - NetworkSetupWidget, 207
 - SoloHotseatSetupWidget, 332
- GameType
 - PIDGIRL, 40
- GeneralSetupWidget, 142
 - GeneralSetupWidget, 143
 - generalSetupWidgetLayout, 143

- GeneralSetupWidget, 143
 - playerChoosesStartLocation, 143
- generalSetupWidget
 - SetupNewGame, 326
- generalSetupWidgetLayout
 - GeneralSetupWidget, 143
- generateKeys
 - RSAMethod, 295
- geometryPropeties
 - MapPropertiesDialog, 186
- geometryPropetiesLayout
 - MapPropertiesDialog, 186
- getButtonClicked
 - YesNoCancelDialog, 372
- getCommandIcon
 - BaseCommand, 65
- getCommandList
 - BrickWall, 67
 - ObjectBase, 220
 - Player, 251
 - StartPosition, 349
- getConnectedPlayerIDs
 - ClientModel, 89
 - NetworkModel, 205
 - ServerModel, 322
- getConnectionList
 - ClientModel, 89
 - NetworkModel, 205
 - ServerModel, 322
- getCurrentHeight
 - MapModel, 179
- getCurrentImagePath
 - ObjectBase, 220
- getCurrentPath
 - MapModel, 179
- getCurrentPlayerID
 - GameModel, 138
 - ServerGameModel, 319
- getDescription
 - ObjectBase, 220
- getEmailGameDir
 - MailGameInfo, 163
- getEncryptedText
 - MailGameInfo, 163
- getF
 - AStarMapEntry, 60
- getFileInfo
 - Serialize, 314
- getG
 - AStarMapEntry, 60
- getGameType
 - ClientGameModel, 85
 - EmailGameModel, 122
 - GameModel, 138
 - HotseatGameModel, 145
 - ServerGameModel, 319
- getGridLineSize
 - MapModel, 179
- getHSquareCount
 - MapModel, 179
- getHeight
 - StackModel, 341
- getID
 - PlayerModel, 253
- getImagePathList
 - ObjectBase, 220
- getIsEditor
 - EditorModel, 117
- getIsVisible
 - ObjectBase, 220
- getLocalEmailGamePath
 - MailGameInfo, 163
- getLocalMapPath
 - MapModel, 179
 - PlayerModel, 253
- getMapHeight
 - AStarMap, 58
- getMapSize
 - MapModel, 179
- getMapWidth
 - AStarMap, 58
- getMaxAmountPerStack
 - ObjectSpecial, 236
- getMaxCipherTextLength
 - RSAMethod, 295
- getMaxHeight
 - MapModel, 179
- getMaxPlainTextLength
 - RSAMethod, 295
- getNewSpawnIDFor
 - ObjectID, 229
- getNewSpawnIDFromMap
 - ObjectID, 229
- getObjectID
 - BrickWall, 67
 - ObjectCharacter, 225
 - ObjectItem, 232
 - ObjectSpecial, 236
 - ObjectTile, 239
 - Player, 251
 - StartPosition, 349
- getObjectIDNumber
 - ObjectBase, 220
 - ObjectCharacter, 225
 - ObjectItem, 232
 - ObjectSpecial, 236
 - ObjectTile, 239
- getObjectName
 - ObjectBase, 220
- getPath
 - AStarMap, 58
 - ObjectImage, 231
- getPathIndication
 - StackModel, 341
- getPlayerChoosesStartLocation
 - GameModel, 138

- getPlayerID
 - ClientSocket, 92
 - GameModel, 138
- getPlayerModels
 - ClientGameModel, 85
 - GameModel, 138
- getPosition
 - AStarMapEntry, 60
- getPossibleNeighborsOf
 - AStarMap, 58
- getPreloadedMaps
 - ClientGameModel, 85
 - GameModel, 138
 - LocalGameModel, 160
- getSaveDir
 - MailGameInfo, 163
- getSaveDirAbsolutePath
 - MailGameInfo, 163
- getSpawnID
 - ObjectBase, 221
- getSquarePos
 - StackModel, 341
- getSquareSize
 - MapModel, 179
- getSquareSizeZoomFactor
 - MapModel, 179
- getStartMapWorldPaths
 - WorldModel, 367
- getTileType
 - ObjectTile, 239
- getTurnCount
 - EMailGameModel, 122
- getType
 - ObjectBase, 221
 - ObjectCharacter, 225
 - ObjectItem, 232
 - ObjectSpecial, 236
 - ObjectTile, 239
- getUsedImageIndex
 - ObjectBase, 221
- getVSquareCount
 - MapModel, 179
- getWeight
 - AStarMapEntry, 60
- getX
 - ObjectBase, 221
- getY
 - ObjectBase, 221
- getZ
 - ObjectBase, 221
- gotDisconnected
 - ClientSocket, 92
- gotReadyRead
 - ClientSocket, 92
- gridChanged
 - MapModel, 179
- gridLayout
 - EditorTools, 120
- gridLineSize
 - MapModel, 184
- gridLineSizeChanged
 - MapModel, 179
- HOTSEAT
 - PIDGIRL, 40
- hasBeenReceived
 - WaitingForDataDialog, 362
- hasCharacter
 - StackModel, 341
- hasCharacterSpawned
 - PlayerModel, 253
- hasEMailGameDirSet
 - MailGameInfo, 163
- hasFloor
 - StackModel, 341
- hasInfinityWeight
 - AStarMapEntry, 60
- hasOnlySavedMapChanges
 - UndoRedo, 357
- hasOnlySavedWorldChanges
 - UndoRedo, 357
- hasSaveDirSet
 - MailGameInfo, 163
- hasSpaceForSpecial
 - StackModel, 341
- hasStartPositions
 - MapModel, 179
- hasWall
 - StackModel, 341
- height
 - StackModel, 347
- heightSizeChanged
 - MapModel, 179
- helpMenu
 - EditorMenu, 116
- heuristicDistanceTo
 - AStarMapEntry, 60
- horizontalDirection
 - ApplyMapGeometryChanges, 51
- horizontalNewValue
 - ApplyMapGeometryChanges, 51
- horizontalSizeChanged
 - MapModel, 179
- horizontalSquareSideComboBox
 - MapPropertiesDialog, 186
- horizontalSquareSpinBox
 - MapPropertiesDialog, 186
- HotseatGameModel, 143
 - beginTurn, 144
 - changePlayerID, 144
 - convertFromDifferentGameModel, 144
 - deserialize, 145
 - endTurn, 145
 - firstTurn, 145
 - getGameType, 145
 - HotseatGameModel, 144
 - HotseatGameModel, 144

- nextPlayerID, 145
- nextTurn, 145
- serialize, 145
- ID
 - PlayerModel, 254
- IP_ADDRESS_REG_EXP_STRING
 - PIDGIRL, 41
- IPAddressEdit
 - ConnectTo, 101
- IPAddressLabel
 - ConnectTo, 101
- IPAddressValidator
 - ConnectTo, 101
- ITEM
 - ObjectBase, 219
- IdleEditorMouseState, 145
 - IdleEditorMouseState, 146
 - IdleEditorMouseState, 146
 - nextState, 146
- IdleGameMouseState, 146
 - IdleGameMouseState, 147
 - IdleGameMouseState, 147
 - nextState, 147
- imageList
 - ObjectBase, 223
- inactivePlayerChosen
 - FirstTurn, 131
- inactivePlayers
 - FirstTurn, 131
- index
 - DuplicateWorldEntry, 112
 - NewFolder, 209
 - NewMapFile, 210
 - RemoveWorldEntry, 282
 - RenameWorldEntry, 283
- indexChanged
 - FirstTurn, 131
- indexDoubleClicked
 - WorldFileTree, 364
- informationWidget
 - CreateCharacter, 109
- informationWidgetLayout
 - CreateCharacter, 109
- init
 - AStarMapEntry, 60
 - CreateCharacter, 109
 - FirstTurn, 131
 - LoadGame, 150
 - MainWindow, 171
 - PlayerModel, 253
 - RSAMethod, 295
 - ShowGameOptions, 330
- initEmailGame
 - SetupNewGame, 325
- initGameSetupDialog
 - SetupNewGame, 325
- initMapModel
 - MapModel, 179
- initNetwork
 - SetupNewGame, 326
- initObjectBase
 - ObjectBase, 221
- initObjectCharacter
 - ObjectCharacter, 225
- initObjectItem
 - ObjectItem, 232
- initObjectSpecial
 - ObjectSpecial, 236
- initObjectTile
 - ObjectTile, 239
- initPlayer
 - Player, 251
- initSoloHotseat
 - SetupNewGame, 326
- initStack
 - StackModel, 342
- initToConnectDialog
 - ConnectTo, 100
- initializeEditor
 - MainWindow, 171
- initializeGame
 - MainWindow, 171
- initializeMailGameInfo
 - MailGameInfo, 164
- initializeToolButtons
 - EditorTools, 119
- ipAddress
 - ClientModel, 91
- isActivePlayer
 - PlayerModel, 253
- isClearingUndoRedo
 - BaseCommand, 65
 - Command, 95
 - NotUndoable, 215
 - Undoable, 354
- isClient
 - ClientModel, 90
 - NetworkModel, 205
 - ServerModel, 322
- isEditor
 - EditorModel, 118
- isEmpty
 - StackModel, 342
- isErrorEnabled
 - BaseCommand, 65
- isExecutable
 - BaseCommand, 65
 - ClientCommand, 80
 - Command, 95
 - CommonCommand, 98
 - GameCommand, 133
 - NetworkCommand, 203
 - NotUndoable, 216
 - ObjectCommand, 227
 - Processor, 260
 - ServerCommand, 316

- Undoable, 355
- isServer
 - ClientModel, 90
 - NetworkModel, 205
 - ServerModel, 322
- isSet
 - ObjectBase, 221
- isUndoable
 - BaseCommand, 65
 - Command, 95
 - NotUndoable, 216
 - Undoable, 355
- isVisible
 - ObjectBase, 223
- isVisibleInGame
 - ObjectSpecial, 236
- isYourTurn
 - ClientGameModel, 85
 - GameModel, 138
 - LocalGameModel, 161
 - ServerGameModel, 319
- ItemID
 - ObjectID, 228
- itemList
 - StackModel, 347
- itemListChanged
 - StackModel, 342
- itemListSize
 - StackModel, 342
- itemSpawnIDMap
 - ObjectID, 229
- itemsMenu
 - MouseEditorInput, 192
- KNIFE
 - ObjectID, 229
- leaveGame
 - ShowGameOptions, 330
- leaveGameButton
 - ShowGameOptions, 331
- Left
 - MapModel, 176
- List, 148
 - List, 148
 - mouseMoveEvent, 148
 - mousePressEvent, 148
- listen
 - ServerModel, 322
- load
 - RSAMethod, 295
- LoadGame, 149
 - execute, 150
 - fileManager, 150
 - gameModelPointer, 150
 - init, 150
 - LoadGame, 150
 - loadNetworkGame, 150
 - loadSetupDialog, 150
- loadSetupLayout, 150
- loadSoloHotseatGame, 150
- LoadGame, 150
 - networkSetupWidget, 150
 - saveFilePath, 150
 - soloHotseatSetupWidget, 150
- loadImage
 - ObjectBase, 221
- loadImages
 - ObjectBase, 221
- loadKey
 - RSAMethod, 295
- loadKeys
 - RSAMethod, 295
- LoadMap, 151
 - ask, 152
 - execute, 152
 - fileManager, 152
 - LoadMap, 152
 - LoadMap, 152
 - path, 152
- loadMap
 - ClientGameModel, 85
 - GameModel, 138, 140
 - LocalGameModel, 161
- LoadMapByIndex, 152
 - ask, 154
 - execute, 153
 - LoadMapByIndex, 153
 - LoadMapByIndex, 153
 - modelIndex, 154
- loadNetworkGame
 - LoadGame, 150
- loadSetupDialog
 - LoadGame, 150
- loadSetupLayout
 - LoadGame, 150
- loadSoloHotseatGame
 - LoadGame, 150
- LoadStartMapFromIndex, 154
 - execute, 155
 - LoadStartMapFromIndex, 155
 - LoadStartMapFromIndex, 155
 - startMapIndex, 155
- LoadWorld, 155
 - execute, 157
 - fileManager, 157
 - LoadWorld, 157
 - loadWorld, 157
 - LoadWorld, 157
 - mayAsk, 157
 - resetToStandardMap, 157
- loadWorld
 - EditorMenu, 114
 - LoadWorld, 157
- loadWorldAction
 - EditorMenu, 116
- LocalFilePaths, 157

- convertToFileName, 158
- convertToFileNameVector, 158
- LocalFilePaths, 158
- LocalFilePaths, 158
- LocalGameModel, 158
 - addNewPlayer, 160
 - addtoPreloadedMaps, 160
 - convertFromDifferentGameModel, 160
 - deactivatePlayer, 160
 - deserialize, 160
 - getPreloadedMaps, 160
 - isYourTurn, 161
 - loadMap, 161
 - LocalGameModel, 160
 - LocalGameModel, 160
 - preloadedMaps, 162
 - processor, 162
 - refCurrentMap, 161
 - refPlayer, 161
 - refPlayerModel, 161
 - refPreloadedMap, 161
 - serialize, 161
 - setCurrentMap, 161
- localMapPath
 - MapModel, 184
 - RequestMapFromServer, 289
- localMapPathToAbsoluteMapPath
 - WorldModel, 367
- MAP_SUFFIX
 - PIDGIRL, 41
- MAX_EMAIL_PLAYER_COUNT
 - PIDGIRL, 42
- MAX_FILE_ACCESS_RETRIES
 - PIDGIRL, 42
- MAX_GRID_LINE_SIZE
 - PIDGIRL, 42
- MAX_HEIGHT
 - PIDGIRL, 42
- MAX_HORIZONTAL_SQUARE_COUNT
 - PIDGIRL, 42
- MAX_PLAYER_NAME_LENGTH
 - PIDGIRL, 42
- MAX_SQUARE_SIZE
 - PIDGIRL, 42
- MAX_UNDO_REDO_SIZE
 - PIDGIRL, 42
- MAX_VERTICAL_SQUARE_COUNT
 - PIDGIRL, 42
- MAX_ZOOM_FACTOR
 - PIDGIRL, 42
- MIN_GRID_LINE_SIZE
 - PIDGIRL, 42
- MIN_HEIGHT
 - PIDGIRL, 42
- MIN_HORIZONTAL_SQUARE_COUNT
 - PIDGIRL, 42
- MIN_PLAYER_NAME_LENGTH
 - PIDGIRL, 43
- MIN_SQUARE_SIZE
 - PIDGIRL, 43
- MIN_VERTICAL_SQUARE_COUNT
 - PIDGIRL, 43
- MIN_ZOOM_FACTOR
 - PIDGIRL, 43
- MailGameInfo, 162
 - authenticatePlayer, 163
 - cipherText, 164
 - deserialize, 163
 - emailGameDir, 164
 - emailGameDirSet, 164
 - getEmailGameDir, 163
 - getEncryptedText, 163
 - getLocalEmailGamePath, 163
 - getSaveDir, 163
 - getSaveDirAbsolutePath, 163
 - hasEmailGameDirSet, 163
 - hasSaveDirSet, 163
 - initializeMailGameInfo, 164
 - MailGameInfo, 163
 - MailGameInfo, 163
 - saveDir, 164
 - saveDirSet, 164
 - serialize, 164
 - setLocalEmailGamePath, 164
 - setSaveDir, 164
- mailGameInfo
 - SaveEmailGame, 297
- mailGameInfos
 - EmailGameModel, 122
- main
 - main.cpp, 435
- main.cpp
 - checkLocale, 435
 - main, 435
- MainToolButton
 - PIDGIRL, 40
- mainToolObject
 - EditorModel, 118
- MainViewMap, 164
 - createStackGrid, 165
 - deleteStackGrid, 166
 - editorModel, 166
 - gameModel, 166
 - MainViewMap, 165
 - MainViewMap, 165
 - mapModel, 166
 - mapSizeUpdated, 166
 - mouseInput, 166
 - paintEvent, 166
 - paintGrid, 166
 - processor, 166
 - showContextMenu, 166
 - stackGrid, 166
 - updateMap, 166
- mainViewMap
 - MainWindow, 172

- MainViewStack, 166
 - drawDirectionIndicator, 168
 - drawObject, 168
 - editorModel, 169
 - gameModel, 169
 - MainViewStack, 168
 - MainViewStack, 168
 - mapModel, 169
 - mouseInput, 169
 - mouseMoveEvent, 168
 - mousePressEvent, 168
 - mouseReleaseEvent, 168
 - paintEvent, 168
 - processor, 169
 - setStackModel, 168
 - stackModel, 169
 - updateAll, 168
 - updateSize, 169
- MainWindow, 169
 - closeEvent, 171
 - createMainViewMap, 171
 - done, 172
 - editorMenu, 172
 - editorModel, 172
 - editorTools, 172
 - editorToolsDock, 172
 - gameModel, 172
 - init, 171
 - initializeEditor, 171
 - initializeGame, 171
 - mainViewMap, 172
 - MainWindow, 171
 - MainWindow, 171
 - mapModel, 172
 - processor, 172
 - setNewMap, 171
 - updateEditorSize, 172
 - updateSize, 172
 - updateWindowTitle, 172
 - windowSizeChanged, 172
 - worldModel, 172
 - worldView, 172
 - worldViewDock, 172
- manageStartingMaps
 - WorldPropertiesDialog, 369
- manageStartingMapsLayout
 - WorldPropertiesDialog, 369
- map
 - AStarMap, 58
- MapInfos, 173
- MapModel, 173
 - addPlayerStack, 177
 - addStartPositionStack, 177
 - appendHeightLeft, 177
 - appendHeightRight, 177
 - appendHorizontalLeft, 177
 - appendHorizontalRight, 177
 - appendVerticalLeft, 177
 - appendVerticalRight, 178
 - clearPath, 178
 - clearStackModelGrid, 178
 - clipboardObject, 184
 - clipboardObjectSet, 178
 - correctPositionEntries, 178
 - createStackModelAt, 178
 - currentHeight, 184
 - currentPath, 184
 - deleteStackModelAt, 178
 - deserialize, 178
 - firstTimeInit, 179
 - getCurrentHeight, 179
 - getCurrentPath, 179
 - getGridLineSize, 179
 - getHSquareCount, 179
 - getLocalMapPath, 179
 - getMapSize, 179
 - getMaxHeight, 179
 - getSquareSize, 179
 - getSquareSizeZoomFactor, 179
 - getVSquareCount, 179
 - gridChanged, 179
 - gridLineSize, 184
 - gridLineSizeChanged, 179
 - hasStartPosition, 179
 - heightSizeChanged, 179
 - horizontalSizeChanged, 179
 - initMapModel, 179
 - Left, 176
 - localMapPath, 184
 - MapModel, 176
 - mapPathChanged, 179
 - MapModel, 176
 - playerStacks, 184
 - refClipboardObject, 179
 - refDestinationStackModel, 179
 - refPlayerStack, 179
 - refStackModel, 179
 - refStartPositionStack, 179
 - removeHeightLeft, 179
 - removeHeightRight, 181
 - removeHorizontalLeft, 181
 - removeHorizontalRight, 181
 - removePlayerStack, 181
 - removeStackModelAt, 181
 - removeStartPositionStack, 181
 - removeVerticalLeft, 181
 - removeVerticalRight, 182
 - resizeStackModelGrid, 182
 - Right, 176
 - serialize, 182
 - setClipboardObject, 182
 - setConvertedPath, 182
 - setCurrentHeight, 182
 - setGridLineSize, 182
 - setHSquareCount, 182
 - setLocalMapPath, 182

- setMaxHeight, 182
- setNewStandardMap, 184
- setSquareSizeZoomFactor, 184
- setVSquareCount, 184
- Side, 176
- squareSize, 184
- squareSizeChanged, 184
- squareSizeZoomFactor, 184
- stackModelGrid, 185
- startPositionStacks, 185
- updateSquareSize, 184
- verticalSizeChanged, 184
- mapModel
 - Command, 96
 - MainViewMap, 166
 - MainViewStack, 169
 - MainWindow, 172
 - MouseInput, 196
 - Processor, 260
- mapPathChanged
 - MapModel, 179
- mapPropertiesAction
 - EditorMenu, 116
- MapPropertiesDialog, 185
 - applyChanges, 186
 - geometryPropeties, 186
 - geometryPropetiesLayout, 186
 - horizontalSquareSideComboBox, 186
 - horizontalSquareSpinBox, 186
 - MapPropertiesDialog, 186
 - MapPropertiesDialog, 186
 - maxHeightComboBox, 186
 - maxHeightSpinBox, 186
 - optionsLayout, 186
 - processor, 186
 - updateHorizontal, 186
 - updateMaxHeight, 186
 - updateVertical, 186
 - verticalSquareSideComboBox, 186
 - verticalSquareSpinBox, 186
- mapPropertiesDialog
 - EditorMenu, 116
- mapSizeUpdated
 - MainViewMap, 166
- maxAmountPerStack
 - ObjectSpecial, 237
- maxCipherTextLength
 - RSAMethod, 296
- maxHeightComboBox
 - MapPropertiesDialog, 186
- maxHeightDirection
 - ApplyMapGeometryChanges, 51
- maxHeightNewValue
 - ApplyMapGeometryChanges, 51
- maxHeightSpinBox
 - MapPropertiesDialog, 186
- maxPlainTextLength
 - RSAMethod, 296
- mayAsk
 - LoadWorld, 157
 - RemoveWorldEntry, 282
- messageBox
 - WaitingForDataDialog, 363
- Model, 31
- modelIndex
 - LoadMapByIndex, 154
- MouseEditorInput, 186
 - characterMenu, 192
 - checkCharacterActions, 189
 - checkFloorActions, 189
 - checkItemsActions, 189
 - checkPasteAction, 189
 - checkRemoveAllAction, 189
 - checkSpecialsActions, 189
 - checkWallActions, 189
 - contextMenuStack, 192
 - copyCharacter, 190
 - copyCharacterAction, 192
 - copyFloor, 190
 - copyFloorAction, 192
 - copyItems, 190
 - copyItemsAction, 192
 - copySpecials, 190
 - copySpecialsAction, 192
 - copyWall, 190
 - copyWallAction, 192
 - cutCharacter, 190
 - cutCharacterAction, 192
 - cutFloor, 190
 - cutFloorAction, 192
 - cutItems, 190
 - cutItemsAction, 192
 - cutSpecials, 190
 - cutSpecialsAction, 192
 - cutWall, 190
 - cutWallAction, 192
 - editCharacter, 190
 - editCharacterAction, 192
 - editFloor, 190
 - editFloorAction, 192
 - editItems, 191
 - editItemsAction, 192
 - editSpecials, 191
 - editSpecialsAction, 192
 - editWall, 191
 - editWallAction, 192
 - floorMenu, 192
 - itemsMenu, 192
 - MouseEditorInput, 189
 - MouseEditorInput, 189
 - nextState, 191
 - pasteObject, 191
 - pasteObjectAction, 192
 - removeAll, 191
 - removeAllAction, 192
 - removeCharacter, 191

- removeCharacterAction, 192
- removeFloor, 191
- removeFloorAction, 192
- removeItems, 191
- removeItemsAction, 192
- removeSpecials, 191
- removeSpecialsAction, 192
- removeWall, 191
- removeWallAction, 192
- showEditorContextMenu, 191
- specialsMenu, 192
- wallMenu, 193
- MouseEventType
 - MouseState, 197
- MouseGameInput, 193
 - endTurn, 194
 - MouseGameInput, 194
 - MouseGameInput, 194
 - nextState, 194
 - showGameContextMenu, 194
 - showGameOptions, 194
- MouseInput, 194
 - ~MouseInput, 195
 - contextMenu, 195
 - currentMouseState, 195
 - deleteContextMenu, 195
 - gameModel, 196
 - mapModel, 196
 - MouseInput, 195
 - mousePressed, 195
 - mouseReleased, 195
 - MouseInput, 195
 - nextState, 195
 - processor, 196
 - showContextMenu, 195
- mouseInput
 - MainViewMap, 166
 - MainViewStack, 169
- mouseMoveEvent
 - List, 148
 - MainViewStack, 168
- mousePressEvent
 - List, 148
 - MainViewStack, 168
 - WorldFileTree, 364
- mousePressed
 - MouseInput, 195
- mouseReleaseEvent
 - MainViewStack, 168
 - WorldFileTree, 365
- mouseReleased
 - MouseInput, 195
- MouseState, 196
 - ~MouseState, 197
 - MouseEventType, 197
 - MouseState, 197
 - MouseState, 197
 - nextState, 197
- PRESSED, 197
- processor, 198
- RELEASED, 197
- refTarget, 198
- setTarget, 198
- showContextMenu, 198
- target, 198
- MoveCharacter, 198
 - execute, 199
 - from, 199
 - MoveCharacter, 199
 - MoveCharacter, 199
 - to, 199
- MoveFile, 200
 - execute, 201
 - fromIndex, 201
 - MoveFile, 201
 - MoveFile, 201
 - toIndex, 201
- N
 - StackModel, 338
- NAME_REG_EXP_STRING
 - PIDGIRL, 43
- NE
 - StackModel, 339
- NETWORK
 - PIDGIRL, 40
- NO_BUTTON
 - YesNoCancelDialog, 372
- NONE
 - StackModel, 338
- NW
 - StackModel, 338
- Network, 34
- NetworkCommand, 201
 - execute, 202
 - isExecutable, 203
 - NetworkCommand, 202
 - NetworkCommand, 202
- NetworkModel, 203
 - ~NetworkModel, 204
 - allDataReceivedFromClient, 204
 - allDataReceivedFromServer, 204
 - changeClientID, 204
 - clientDisconnect, 205
 - connectionEstablished, 205
 - getConnectedPlayerIDs, 205
 - getConnectionList, 205
 - isClient, 205
 - isServer, 205
 - NetworkModel, 204
 - NetworkModel, 204
 - port, 206
 - receiveData, 205
 - sendData, 206
- networkModel
 - Command, 96
 - Processor, 260

- WaitingForDataDialog, 363
- NetworkModel.h
 - BROADCAST_SPAWN_CHARACTER, 448
 - BROADCAST_UPDATE_CHARACTER, 448
 - CHANGE_PLAYER_ID, 448
 - END_TURN, 448
 - FROM_CLIENT_NEXT_TURN, 448
 - FROM_SERVER_NEXT_TURN, 448
 - FromClient, 448
 - FromServer, 448
 - RECEIVE_INITIAL_PLAYER_INFORMATION, 448
 - RECEIVE_REQUEST_FOR_PLAYER, 448
 - RECEIVE_REQUEST_FOR_PLAYER_MODELS, 448
 - RECEIVE_REQUEST_FOR_SPECIFIC_MAP, 448
 - SEND_PLAYER_MODELS, 448
 - SEND_SPAWN_CHARACTER, 448
 - SEND_SPECIFIC_MAP, 448
 - SEND_SPECIFIC_PLAYER, 448
 - SEND_UPDATE_CHARACTER, 448
- NetworkSetupWidget, 206
 - gameStartButton, 207
 - NetworkSetupWidget, 207
 - networkWidgetLayout, 207
 - NetworkSetupWidget, 207
 - portEdit, 207
 - portLabel, 207
 - portValidator, 207
- networkSetupWidget
 - LoadGame, 150
 - SetupNewGame, 326
- networkWidgetLayout
 - NetworkSetupWidget, 207
- newCharacter
 - BroadcastSpawnCharacter, 71
 - BroadcastUpdateCharacter, 73
 - SendSpawnCharacterToServer, 309
 - SendUpdateCharacterToServer, 313
- newCharacterLayout
 - FirstTurn, 131
- newCharacterWidget
 - FirstTurn, 131
- newConnection
 - ServerModel, 322
- newFactor
 - ChangeZoomFactor, 76
- NewFolder, 207
 - execute, 208
 - index, 209
 - NewFolder, 208
 - NewFolder, 208
- newFolder
 - WorldFileTree, 365
- newFolderAction
 - WorldFileTree, 365
- newID
 - ChangePlayerIDOnServer, 75
- newMap
 - WorldFileTree, 365
- newMapAction
 - WorldFileTree, 365
- NewMapFile, 209
 - dir, 210
 - execute, 210
 - file, 210
 - fileManager, 210
 - index, 210
 - NewMapFile, 210
 - NewMapFile, 210
- newSeed
 - PIDGIRL, 40
- NewWorld, 210
 - execute, 211
 - fileManager, 212
 - NewWorld, 211
 - NewWorld, 211
- newWorldAction
 - EditorMenu, 116
- nextCurrentPlayer
 - ServerGameModel, 319
- nextID
 - PIDGIRL, 40
- nextPlayerID
 - EMailGameModel, 122
 - HotseatGameModel, 145
- nextState
 - ContextEditorMenuState, 102
 - ContextGameMenuState, 103
 - IdleEditorMouseState, 146
 - IdleGameMouseState, 147
 - MouseEditorInput, 191
 - MouseGameInput, 194
 - MouseInput, 195
 - MouseState, 197
 - PlaceObjectsState, 245
 - RemoveObjectsState, 278
 - TargetChosenState, 350
 - TargetConfirmedState, 352
 - TargetMoveState, 353
- NextTurn, 212
 - execute, 213
 - NextTurn, 213
 - NextTurn, 213
- nextTurn
 - ClientGameModel, 86
 - EMailGameModel, 122
 - GameModel, 140
 - HotseatGameModel, 145
 - ServerGameModel, 319
- noButton
 - YesNoCancelDialog, 372
- NotUndoable, 213
 - clearsUndoRedo, 216
 - execute, 215
 - isClearingUndoRedo, 215

- isExecutable, 216
 - isUndoable, 216
 - NotUndoable, 215
 - NotUndoable, 215
 - redo, 216
 - undo, 216
- Object, 35
- object
- ObjectCommand, 227
 - ObjectMenu, 234
- ObjectBase, 216
- appendImage, 219
 - appendImages, 219
 - CHARACTER, 219
 - clearImageList, 220
 - copyBase, 220
 - createInformationWidget, 220
 - description, 223
 - deserialize, 220
 - getCommandList, 220
 - getCurrentImagePath, 220
 - getDescription, 220
 - getImagePathList, 220
 - getIsVisible, 220
 - getObjectIDNumber, 220
 - getObjectName, 220
 - getSpawnID, 221
 - getType, 221
 - getUsedImageIndex, 221
 - getX, 221
 - getY, 221
 - getZ, 221
 - ITEM, 219
 - imageList, 223
 - initObjectBase, 221
 - isSet, 221
 - isVisible, 223
 - loadImage, 221
 - loadImages, 221
 - ObjectBase, 219
 - ObjectType, 219
 - ObjectBase, 219
 - refImage, 221
 - SPECIAL, 219
 - serialize, 222
 - setDescription, 222
 - setIsVisible, 222
 - setPosition, 222
 - setSpawnID, 222
 - setUsedImageIndex, 222
 - spawnID, 223
 - TILE, 219
 - toggleVisibility, 222
 - unsetPosition, 222
 - update, 222
 - updateBase, 222
 - usedImageIndex, 223
 - x, 223
 - y, 223
 - z, 223
- ObjectCharacter, 223
- copy, 225
 - copyBase, 225
 - deserialize, 225
 - getObjectID, 225
 - getObjectIDNumber, 225
 - getType, 225
 - initObjectCharacter, 225
 - ObjectCharacter, 225
 - ObjectCharacter, 225
 - serialize, 225
 - update, 226
 - updateCharacter, 226
- ObjectCommand, 226
- execute, 227
 - isExecutable, 227
 - object, 227
 - ObjectCommand, 227
 - ObjectCommand, 227
- objectFrom
- ObjectFromID, 37
- ObjectFromID, 37
- objectFrom, 37
- ObjectID, 227
- BRICK_WALL, 229
 - CharacterID, 228
 - characterSpawnIDMap, 229
 - deserialize, 229
 - getNewSpawnIDFor, 229
 - getNewSpawnIDFromMap, 229
 - ItemID, 228
 - itemSpawnIDMap, 229
 - KNIFE, 229
 - ObjectID, 229
 - ObjectID, 229
 - PLAYER, 228
 - START_POSITION, 229
 - serialize, 229
 - SpecialID, 229
 - specialSpawnIDMap, 229
 - TileID, 229
 - tileSpawnIDMap, 229
- objectID
- WorldModel, 367
- ObjectImage, 230
- getPath, 231
 - ObjectImage, 230
 - ObjectImage, 230
 - path, 231
- ObjectItem, 231
- copy, 232
 - copyBase, 232
 - deserialize, 232
 - getObjectID, 232
 - getObjectIDNumber, 232
 - getType, 232

- initObjectItem, 232
- ObjectItem, 232
- ObjectItem, 232
- serialize, 232, 233
- update, 233
- updateItem, 233
- ObjectMenu, 233
 - object, 234
 - ObjectMenu, 234
 - ObjectMenu, 234
 - processor, 234
 - triggerCommandByName, 234
- ObjectSpecial, 234
 - copy, 235
 - copyBase, 235
 - deserialize, 235, 236
 - getMaxAmountPerStack, 236
 - getObjectID, 236
 - getObjectIDNumber, 236
 - getType, 236
 - initObjectSpecial, 236
 - isVisibleInGame, 236
 - maxAmountPerStack, 237
 - ObjectSpecial, 235
 - ObjectSpecial, 235
 - serialize, 236
 - setMaxAmountPerStack, 236
 - setVisibleInGame, 236
 - update, 236
 - updateSpecial, 236
 - visibleInGame, 237
- ObjectTile, 237
 - copy, 238
 - copyBase, 238
 - deserialize, 239
 - FLOOR, 238
 - getObjectID, 239
 - getObjectIDNumber, 239
 - getTileType, 239
 - getType, 239
 - initObjectTile, 239
 - ObjectTile, 238
 - ObjectTile, 238
 - serialize, 239
 - setTileType, 239
 - TileType, 238
 - tileType, 240
 - update, 239
 - updateTile, 239
 - WALL, 238
- ObjectType
 - ObjectBase, 219
- okButton
 - OptionsLayout, 241
- oldCharacter
 - BroadcastUpdateCharacter, 73
 - SendUpdateCharacterToServer, 313
- oldWall
 - PlaceWall, 249
 - RemoveWall, 280
- onlySavedMapChanges
 - UndoRedo, 358
- onlySavedWorldChanges
 - UndoRedo, 358
- operator=
 - AStarMapEntry, 62
- operator==
 - AStarMapEntry, 62
- OptionsLayout, 240
 - addPropertyWidget, 241
 - applyButton, 241
 - currentIndexChangedTo, 241
 - okButton, 241
 - OptionsLayout, 241
 - OptionsLayout, 241
 - propertyList, 241
 - refApplyButton, 241
 - refOkButton, 241
 - setPropertyListWidth, 241
 - showCurrentWidget, 241
 - stackLayout, 241
 - stackedProperties, 241
- optionsLayout
 - MapPropertiesDialog, 186
 - WorldPropertiesDialog, 370
- PIDGIRL, 37
 - BRICKWALL, 40
 - done, 41
 - EDITOR_VIEW_HEIGHT, 41
 - EDITOR_VIEW_WIDTH, 41
 - EMAIL, 40
 - EMAIL_FOLDER_NAME, 41
 - EMAIL_SUFFIX, 41
 - GAME_INFO_SUFFIX, 41
 - GameType, 40
 - HOTSEAT, 40
 - IP_ADDRESS_REG_EXP_STRING, 41
 - MAP_SUFFIX, 41
 - MAX_EMAIL_PLAYER_COUNT, 42
 - MAX_FILE_ACCESS_RETRIES, 42
 - MAX_GRID_LINE_SIZE, 42
 - MAX_HEIGHT, 42
 - MAX_HORIZONTAL_SQUARE_COUNT, 42
 - MAX_PLAYER_NAME_LENGTH, 42
 - MAX_SQUARE_SIZE, 42
 - MAX_UNDO_REDO_SIZE, 42
 - MAX_VERTICAL_SQUARE_COUNT, 42
 - MAX_ZOOM_FACTOR, 42
 - MIN_GRID_LINE_SIZE, 42
 - MIN_HEIGHT, 42
 - MIN_HORIZONTAL_SQUARE_COUNT, 42
 - MIN_PLAYER_NAME_LENGTH, 43
 - MIN_SQUARE_SIZE, 43
 - MIN_VERTICAL_SQUARE_COUNT, 43
 - MIN_ZOOM_FACTOR, 43
 - MainToolButton, 40

- NAME_REG_EXP_STRING, 43
- NETWORK, 40
- newSeed, 40
- nextID, 40
- PIDGIRL_VERSION, 43
- RSA_FOLDER_NAME, 43
- RSA_PLAINTEXT_FILE_NAME, 43
- RSA_PRIVATE_KEY_FILE_NAME, 43
- RSA_PUBLIC_KEY_FILE_NAME, 43
- SAVE_FOLDER_NAME, 43
- SAVE_SUFFIX, 43
- SECONDS_UNTIL_RETRYING_FILE_ACCESS, 44
- SEND_DATA_RETRIES, 44
- SERVER_PLAYER_ID, 44
- STANDARD_GRID_LINE_SIZE, 44
- STANDARD_HEIGHT, 44
- STANDARD_HORIZONTAL_SQUARE_COUNT, 44
- STANDARD_SQUARE_SIZE, 44
- STANDARD_VERTICAL_SQUARE_COUNT, 44
- STANDARD_ZOOM_FACTOR, 44
- START_POSITION, 40
- showUsage, 40
- TOOL_BUTTON_COUNT_IN_ROW, 44
- WORLD_FOLDER_NAME, 44
- WORLD_SUFFIX, 44
- writeLog, 40
- writeLogDone, 41
- writeLogFail, 41
- PIDGIRL_VERSION
 - PIDGIRL, 43
- PLAYER
 - ObjectID, 228
- PRESSED
 - MouseState, 197
- paintEvent
 - MainViewMap, 166
 - MainViewStack, 168
- paintGrid
 - MainViewMap, 166
- PasteObject, 241
 - execute, 242
 - PasteObject, 242
 - PasteObject, 242
 - stackModel, 243
- pasteObject
 - MouseEditorInput, 191
- pasteObjectAction
 - MouseEditorInput, 192
- path
 - AStarMap, 58
 - LoadMap, 152
 - ObjectImage, 231
- PathIndication
 - StackModel, 338
- pathIndication
 - StackModel, 347
- pathIndicationChanged
 - StackModel, 342
- PlaceEditorObject, 243
 - execute, 244
 - PlaceEditorObject, 243
 - placeableObject, 244
 - PlaceEditorObject, 243
 - stackModel, 244
- PlaceObject, 244
- PlaceObjectsState, 245
 - alreadyPlaced, 246
 - nextState, 245
 - PlaceObjectsState, 245
 - PlaceObjectsState, 245
- PlaceSpecial, 246
 - execute, 247
 - PlaceSpecial, 247
 - PlaceSpecial, 247
 - redo, 247
 - special, 248
 - stackModel, 248
 - undo, 247
- PlaceWall, 248
 - execute, 249
 - oldWall, 249
 - PlaceWall, 249
 - PlaceWall, 249
 - redo, 249
 - stackModel, 249
 - undo, 249
 - wall, 249
- placeableObject
 - PlaceEditorObject, 244
- Player, 250
 - copy, 251
 - createInformationWidget, 251
 - deserialize, 251
 - getCommandList, 251
 - getObjectID, 251
 - initPlayer, 251
 - Player, 251
 - serialize, 252
 - update, 252
- playerChoosesStartLocation
 - GameModel, 141
 - GeneralSetupWidget, 143
- playerCountLabel
 - EmailSetupWidget, 123
- playerCountSpinBox
 - EmailSetupWidget, 123
- playerID
 - ClientSocket, 92
 - GameModel, 141
 - RequestPlayerFromServer, 290
 - ServerCommand, 316
- playerMapPath
 - PlayerModel, 254
- PlayerModel, 252

- activePlayer, 254
- characterSpawned, 254
- deserialize, 253
- getID, 253
- getLocalMapPath, 253
- hasCharacterSpawned, 253
- ID, 254
- init, 253
- isActivePlayer, 253
- playerMapPath, 254
- PlayerModel, 253
- PlayerModel, 253
- serialize, 253
- setActivePlayer, 254
- setCharacterSpawned, 254
- setPlayerMapPath, 254
- playerModels
 - GameModel, 142
- playerName
 - SaveEMailGame, 297
 - SpawnPlayerOnCurrentMap, 335
- playerNameEdit
 - CreateCharacter, 109
- playerNameValidator
 - CreateCharacter, 109
- playerRemoved
 - StackModel, 342
- playerSet
 - StackModel, 342
- playerStacks
 - MapModel, 184
- players
 - FirstTurn, 131
- port
 - NetworkModel, 206
- portEdit
 - ConnectTo, 101
 - NetworkSetupWidget, 207
- portLabel
 - ConnectTo, 101
 - NetworkSetupWidget, 207
- portValidator
 - ConnectTo, 101
 - NetworkSetupWidget, 207
- position
 - AStarMapEntry, 62
- possibleToReachOriginalState
 - UndoRedo, 358
- preload
 - PreloadAllMaps, 255
- PreloadAllMaps, 254
 - execute, 255
 - fileManager, 256
 - preload, 255
 - PreloadAllMaps, 255
 - PreloadAllMaps, 255
- PreloadWorld, 256
 - execute, 257
 - fileManager, 257
 - PreloadWorld, 257
 - PreloadWorld, 257
- preloadedMaps
 - LocalGameModel, 162
- prependItem
 - StackModel, 342
- prependSpecial
 - StackModel, 342
- privateKey
 - RSAMethod, 296
- privateKeyFilePath
 - RSAMethod, 296
- Processor, 257
 - commandProcessed, 258
 - editorModel, 260
 - execute, 258
 - gameModel, 260
 - isExecutable, 260
 - mapModel, 260
 - networkModel, 260
 - Processor, 258
 - refUndoRedo, 260
 - setEditorModel, 260
 - setGameModel, 260
 - setMapModel, 260
 - setNetworkModel, 260
 - setWorldModel, 260
 - undoRedo, 260
 - worldModel, 260
- processor
 - ClientGameModel, 87
 - ClientModel, 91
 - Command, 96
 - EditorMenu, 116
 - EditorTools, 120
 - LocalGameModel, 162
 - MainViewMap, 166
 - MainViewStack, 169
 - MainWindow, 172
 - MapPropertiesDialog, 186
 - MouseInput, 196
 - MouseState, 198
 - ObjectMenu, 234
 - SaveGameFile, 300
 - ServerModel, 323
 - WorldFileTree, 365
 - WorldPropertiesDialog, 370
 - WorldView, 371
- propertyList
 - OptionsLayout, 241
- publicKey
 - RSAMethod, 296
- publicKeyFilePath
 - RSAMethod, 296
- Quit, 261
 - askToSaveGame, 262
 - execute, 262

- Quit, 261
- quit
 - EditorMenu, 115
 - GameCommand, 133
- quitAction
 - EditorMenu, 116
- RECEIVE_INITIAL_PLAYER_INFORMATION
 - NetworkModel.h, 448
- RECEIVE_REQUEST_FOR_PLAYER
 - NetworkModel.h, 448
- RECEIVE_REQUEST_FOR_PLAYER_MODELS
 - NetworkModel.h, 448
- RECEIVE_REQUEST_FOR_SPECIFIC_MAP
 - NetworkModel.h, 448
- RELEASED
 - MouseState, 197
- RSA, 36
- RSA_FOLDER_NAME
 - PIDGIRL, 43
- RSA_PLAINTEXT_FILE_NAME
 - PIDGIRL, 43
- RSA_PRIVATE_KEY_FILE_NAME
 - PIDGIRL, 43
- RSA_PUBLIC_KEY_FILE_NAME
 - PIDGIRL, 43
- RSAMethod, 293
 - decryptString, 295
 - encryptString, 295
 - generateKeys, 295
 - getMaxCipherTextLength, 295
 - getMaxPlainTextLength, 295
 - init, 295
 - load, 295
 - loadKey, 295
 - loadKeys, 295
 - maxCipherTextLength, 296
 - maxPlainTextLength, 296
 - privateKey, 296
 - privateKeyFilePath, 296
 - publicKey, 296
 - publicKeyFilePath, 296
 - RSAMethod, 294
 - randomNumberGenerator, 296
 - RSAMethod, 294
 - save, 295
 - saveKey, 296
- randomNumberGenerator
 - RSAMethod, 296
- readyRead
 - ClientSocket, 92
- ReceiveChangePlayerID, 262
 - execute, 263
 - ReceiveChangePlayerID, 263
 - ReceiveChangePlayerID, 263
- receiveData
 - ClientModel, 90
 - NetworkModel, 205
 - ServerModel, 322, 323
- ReceiveInitialPlayerInformation, 263
 - execute, 264
 - ReceiveInitialPlayerInformation, 264
 - ReceiveInitialPlayerInformation, 264
- ReceiveMap, 265
 - execute, 266
 - ReceiveMap, 266
 - ReceiveMap, 266
- receivePendingData
 - ClientModel, 90
 - ServerModel, 323
- ReceivePlayer, 266
 - execute, 267
 - ReceivePlayer, 267
 - ReceivePlayer, 267
- ReceivePlayerModels, 267
 - execute, 268
 - ReceivePlayerModels, 268
 - ReceivePlayerModels, 268
- ReceiveSpawnCharacterFrom, 269
 - execute, 270
 - ReceiveSpawnCharacterFrom, 270
 - ReceiveSpawnCharacterFrom, 270
 - senderID, 270
- ReceiveUpdateCharacterFrom, 270
 - execute, 271
 - ReceiveUpdateCharacterFrom, 271
 - ReceiveUpdateCharacterFrom, 271
 - senderID, 271
- redo
 - BaseCommand, 65
 - Command, 95
 - NotUndoable, 216
 - PlaceSpecial, 247
 - PlaceWall, 249
 - RemoveAllObjects, 273
 - RemoveWall, 280
 - Undoable, 355
 - UndoRedo, 357
- redoAction
 - EditorMenu, 116
- redoChanged
 - UndoRedo, 357
- redoList
 - UndoRedo, 358
- redoListSavedState
 - UndoRedo, 358
- refApplyButton
 - OptionsLayout, 241
- refCameFrom
 - AStarMapEntry, 62
- refCharacter
 - StackModel, 342
- refClipboardObject
 - MapModel, 179
- refContextMenu
 - WorldFileTree, 365
- refCurrentMap

- ClientGameModel, 86
- GameModel, 140
- LocalGameModel, 161
- ServerGameModel, 319
- refDestinationStackModel
 - MapModel, 179
- refEntry
 - AStarMap, 58
- refFile
 - FileManager, 127
- refFloor
 - StackModel, 342
- refImage
 - ObjectBase, 221
- refItem
 - StackModel, 342
- refItemList
 - StackModel, 342
- refMailGameInfo
 - EmailGameModel, 122
- refMainObject
 - EditorModel, 117
- refObjectID
 - WorldModel, 367
- refOkButton
 - OptionsLayout, 241
- refPlayer
 - ClientGameModel, 86
 - GameModel, 140
 - LocalGameModel, 161
- refPlayerModel
 - ClientGameModel, 86
 - GameModel, 140
 - LocalGameModel, 161
- refPlayerStack
 - MapModel, 179
- refPreloadedMap
 - ClientGameModel, 86
 - GameModel, 140
 - LocalGameModel, 161
- refSocket
 - ClientSocket, 92
- refSpecial
 - StackModel, 342
- refSpecialList
 - StackModel, 343
- refStackModel
 - MapModel, 179
- refStartPositionStack
 - MapModel, 179
- refTarget
 - MouseState, 198
- refUndoRedo
 - Processor, 260
- refWall
 - StackModel, 343
- refWorldFileInfo
 - WorldModel, 367
- remove
 - WorldFileTree, 365
- removeAction
 - WorldFileTree, 365
- removeAll
 - MouseEditorInput, 191
 - StackModel, 343
- removeAllAction
 - MouseEditorInput, 192
- removeAllItems
 - StackModel, 343
- RemoveAllObjects, 272
 - execute, 273
 - redo, 273
 - RemoveAllObjects, 273
 - RemoveAllObjects, 273
 - stackModel, 273
 - undo, 273
 - undoCharacter, 273
 - undoFloor, 273
 - undoItemList, 273
 - undoSpecialList, 273
 - undoWall, 274
- removeAllSpecials
 - StackModel, 343
- removeButton
 - WorldPropertiesDialog, 370
- removeCharacter
 - MouseEditorInput, 191
 - StackModel, 343
- removeCharacterAction
 - MouseEditorInput, 192
- removeDirRecursive
 - RemoveFolder, 275
- removeEmpty
 - RemoveFolder, 275
- removeFloor
 - MouseEditorInput, 191
 - StackModel, 343
- removeFloorAction
 - MouseEditorInput, 192
- RemoveFolder, 274
 - execute, 275
 - folder, 275
 - removeDirRecursive, 275
 - removeEmpty, 275
 - RemoveFolder, 275
 - RemoveFolder, 275
- RemoveFromChosenList, 276
 - chosenListModel, 277
 - currentIndex, 277
 - execute, 277
 - RemoveFromChosenList, 277
 - RemoveFromChosenList, 277
- removeHeightLeft
 - MapModel, 179
- removeHeightRight
 - MapModel, 181

- removeHorizontalLeft
 - MapModel, 181
- removeHorizontalRight
 - MapModel, 181
- removeItem
 - StackModel, 343
- removeItems
 - MouseEditorInput, 191
- removeItemsAction
 - MouseEditorInput, 192
- RemoveObjectsState, 277
 - alreadyPlaced, 278
 - nextState, 278
 - RemoveObjectsState, 278
 - RemoveObjectsState, 278
- removePathFromChosen
 - WorldPropertiesDialog, 369
- removePlayerStack
 - MapModel, 181
- removeSpecial
 - StackModel, 343
- removeSpecials
 - MouseEditorInput, 191
- removeSpecialsAction
 - MouseEditorInput, 192
- removeStackModelAt
 - MapModel, 181
- removeStartPositionStack
 - MapModel, 181
- removeVerticalLeft
 - MapModel, 181
- removeVerticalRight
 - MapModel, 182
- RemoveWall, 279
 - execute, 280
 - oldWall, 280
 - redo, 280
 - RemoveWall, 280
 - RemoveWall, 280
 - stackModel, 280
 - undo, 280
- removeWall
 - MouseEditorInput, 191
 - StackModel, 343
- removeWallAction
 - MouseEditorInput, 192
- RemoveWorldEntry, 280
 - execute, 281
 - index, 282
 - mayAsk, 282
 - RemoveWorldEntry, 281
 - RemoveWorldEntry, 281
- rename
 - WorldFileTree, 365
- renameAction
 - WorldFileTree, 365
- RenameWorldEntry, 282
 - execute, 283
 - index, 283
 - RenameWorldEntry, 283
 - RenameWorldEntry, 283
- replaceCharacter
 - StackModel, 343
- replaceFloor
 - StackModel, 344
- replaceWall
 - StackModel, 344
- ReplyToRequestForMap, 283
 - execute, 284
 - ReplyToRequestForMap, 284
 - ReplyToRequestForMap, 284
- ReplyToRequestForPlayer, 285
 - execute, 286
 - ReplyToRequestForPlayer, 286
 - ReplyToRequestForPlayer, 286
- ReplyToRequestForPlayerModels, 286
 - execute, 287
 - ReplyToRequestForPlayerModels, 287
 - ReplyToRequestForPlayerModels, 287
- RequestMapFromServer, 287
 - execute, 288
 - localMapPath, 289
 - RequestMapFromServer, 288
 - RequestMapFromServer, 288
- RequestPlayerFromServer, 289
 - execute, 290
 - playerID, 290
 - RequestPlayerFromServer, 290
 - RequestPlayerFromServer, 290
- RequestPlayerModelsFromServer, 290
 - execute, 291
 - RequestPlayerModelsFromServer, 291
 - RequestPlayerModelsFromServer, 291
- requestedData
 - GameModel, 142
- requestedObject
 - GameModel, 142
- resetFileInfo
 - Serialize, 314
- resetPathDirection
 - StackModel, 344
- resetToStandardMap
 - LoadWorld, 157
- resizeStackModelGrid
 - MapModel, 182
- resizeWorldView
 - WorldView, 371
- ResumeEMailGame, 292
 - execute, 293
 - fileManager, 293
 - ResumeEMailGame, 293
 - ResumeEMailGame, 293
- retracePath
 - AStarMap, 58
- Right
 - MapModel, 176

- S
 - StackModel, 338
- SAVE_FOLDER_NAME
 - PIDGIRL, 43
- SAVE_SUFFIX
 - PIDGIRL, 43
- SE
 - StackModel, 339
- SECONDS_UNTIL_RETRYING_FILE_ACCESS
 - PIDGIRL, 44
- SEND_DATA_RETRIES
 - PIDGIRL, 44
- SEND_PLAYER_MODELS
 - NetworkModel.h, 448
- SEND_SPAWN_CHARACTER
 - NetworkModel.h, 448
- SEND_SPECIFIC_MAP
 - NetworkModel.h, 448
- SEND_SPECIFIC_PLAYER
 - NetworkModel.h, 448
- SEND_UPDATE_CHARACTER
 - NetworkModel.h, 448
- SERVER_PLAYER_ID
 - PIDGIRL, 44
- SPECIAL
 - ObjectBase, 219
- STANDARD_GRID_LINE_SIZE
 - PIDGIRL, 44
- STANDARD_HEIGHT
 - PIDGIRL, 44
- STANDARD_HORIZONTAL_SQUARE_COUNT
 - PIDGIRL, 44
- STANDARD_SQUARE_SIZE
 - PIDGIRL, 44
- STANDARD_VERTICAL_SQUARE_COUNT
 - PIDGIRL, 44
- STANDARD_ZOOM_FACTOR
 - PIDGIRL, 44
- START_POSITION
 - ObjectID, 229
 - PIDGIRL, 40
- SW
 - StackModel, 338
- save
 - RSAMethod, 295
- saveDir
 - MailGameInfo, 164
- saveDirChooseButton
 - ShowGameOptions, 331
- saveDirLabel
 - ShowGameOptions, 331
- saveDirLineEdit
 - ShowGameOptions, 331
- saveDirSet
 - MailGameInfo, 164
- SaveEMailGame, 296
 - execute, 297
 - fileManager, 297
- mailGameInfo, 297
 - playerName, 297
- SaveEMailGame, 297
 - turnCount, 297
- saveFilePath
 - LoadGame, 150
- SaveGame, 298
 - execute, 299
 - fileManager, 299
 - SaveGame, 298
 - SaveGame, 298
- saveGame
 - ShowGameOptions, 330
- saveGameButton
 - ShowGameOptions, 331
- SaveGameFile, 299
 - deserialize, 300
 - fromGameType, 300
 - gameModel, 300
 - processor, 300
 - SaveGameFile, 300
 - SaveGameFile, 300
 - serialize, 300
 - worldModel, 300
- saveKey
 - RSAMethod, 296
- SaveMap, 301
 - execute, 301
 - fileManager, 302
 - SaveMap, 301
 - SaveMap, 301
- saveMap
 - EditorMenu, 115
- saveMapAction
 - EditorMenu, 116
- SaveMapAs, 302
 - execute, 303
 - fileManager, 303
 - SaveMapAs, 303
 - SaveMapAs, 303
- saveMapAs
 - EditorMenu, 115
- saveMapAsAction
 - EditorMenu, 116
- SaveWorld, 303
 - execute, 304
 - fileManager, 305
 - SaveWorld, 304
 - SaveWorld, 304
- saveWorld
 - EditorMenu, 115
- saveWorldAction
 - EditorMenu, 116
- SaveWorldAs, 305
 - execute, 306
 - fileManager, 306
 - SaveWorldAs, 306

- SaveWorldAs, 306
- saveWorldAs
 - EditorMenu, 115
- saveWorldAsAction
 - EditorMenu, 116
- sendData
 - ClientModel, 90, 91
 - NetworkModel, 206
 - ServerModel, 323
- SendInitialPlayerInformation, 306
 - execute, 307
 - SendInitialPlayerInformation, 307
 - SendInitialPlayerInformation, 307
- SendSpawnCharacterToServer, 307
 - execute, 308
 - newCharacter, 309
 - SendSpawnCharacterToServer, 308
 - SendSpawnCharacterToServer, 308
- SendToClientNextTurn, 309
 - execute, 310
 - SendToClientNextTurn, 310
 - SendToClientNextTurn, 310
- SendToServerEndTurn, 310
 - execute, 311
 - SendToServerEndTurn, 311
 - SendToServerEndTurn, 311
- SendUpdateCharacterToServer, 311
 - execute, 313
 - newCharacter, 313
 - oldCharacter, 313
 - SendUpdateCharacterToServer, 312
 - SendUpdateCharacterToServer, 312
- senderID
 - ReceiveSpawnCharacterFrom, 270
 - ReceiveUpdateCharacterFrom, 271
- Serialize, 313
 - deserialize, 314
 - fileInfo, 314
 - getFileInfo, 314
 - resetFileInfo, 314
 - Serialize, 314
 - serialize, 314
 - setFileInfo, 314
- serialize
 - BrickWall, 68
 - ClientGameModel, 86
 - EMailGameModel, 122
 - FileManager, 127
 - GameModel, 140
 - HotseatGameModel, 145
 - LocalGameModel, 161
 - MailGameInfo, 164
 - MapModel, 182
 - ObjectBase, 222
 - ObjectCharacter, 225
 - ObjectID, 229
 - ObjectItem, 232, 233
 - ObjectSpecial, 236
 - ObjectTile, 239
 - Player, 252
 - PlayerModel, 253
 - SaveGameFile, 300
 - Serialize, 314
 - ServerGameModel, 319
 - StackModel, 344
 - StartPosition, 349
 - WorldModel, 367
- server
 - ServerModel, 323
- ServerCommand, 315
 - execute, 316
 - isExecutable, 316
 - playerID, 316
 - ServerCommand, 316
 - ServerCommand, 316
- ServerGameModel, 317
 - beginTurn, 318
 - convertFromDifferentGameModel, 318
 - currentPlayerID, 320
 - deactivatePlayer, 318
 - deserialize, 318
 - endTurn, 318
 - firstTurn, 319
 - getCurrentPlayerID, 319
 - getGameType, 319
 - isYourTurn, 319
 - nextCurrentPlayer, 319
 - nextTurn, 319
 - refCurrentMap, 319
 - serialize, 319
 - ServerGameModel, 318
 - ServerGameModel, 318
 - setCurrentPlayerID, 319
 - spawnCharacter, 319
 - updateCharacter, 320
- ServerModel, 320
 - ~ServerModel, 321
 - changeClientID, 321
 - clients, 323
 - deleteDisconnectedClient, 322
 - getConnectedPlayerIDs, 322
 - getConnectionList, 322
 - isClient, 322
 - isServer, 322
 - listen, 322
 - newConnection, 322
 - processor, 323
 - receiveData, 322, 323
 - receivePendingData, 323
 - sendData, 323
 - server, 323
 - ServerModel, 321
 - ServerModel, 321
 - socketError, 323
 - worldModel, 323
- serverOverview

- CreateCharacter, 110
- serverOverviewLayout
 - CreateCharacter, 110
- setActivePlayer
 - PlayerModel, 254
- setCameFrom
 - AStarMapEntry, 62
- setCharacter
 - StackModel, 344
- setCharacterSpwawned
 - PlayerModel, 254
- setClipboardObject
 - MapModel, 182
- setConvertedPath
 - MapModel, 182
- setCurrentHeight
 - MapModel, 182
- setCurrentMap
 - ClientGameModel, 86
 - GameModel, 140
 - LocalGameModel, 161
- setCurrentPlayerID
 - ServerGameModel, 319
- setDescription
 - ObjectBase, 222
- setEditorModel
 - Processor, 260
- setErrorEnabled
 - BaseCommand, 65
- setF
 - AStarMapEntry, 62
- setFileInfo
 - Serialize, 314
- setFilePath
 - FileManager, 127
- setFloor
 - StackModel, 344
- setG
 - AStarMapEntry, 62
- setGameModel
 - Processor, 260
- setGridLineSize
 - MapModel, 182
- setHSquareCount
 - MapModel, 182
- setIsVisible
 - ObjectBase, 222
- setItemList
 - StackModel, 344
- setLocalEMailGamePath
 - MailGameInfo, 164
- setLocalMapPath
 - MapModel, 182
- setMainTool
 - EditorTools, 119
- setMainToolObject
 - EditorModel, 117
- setMapModel
 - Processor, 260
- setMaxAmountPerStack
 - ObjectSpecial, 236
- setMaxHeight
 - MapModel, 182
- setModels
 - BaseCommand, 65
 - Command, 95
- setNetworkModel
 - Processor, 260
- setNewMap
 - MainWindow, 171
- setNewMapSignal
 - GameModel, 140
- setNewStandardMap
 - MapModel, 184
- setOnlySavedMapChanges
 - UndoRedo, 358
- setOnlySavedWorldChanges
 - UndoRedo, 358
- setPathIndication
 - StackModel, 344
- setPlayerChoosesStartLocation
 - GameModel, 141
- setPlayerID
 - ClientSocket, 92
 - GameModel, 141
- setPlayerMapPath
 - PlayerModel, 254
- setPlayerModels
 - GameModel, 141
- setPosition
 - ObjectBase, 222
 - StackModel, 344
- setPossibleToReachOriginalState
 - UndoRedo, 358
- setPropertyListWidth
 - OptionsLayout, 241
- setRequestedData
 - GameModel, 141
- setRequestedObject
 - GameModel, 141
- setSaveDir
 - MailGameInfo, 164
- setSavedStates
 - UndoRedo, 358
- setSpawnID
 - ObjectBase, 222
- setSpecialList
 - StackModel, 344
- setSquareSizeZoomFactor
 - MapModel, 184
- setStackModel
 - MainViewStack, 168
- setStartMaps
 - WorldModel, 367
- setTarget
 - MouseState, 198

- setTileType
 - ObjectTile, 239
- setTurnCount
 - EmailGameModel, 122
- setUsedImageIndex
 - ObjectBase, 222
- setVSSquareCount
 - MapModel, 184
- setVisibleInGame
 - ObjectSpecial, 236
- setWall
 - StackModel, 344
- setWeight
 - AStarMapEntry, 62
- setWeightToInfinity
 - AStarMapEntry, 62
- setWorldFileInfo
 - WorldModel, 367
- setWorldModel
 - Processor, 260
- setYourTurn
 - ClientGameModel, 86
- SetupNewGame, 324
 - applyGeneralProperties, 325
 - emailSetupWidget, 326
 - execute, 325
 - gameModelPointer, 326
 - gameSetupDialog, 326
 - gameSetupLayout, 326
 - generalSetupWidget, 326
 - initEmailGame, 325
 - initGameSetupDialog, 325
 - initNetwork, 326
 - initSoloHotseat, 326
 - networkSetupWidget, 326
 - SetupNewGame, 325
 - SetupNewGame, 325
 - soloHotseatSetupWidget, 326
- ShowAbout, 326
 - execute, 327
 - ShowAbout, 327
 - ShowAbout, 327
- showContextMenu
 - ContextEditorMenuState, 102
 - ContextGameMenuState, 103
 - MainViewMap, 166
 - MouseInput, 195
 - MouseState, 198
- showCurrentWidget
 - OptionsLayout, 241
- ShowDescription, 328
 - execute, 328
 - ShowDescription, 328
 - ShowDescription, 328
- showDialog
 - YesNoCancelDialog, 372
- showEditorContextMenu
 - MouseEditorInput, 191
- showGameContextMenu
 - MouseGameInput, 194
- ShowGameOptions, 329
 - addHotseatPlayer, 330
 - addNewPlayerButton, 331
 - choseSaveDir, 330
 - execute, 330
 - gameOptionsDialog, 331
 - gameOptionsLayout, 331
 - init, 330
 - leaveGame, 330
 - leaveGameButton, 331
 - saveDirChooseButton, 331
 - saveDirLabel, 331
 - saveDirLineEdit, 331
 - saveGame, 330
 - saveGameButton, 331
 - ShowGameOptions, 330
 - ShowGameOptions, 330
 - specificGameTypeLayout, 331
 - specificGameTypeWidget, 331
- showGameOptions
 - MouseGameInput, 194
- showMapProperties
 - EditorMenu, 115
- showUsage
 - PIDGIRL, 40
- showWorldProperties
 - EditorMenu, 115
- Side
 - MapModel, 176
- socket
 - ClientSocket, 92
- socketError
 - ClientModel, 91
 - ServerModel, 323
- SoloHotseatSetupWidget, 331
 - gameStartButton, 332
 - SoloHotseatSetupWidget, 332
 - soloHotseatWidgetLayout, 332
 - SoloHotseatSetupWidget, 332
- soloHotseatSetupWidget
 - LoadGame, 150
 - SetupNewGame, 326
- soloHotseatWidgetLayout
 - SoloHotseatSetupWidget, 332
- spawnCharacter
 - ClientGameModel, 86
 - GameModel, 141
 - ServerGameModel, 319
- SpawnGameObject, 332
 - execute, 333
 - SpawnGameObject, 333
 - spawnableObject, 333
 - SpawnGameObject, 333
 - stackModel, 333
- spawnID
 - ObjectBase, 223

- SpawnPlayer, 334
- SpawnPlayerOnCurrentMap, 334
 - execute, 335
 - playerName, 335
 - SpawnPlayerOnCurrentMap, 335
 - SpawnPlayerOnCurrentMap, 335
- spawnableObject
 - SpawnGameObject, 333
- special
 - PlaceSpecial, 248
- SpecialID
 - ObjectID, 229
- specialIDExists
 - StackModel, 344
- specialList
 - StackModel, 347
- specialListChanged
 - StackModel, 345
- specialListSize
 - StackModel, 345
- specialSpawnIDMap
 - ObjectID, 229
- specialsMenu
 - MouseEditorInput, 192
- specificGameTypeLayout
 - ShowGameOptions, 331
- specificGameTypeWidget
 - ShowGameOptions, 331
- squarePos
 - StackModel, 347
- squareSize
 - MapModel, 184
- squareSizeChanged
 - MapModel, 184
- squareSizeZoomFactor
 - MapModel, 184
- stackGrid
 - MainViewMap, 166
- stackLayout
 - OptionsLayout, 241
- StackModel, 335
 - character, 347
 - characterChanged, 339
 - characterExists, 339
 - DESTINATION, 339
 - deleteAll, 339
 - deleteAllItems, 339
 - deleteAllSpecials, 339
 - deleteCharacter, 339
 - deleteFloor, 339
 - deleteItem, 339
 - deleteSpecial, 341
 - deleteWall, 341
 - deserialize, 341
 - E, 339
 - floor, 347
 - floorChanged, 341
 - getHeight, 341
 - getPathIndication, 341
 - getSquarePos, 341
 - hasCharacter, 341
 - hasFloor, 341
 - hasSpaceForSpecial, 341
 - hasWall, 341
 - height, 347
 - initStack, 342
 - isEmpty, 342
 - itemList, 347
 - itemListChanged, 342
 - itemListSize, 342
 - N, 338
 - NE, 339
 - NONE, 338
 - NW, 338
 - PathIndication, 338
 - pathIndication, 347
 - pathIndicationChanged, 342
 - playerRemoved, 342
 - playerSet, 342
 - prependItem, 342
 - prependSpecial, 342
 - refCharacter, 342
 - refFloor, 342
 - refItem, 342
 - refItemList, 342
 - refSpecial, 342
 - refSpecialList, 343
 - refWall, 343
 - removeAll, 343
 - removeAllItems, 343
 - removeAllSpecials, 343
 - removeCharacter, 343
 - removeFloor, 343
 - removeItem, 343
 - removeSpecial, 343
 - removeWall, 343
 - replaceCharacter, 343
 - replaceFloor, 344
 - replaceWall, 344
 - resetPathDirection, 344
 - S, 338
 - SE, 339
 - SW, 338
 - serialize, 344
 - setCharacter, 344
 - setFloor, 344
 - setItemList, 344
 - setPathIndication, 344
 - setPosition, 344
 - setSpecialList, 344
 - setWall, 344
 - specialIDExists, 344
 - specialList, 347
 - specialListChanged, 345
 - specialListSize, 345
 - squarePos, 347

- StackModel, 339
- StackModel, 339
- startPositionRemoved, 345
- startPositionSet, 345
- takeAllItems, 345
- takeAllSpecial, 345
- takeCharacter, 345
 - StackModel, 345
- takeFloor, 345
 - StackModel, 345
- takeItem, 345
 - StackModel, 345
- takeSpecial, 347
 - StackModel, 347
- takeWall, 347
 - StackModel, 347
- W, 338
- wall, 347
- wallChanged, 347
- stackModel
 - CutWall, 111
 - MainViewStack, 169
 - PasteObject, 243
 - PlaceEditorObject, 244
 - PlaceSpecial, 248
 - PlaceWall, 249
 - RemoveAllObjects, 273
 - RemoveWall, 280
 - SpawnGameObject, 333
- stackModelGrid
 - MapModel, 185
- stackedProperties
 - OptionsLayout, 241
- startMapIndex
 - LoadStartMapFromIndex, 155
- startMaps
 - WorldModel, 367
- startNetworkButton
 - ConnectTo, 101
- StartPosition, 348
 - copy, 349
 - createInformationWidget, 349
 - deserialize, 349
 - getCommandList, 349
 - getObjectID, 349
 - serialize, 349
 - StartPosition, 349
 - StartPosition, 349
 - update, 349
- startPositionRemoved
 - StackModel, 345
- startPositionSet
 - StackModel, 345
- startPositionStacks
 - MapModel, 185
- State, 33
- TILE
 - ObjectBase, 219
- TOOL_BUTTON_COUNT_IN_ROW
 - PIDGIRL, 44
- takeAllItems
 - StackModel, 345
- takeAllSpecial
 - StackModel, 345
- takeCharacter
 - StackModel, 345
- takeFloor
 - StackModel, 345
- takeItem
 - StackModel, 345
- takeSpecial
 - StackModel, 347
- takeWall
 - StackModel, 347
- target
 - MouseState, 198
- TargetChosenState, 350
 - nextState, 350
 - TargetChosenState, 350
 - TargetChosenState, 350
- TargetConfirmedState, 351
 - nextState, 352
 - TargetConfirmedState, 352
 - TargetConfirmedState, 352
- TargetMoveState, 352
 - nextState, 353
 - TargetMoveState, 353
 - TargetMoveState, 353
- TileID
 - ObjectID, 229
- tileSpawnIDMap
 - ObjectID, 229
- TileType
 - ObjectTile, 238
- tileType
 - ObjectTile, 240
- to
 - CopyFolder, 105
 - FindPath, 129
 - MoveCharacter, 199
- toIndex
 - MoveFile, 201
- toggleVisibility
 - ObjectBase, 222
- toolButtons
 - EditorTools, 120
- triggerCommandByName
 - ObjectMenu, 234
- triggerWithName
 - CommandAction, 97
- turnCount
 - EmailGameModel, 122
 - SaveEmailGame, 297
- undo
 - BaseCommand, 65
 - Command, 96
 - NotUndoable, 216
 - PlaceSpecial, 247
 - PlaceWall, 249
 - RemoveAllObjects, 273
 - RemoveWall, 280

- Undoable, 355
- UndoRedo, 358
- undoAction
 - EditorMenu, 116
- undoChanged
 - UndoRedo, 358
- undoCharacter
 - RemoveAllObjects, 273
- undoFloor
 - RemoveAllObjects, 273
- undoItemList
 - RemoveAllObjects, 273
- undoList
 - UndoRedo, 358
- undoListSavedState
 - UndoRedo, 358
- UndoRedo, 355
 - ~UndoRedo, 357
 - add, 357
 - clear, 357
 - clearRedoList, 357
 - clearUndoList, 357
 - hasOnlySavedMapChanges, 357
 - hasOnlySavedWorldChanges, 357
 - onlySavedMapChanges, 358
 - onlySavedWorldChanges, 358
 - possibleToReachOriginalState, 358
 - redo, 357
 - redoChanged, 357
 - redoList, 358
 - redoListSavedState, 358
 - setOnlySavedMapChanges, 358
 - setOnlySavedWorldChanges, 358
 - setPossibleToReachOriginalState, 358
 - setSavedStates, 358
 - undo, 358
 - undoChanged, 358
 - undoList, 358
 - undoListSavedState, 358
 - UndoRedo, 357
 - UndoRedo, 357
- undoRedo
 - Command, 96
 - Processor, 260
- undoSpecialList
 - RemoveAllObjects, 273
- undoWall
 - RemoveAllObjects, 274
- Undoable, 353
 - execute, 354
 - isClearingUndoRedo, 354
 - isExecutable, 355
 - isUndoable, 355
 - redo, 355
 - undo, 355
 - Undoable, 354
- unsetPosition
 - ObjectBase, 222
- update
 - BrickWall, 68
 - ObjectBase, 222
 - ObjectCharacter, 226
 - ObjectItem, 233
 - ObjectSpecial, 236
 - ObjectTile, 239
 - Player, 252
 - StartPosition, 349
- updateAll
 - MainViewStack, 168
- updateAvailableFileSystemModel
 - WorldPropertiesDialog, 369
- updateBase
 - ObjectBase, 222
- updateCharacter
 - ClientGameModel, 87
 - GameModel, 141
 - ObjectCharacter, 226
 - ServerGameModel, 320
- updateChosenMaps
 - WorldPropertiesDialog, 369
- updateColumnWidth
 - WorldFileTree, 365
- updateEditorSize
 - MainWindow, 172
- updateHorizontal
 - MapPropertiesDialog, 186
- updateItem
 - ObjectItem, 233
- updateMap
 - MainViewMap, 166
- updateMaxHeight
 - MapPropertiesDialog, 186
- updateRootIndex
 - WorldFileTree, 365
- updateSize
 - EditorMenu, 115
 - MainViewStack, 169
 - MainWindow, 172
- updateSpecial
 - ObjectSpecial, 236
- updateSquareSize
 - MapModel, 184
- updateTile
 - ObjectTile, 239
- updateVertical
 - MapPropertiesDialog, 186
- updateWindowTitle
 - MainWindow, 172
- UseRedo, 359
 - execute, 359
 - UseRedo, 359
 - UseRedo, 359
- useRedo
 - EditorMenu, 115
- UseUndo, 360
 - execute, 361

- UseUndo, 361
- UseUndo, 361
- useUndo
 - EditorMenu, 115
- usedImageIndex
 - ObjectBase, 223
- verticalDirection
 - ApplyMapGeometryChanges, 51
- verticalNewValue
 - ApplyMapGeometryChanges, 51
- verticalSizeChanged
 - MapModel, 184
- verticalSquareSideComboBox
 - MapPropertiesDialog, 186
- verticalSquareSpinBox
 - MapPropertiesDialog, 186
- viewMenu
 - EditorMenu, 116
- visibleInGame
 - ObjectSpecial, 237
- W
 - StackModel, 338
- WALL
 - ObjectTile, 238
- WORLD_FOLDER_NAME
 - PIDGIRL, 44
- WORLD_SUFFIX
 - PIDGIRL, 44
- waitFor
 - WaitingForDataDialog, 362
- WaitingForDataDialog, 361
 - dataReceived, 362
 - fromClient, 362
 - fromServer, 362
 - hasBeenReceived, 362
 - messageBox, 363
 - networkModel, 363
 - waitFor, 362
 - WaitingForDataDialog, 362
 - WaitingForDataDialog, 362
- wall
 - CopyWall, 107
 - PlaceWall, 249
 - StackModel, 347
- wallChanged
 - StackModel, 347
- wallMenu
 - MouseEditorInput, 193
- weight
 - AStarMapEntry, 62
- windowSizeChanged
 - MainWindow, 172
- worldFileInfo
 - WorldModel, 368
- WorldFileTree, 363
 - contextMenu, 365
 - dragMoveEvent, 364
 - draggedIndex, 365
 - dropEvent, 364
 - duplicate, 364
 - duplicateAction, 365
 - indexDoubleClicked, 364
 - mousePressEvent, 364
 - mouseReleaseEvent, 365
 - newFolder, 365
 - newFolderAction, 365
 - newMap, 365
 - newMapAction, 365
 - processor, 365
 - refContextMenu, 365
 - remove, 365
 - removeAction, 365
 - rename, 365
 - renameAction, 365
 - updateColumnWidth, 365
 - updateRootIndex, 365
 - WorldFileTree, 364
 - worldModel, 365
 - WorldFileTree, 364
- WorldModel, 366
 - absoluteMapPathToLocalMapPath, 367
 - deserialize, 367
 - fileChanged, 367
 - getStartMapWorldPaths, 367
 - localMapPathToAbsolutePath, 367
 - objectID, 367
 - refObjectID, 367
 - refWorldFileInfo, 367
 - serialize, 367
 - setStartMaps, 367
 - setWorldFileInfo, 367
 - startMaps, 367
 - worldFileInfo, 368
 - WorldModel, 367
 - WorldModel, 367
- worldModel
 - Command, 96
 - EditorMenu, 116
 - MainWindow, 172
 - Processor, 260
 - SaveGameFile, 300
 - ServerModel, 323
 - WorldFileTree, 365
 - WorldPropertiesDialog, 370
 - WorldView, 371
- worldPropertiesAction
 - EditorMenu, 116
- WorldPropertiesDialog, 368
 - addButton, 369
 - addPathToChosen, 369
 - applyChanges, 369
 - availableFileSystemModel, 369
 - availableTreeUpdateColumnWidth, 369
 - availableTreeView, 369
 - chosenListModel, 369

- chosenListView, 369
- exec, 369
- manageStartingMaps, 369
- manageStartingMapsLayout, 369
- optionsLayout, 370
- processor, 370
- removeButton, 370
- removePathFromChosen, 369
- updateAvailableFileSystemModel, 369
- updateChosenMaps, 369
- worldModel, 370
- WorldPropertiesDialog, 369
- WorldPropertiesDialog, 369
- worldPropertiesDialog
 - EditorMenu, 116
- WorldView, 370
 - fileTree, 371
 - processor, 371
 - resizeWorldView, 371
 - worldModel, 371
 - WorldView, 371
 - WorldView, 371
- worldView
 - MainWindow, 172
- worldViewDock
 - MainWindow, 172
- writeLog
 - PIDGIRL, 40
- writeLogDone
 - PIDGIRL, 41
- writeLogFail
 - PIDGIRL, 41
- x
 - ObjectBase, 223
- y
 - ObjectBase, 223
- YES_BUTTON
 - YesNoCancelDialog, 372
- yesButton
 - YesNoCancelDialog, 372
- YesNoCancelDialog, 371
 - ButtonClicked, 372
 - buttonClicked, 372
 - CANCEL_BUTTON, 372
 - getButtonClicked, 372
 - NO_BUTTON, 372
 - noButton, 372
 - showDialog, 372
 - YES_BUTTON, 372
 - yesButton, 372
 - YesNoCancelDialog, 372
 - YesNoCancelDialog, 372
- yncDialog
 - AskForSaveMap, 55
 - AskForSaveWorld, 56
- yourTurn
 - ClientGameModel, 87
- YourTurnInformation, 373
 - execute, 374
 - YourTurnInformation, 374
 - YourTurnInformation, 374
- z
 - ObjectBase, 223
- zoom
 - EditorMenu, 115
- zoomActionGroup
 - EditorMenu, 116
- zoomMenu
 - EditorMenu, 116