

mkRPG

Generated by Doxygen 1.8.12



# Contents

<b>1</b>	<b><a href="#">L'éditeur</a></b>	<b>1</b>
<b>2</b>	<b><a href="#">readme</a></b>	<b>3</b>
<b>3</b>	<b><a href="#">Todo List</a></b>	<b>5</b>
<b>4</b>	<b><a href="#">Deprecated List</a></b>	<b>7</b>
<b>5</b>	<b><a href="#">Namespace Index</a></b>	<b>9</b>
5.1	<a href="#">Namespace List</a> . . . . .	9
<b>6</b>	<b><a href="#">Hierarchical Index</a></b>	<b>11</b>
6.1	<a href="#">Class Hierarchy</a> . . . . .	11
<b>7</b>	<b><a href="#">Class Index</a></b>	<b>15</b>
7.1	<a href="#">Class List</a> . . . . .	15
<b>8</b>	<b><a href="#">File Index</a></b>	<b>19</b>
8.1	<a href="#">File List</a> . . . . .	19
<b>9</b>	<b><a href="#">Namespace Documentation</a></b>	<b>21</b>
9.1	<a href="#">src.parsing.map_parser Namespace Reference</a> . . . . .	21
9.1.1	<a href="#">Detailed Description</a> . . . . .	21
9.1.2	<a href="#">Function Documentation</a> . . . . .	21
9.1.2.1	<a href="#">get_size()</a> . . . . .	21
9.1.2.2	<a href="#">map_parser()</a> . . . . .	21
9.1.2.3	<a href="#">parse_cell()</a> . . . . .	21

<b>10 Class Documentation</b>	<b>23</b>
10.1 src.path.Archi Class Reference	23
10.1.1 Detailed Description	23
10.1.2 Member Function Documentation	23
10.1.2.1 get_src_dir()	23
10.1.2.2 get_src_file()	24
10.1.2.3 get_static_dir()	24
10.1.2.4 get_static_file()	24
10.1.2.5 get_xml_dir()	24
10.1.2.6 get_xml_file()	24
10.1.2.7 list_files()	25
10.2 src.backgroundLayer.BackgroundLayer Class Reference	25
10.3 src.world.BaseObject Class Reference	26
10.3.1 Detailed Description	26
10.3.2 Member Function Documentation	26
10.3.2.1 contextEval()	26
10.3.2.2 load()	26
10.4 BColor Class Reference	27
10.4.1 Detailed Description	27
10.4.2 Constructor & Destructor Documentation	27
10.4.2.1 BColor() [1/2]	27
10.4.2.2 BColor() [2/2]	28
10.4.3 Member Function Documentation	28
10.4.3.1 color()	28
10.4.3.2 colorChanged	28
10.4.3.3 name()	28
10.4.3.4 nameChanged	28
10.4.3.5 setColor	29
10.4.3.6 setColorQuiet	29
10.4.3.7 setName	29

10.4.3.8 setNameQuiet . . . . .	29
10.4.4 Property Documentation . . . . .	30
10.4.4.1 color . . . . .	30
10.4.4.2 name . . . . .	30
10.5 BDock Class Reference . . . . .	30
10.5.1 Detailed Description . . . . .	31
10.6 BDocksZone Class Reference . . . . .	31
10.7 BDockWidget Class Reference . . . . .	32
10.7.1 Detailed Description . . . . .	33
10.8 BinaryStateMachine Class Reference . . . . .	33
10.8.1 Detailed Description . . . . .	34
10.9 BLayout Class Reference . . . . .	34
10.10src.world.Cell Class Reference . . . . .	35
10.11Cell Class Reference . . . . .	35
10.11.1 Detailed Description . . . . .	36
10.12CellDock Class Reference . . . . .	36
10.13CellType Class Reference . . . . .	37
10.13.1 Detailed Description . . . . .	37
10.14CellTypeListModel Class Reference . . . . .	37
10.14.1 Detailed Description . . . . .	38
10.15CellTypesDock Class Reference . . . . .	38
10.16src.cursescli.CellViewer Class Reference . . . . .	38
10.17src.character.Character Class Reference . . . . .	39
10.18src.chunk.Chunk Class Reference . . . . .	39
10.19src.cache.ChunkCache Class Reference . . . . .	40
10.20CICoords Class Reference . . . . .	41
10.20.1 Detailed Description . . . . .	41
10.21src.pygamecli.Client Class Reference . . . . .	41
10.22src.client.Client Class Reference . . . . .	42
10.22.1 Detailed Description . . . . .	42

10.22.2 Constructor & Destructor Documentation . . . . .	43
10.22.2.1 __del__() . . . . .	43
10.23src.cursescli.Curses Class Reference . . . . .	43
10.23.1 Detailed Description . . . . .	43
10.24Editor Class Reference . . . . .	44
10.24.1 Detailed Description . . . . .	44
10.25src.world.Entity Class Reference . . . . .	44
10.26Game Class Reference . . . . .	45
10.26.1 Detailed Description . . . . .	45
10.26.2 Member Function Documentation . . . . .	46
10.26.2.1 newIdent() . . . . .	46
10.27GameObject Class Reference . . . . .	46
10.27.1 Detailed Description . . . . .	47
10.27.2 Constructor & Destructor Documentation . . . . .	48
10.27.2.1 GameObject() . . . . .	48
10.27.3 Member Function Documentation . . . . .	48
10.27.3.1 flags() . . . . .	48
10.27.3.2 getFlag() . . . . .	48
10.27.3.3 getParam() . . . . .	49
10.27.3.4 hasFlag() . . . . .	49
10.27.3.5 hasParam() . . . . .	49
10.27.3.6 ident() . . . . .	49
10.27.3.7 init() . . . . .	50
10.27.3.8 isValid() . . . . .	50
10.27.3.9 lastChildrenEdition() . . . . .	50
10.27.3.10lastEdition() . . . . .	50
10.27.3.11lastInternalEdition() . . . . .	51
10.27.3.12params() . . . . .	51
10.27.3.13setFlag() . . . . .	51
10.27.3.14setParam() . . . . .	51

10.27.3.15 touch() . . . . .	52
10.28 src.cache.GlobalCache Class Reference . . . . .	52
10.29 Image Class Reference . . . . .	53
10.29.1 Detailed Description . . . . .	53
10.29.2 Member Function Documentation . . . . .	53
10.29.2.1 isValid() . . . . .	53
10.30 src.cache.ImageCache Class Reference . . . . .	54
10.31 src.interactions.Interaction Class Reference . . . . .	54
10.31.1 Detailed Description . . . . .	55
10.32 src.printWorld.Interface Class Reference . . . . .	55
10.33 src.interface.Interface Class Reference . . . . .	55
10.33.1 Detailed Description . . . . .	56
10.34 Intertie Class Reference . . . . .	56
10.34.1 Detailed Description . . . . .	56
10.35 src.layer.Layer Class Reference . . . . .	57
10.36 Map Class Reference . . . . .	57
10.36.1 Detailed Description . . . . .	58
10.37 src.world.Map Class Reference . . . . .	58
10.37.1 Member Function Documentation . . . . .	59
10.37.1.1 fill() . . . . .	59
10.38 MapDock Class Reference . . . . .	59
10.39 MapPainter Class Reference . . . . .	59
10.39.1 Detailed Description . . . . .	61
10.39.2 Member Enumeration Documentation . . . . .	61
10.39.2.1 Element . . . . .	61
10.39.3 Constructor & Destructor Documentation . . . . .	61
10.39.3.1 MapPainter() [1/2] . . . . .	61
10.39.3.2 MapPainter() [2/2] . . . . .	62
10.39.4 Member Function Documentation . . . . .	62
10.39.4.1 cooToPt() . . . . .	62

10.39.4.2	cooToPxl()	62
10.39.4.3	hasHighlightedCell()	62
10.39.4.4	highlightedCell()	62
10.39.4.5	indToPt()	63
10.39.4.6	isCell()	63
10.39.4.7	mapSizeChanged	63
10.39.4.8	move()	63
10.39.4.9	paint()	63
10.39.4.10	preSelectedCellColor()	64
10.39.4.11	ptToCoo()	64
10.39.4.12	ptToPxl()	64
10.39.4.13	pxlToCoo()	64
10.39.4.14	pxlToPt()	64
10.39.4.15	render()	64
10.39.4.16	resize() [1/2]	65
10.39.4.17	resize() [2/2]	65
10.39.4.18	scale()	65
10.39.4.19	selectedCellColor()	65
10.39.4.20	setHighlightedCell() [1/2]	65
10.39.4.21	setHighlightedCell() [2/2]	66
10.39.4.22	setMap()	66
10.39.4.23	setPaintedElement()	66
10.39.4.24	setPaintedElements()	66
10.39.4.25	setPreSelectedCellColor()	66
10.39.4.26	setScale()	67
10.39.4.27	setScaleDomain()	67
10.39.4.28	setSelectedCellColor()	67
10.39.4.29	setViewCenter() [1/2]	67
10.39.4.30	setViewCenter() [2/2]	68
10.39.4.31	setViewCenterQuiet()	68



10.39.4.32size()	68
10.39.4.33viewCenter()	68
10.39.4.34viewCenterChanged	68
10.39.4.35virtualSize()	69
10.39.4.36zoom()	69
10.40MapsEditor Class Reference	69
10.40.1 Detailed Description	70
10.41MapsListModel Class Reference	70
10.41.1 Detailed Description	70
10.42src.cursescli.MapViewer Class Reference	71
10.43src.map.MapViewer Class Reference	71
10.44MapView Class Reference	72
10.44.1 Detailed Description	73
10.44.2 Member Enumeration Documentation	73
10.44.2.1 SelectionMode	73
10.45src.network.NetworkClient Class Reference	74
10.45.1 Detailed Description	74
10.45.2 Member Function Documentation	74
10.45.2.1 askEntity()	74
10.45.2.2 run()	74
10.45.2.3 sendEvent()	75
10.46src.networkudp.NetworkClient Class Reference	75
10.47src.network.NetworkServer Class Reference	76
10.47.1 Detailed Description	76
10.47.2 Constructor & Destructor Documentation	76
10.47.2.1 __init__()	76
10.47.3 Member Function Documentation	76
10.47.3.1 run()	76
10.47.3.2 sendOrder()	77
10.47.3.3 waitForClients()	77

10.48src.networkudp.NetworkServer Class Reference . . . . .	77
10.49NewGame Class Reference . . . . .	78
10.50Object Class Reference . . . . .	78
10.50.1 Detailed Description . . . . .	78
10.51ObjectEditor Class Reference . . . . .	79
10.52ObjectFlagTableModel Class Reference . . . . .	79
10.53ObjectParamTableModel Class Reference . . . . .	80
10.54src.world.ObjectType Class Reference . . . . .	80
10.54.1 Detailed Description . . . . .	81
10.54.2 Member Function Documentation . . . . .	81
10.54.2.1 create() . . . . .	81
10.55Options Struct Reference . . . . .	81
10.55.1 Detailed Description . . . . .	81
10.55.2 Member Function Documentation . . . . .	83
10.55.2.1 isAdjustable() . . . . .	83
10.55.2.2 load() . . . . .	83
10.55.2.3 options() . . . . .	83
10.55.2.4 reinitialise() . . . . .	83
10.55.2.5 save() . . . . .	84
10.55.2.6 setAdjustable() . . . . .	84
10.55.2.7 setDefault() . . . . .	84
10.56src.orders.Order Class Reference . . . . .	85
10.56.1 Detailed Description . . . . .	85
10.56.2 Member Function Documentation . . . . .	85
10.56.2.1 fromBytes() . . . . .	85
10.56.2.2 load() . . . . .	85
10.56.2.3 toBytes() . . . . .	86
10.57src.orders.OrderDispatcher Class Reference . . . . .	86
10.57.1 Detailed Description . . . . .	86
10.57.2 Member Function Documentation . . . . .	86

10.57.2.1 treat()	86
10.58ParamItemDelegate Class Reference	87
10.59src.tools.Perf Class Reference	87
10.59.1 Detailed Description	87
10.59.2 Member Function Documentation	88
10.59.2.1 show()	88
10.59.2.2 tic()	88
10.59.2.3 toc()	88
10.60PtCoords Class Reference	88
10.60.1 Detailed Description	89
10.61PxCoords Class Reference	89
10.61.1 Detailed Description	89
10.62RlCoords Class Reference	90
10.62.1 Detailed Description	90
10.63src.cache.ScaledCache Class Reference	90
10.64SelectionDock Class Reference	91
10.65src.server.Server Class Reference	91
10.65.1 Detailed Description	92
10.66src.network.ServerConnection Class Reference	92
10.66.1 Detailed Description	92
10.66.2 Member Function Documentation	93
10.66.2.1 run()	93
10.67TabAcces Class Reference	93
10.68TabBar Class Reference	93
10.69src.tools.Timer Class Reference	94
10.69.1 Detailed Description	94
10.69.2 Member Function Documentation	95
10.69.2.1 add()	95
10.70src.utils.WalkableGraph Class Reference	95
10.71Welcome Class Reference	95

10.72src.world.World Class Reference . . . . .	96
10.73World Class Reference . . . . .	96
10.73.1 Detailed Description . . . . .	97
10.74WorldEditor Class Reference . . . . .	97
10.75src.printWorld.WorldViewer Class Reference . . . . .	98
10.75.1 Constructor & Destructor Documentation . . . . .	98
10.75.1.1 __init__() . . . . .	98
10.75.2 Member Function Documentation . . . . .	98
10.75.2.1 end() . . . . .	98
10.75.2.2 get_event() . . . . .	99
10.75.2.3 move() . . . . .	99
10.75.2.4 move_char() . . . . .	99
10.75.2.5 propagate_trigger() . . . . .	99
10.75.2.6 render() . . . . .	99
10.75.2.7 update() . . . . .	100
10.76XmlHandler Class Reference . . . . .	100
<b>11 File Documentation</b>	<b>101</b>
11.1 src/editor/Game/game.h File Reference . . . . .	101
11.1.1 Detailed Description . . . . .	101
11.2 src/editor/Game/map.h File Reference . . . . .	101
11.2.1 Detailed Description . . . . .	102
11.2.2 Macro Definition Documentation . . . . .	102
11.2.2.1 forCells . . . . .	102
11.3 src/editor/Game/mappainter.h File Reference . . . . .	102
11.3.1 Detailed Description . . . . .	103
11.3.2 Function Documentation . . . . .	103
11.3.2.1 operator^() . . . . .	103
11.4 src/editor/Game/mapslistmodel.h File Reference . . . . .	103
11.4.1 Detailed Description . . . . .	104
11.5 src/editor/Game/object.h File Reference . . . . .	104

11.5.1 Detailed Description . . . . .	105
11.5.2 Macro Definition Documentation . . . . .	106
11.5.2.1 Attr . . . . .	106
11.5.2.2 AttrDef . . . . .	107
11.5.2.3 AttrGetter . . . . .	107
11.5.2.4 AttrSetter . . . . .	108
11.5.2.5 AttrT . . . . .	108
11.5.2.6 C . . . . .	109
11.5.2.7 C0 . . . . .	109
11.5.2.8 C1 . . . . .	110
11.5.2.9 Flag . . . . .	110
11.5.2.10 FlagGetter . . . . .	110
11.5.2.11 FlagSetter . . . . .	111
11.5.2.12 ObjectsMap . . . . .	111
11.5.2.13 ObjectsMapC . . . . .	111
11.5.2.14 Param . . . . .	112
11.5.2.15 ParamGetter . . . . .	112
11.5.2.16 ParamSetter . . . . .	113
11.5.2.17 SetFlag . . . . .	113
11.5.2.18 SetParam . . . . .	114
11.6 src/editor/Game/xmlhandler.h File Reference . . . . .	114
11.6.1 Detailed Description . . . . .	115
11.7 src/editor/GUI/options.h File Reference . . . . .	115
11.7.1 Detailed Description . . . . .	115
11.7.2 Macro Definition Documentation . . . . .	116
11.7.2.1 Default . . . . .	116
11.7.2.2 DefaultF . . . . .	116
11.7.2.3 DIR . . . . .	116
11.7.2.4 MAP . . . . .	116
11.7.2.5 WIN . . . . .	116
11.8 src/editor/GUI/Tabs/bcolor.h File Reference . . . . .	116
11.8.1 Detailed Description . . . . .	117
11.9 src/editor/GUI/Tabs/Docks/bdock.h File Reference . . . . .	117
11.9.1 Detailed Description . . . . .	117



# Chapter 1

## L'éditeur

### Compilation

Pour compiler l'éditeur, il est nécessaire de disposer de Qt 5.6.1 (ou version plus récentes). Le plus facile est d'ouvrir le projet avec QtCreator (le fichier .pro) et de demander gentiment une compilation (Ctrl + R).

### Test

Pas besoin d'ouvrir ou de créer de jeu pour pouvoir voir les quelques éléments implémentés : un "jeu" est (temporairement) prédéfini dans le code. Seules quelques fonctionnalités d'affichage existent, ainsi que la modification des angles. Pour l'instant...

### Contrôles

Dans l'onglet Maps (le seul qui est vraiment intéressant pour le moment) les commandes sont :

Zoom : Roulette de la souris

Déplacement : Clic droit + et déplacement

Sélection : Clic gauche ou Alt + déplacement





## Chapter 2

### readme

run `py client.py` to try.

Commands :

- Move map around by putting the mouse cursor on the edges of the screen
- Mouse wheel allows you to zoom in or out



## Chapter 3

## Todo List

Class `GameObject`



## Chapter 4

# Deprecated List

Member [ObjectsMap](#) (pref, ini, Ini, body, sg, pl)

Member [ObjectsMapC](#) (name, names, Type, Types, pref, arg)



## Chapter 5

# Namespace Index

### 5.1 Namespace List

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

<a href="#">src.parsing.map_parser</a> . . . . .	21
--	----





## Chapter 6

# Hierarchical Index

### 6.1 Class Hierarchy

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

src.path.Archi . . . . .	23
src.world.BaseObject . . . . .	26
CellDock	
CellDock . . . . .	36
CellTypesDock	
CellTypesDock . . . . .	38
src.cursescli.CellViewer . . . . .	38
src.character.Character . . . . .	39
src.pygamecli.Client . . . . .	41
src.client.Client . . . . .	42
Editor	
Editor . . . . .	44
GameObject . . . . .	46
Cell . . . . .	35
CellType . . . . .	37
Game . . . . .	45
Image . . . . .	53
Map . . . . .	57
Object . . . . .	78
World . . . . .	96
src.cache.GlobalCache . . . . .	52
src.cache.ScaledCache . . . . .	90
src.cache.ChunkCache . . . . .	40
src.cache.ImageCache . . . . .	54
Group	
src.map.MapViewer . . . . .	71
src.interactions.Interaction . . . . .	54
src.printWorld.Interface . . . . .	55
src.printWorld.WorldViewer . . . . .	98
src.interface.Interface . . . . .	55
src.cursescli.Curses . . . . .	43
MagicObject	
src.world.Cell . . . . .	35
src.world.Entity . . . . .	44

src.world.Map	58
src.world.ObjectType	80
src.world.World	96
MapDock	
MapDock	59
MapsEditor	
MapsEditor	69
src.cursescli.MapViewer	71
src.network.NetworkClient	74
src.network.NetworkServer	76
NewGame	
NewGame	78
ObjectEditor	
ObjectEditor	79
Options	81
src.orders.Order	85
src.orders.OrderDispatcher	86
src.tools.Perf	87
QAbstractListModel	
CellTypeListModel	37
MapsListModel	70
QAbstractTableModel	
ObjectFlagTableModel	79
ObjectParamTableModel	80
QDialog	
NewGame	78
QFrame	
BColor	27
BDock	30
QMainWindow	
Editor	44
QObject	
Intertie	56
MapPainter	59
QPointF	
CICoords	41
PtCoords	88
PxCoords	89
RICoords	90
QStateMachine	
BinaryStateMachine	33
QStyledItemDelegate	
ParamItemDelegate	87
QWidget	
BDocksZone	31
BDockWidget	32
CellDock	36
CellTypesDock	38
MapDock	59
SelectionDock	91
BLayout	34
MapsEditor	69
MapViewer	72
ObjectEditor	79
TabAcces	93
TabBar	93
Welcome	95
WorldEditor	97

QXmlDefaultHandler	
XmlHandler . . . . .	100
SelectionDock	
SelectionDock . . . . .	91
src.server.Server . . . . .	91
src.network.ServerConnection . . . . .	92
Sprite	
src.chunk.Chunk . . . . .	39
src.layer.Layer . . . . .	57
src.backgroundLayer.BackgroundLayer . . . . .	25
TabAcces	
TabAcces . . . . .	93
src.tools.Timer . . . . .	94
src.utils.WalkableGraph . . . . .	95
Welcome	
Welcome . . . . .	95
WorldEditor	
WorldEditor . . . . .	97
Thread	
src.networkudp.NetworkClient . . . . .	75
src.networkudp.NetworkServer . . . . .	77



## Chapter 7

# Class Index

### 7.1 Class List

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

<a href="#">src.path.Archi</a>	23
<a href="#">src.backgroundLayer.BackgroundLayer</a>	25
<a href="#">src.world.BaseObject</a>	26
<a href="#">BColor</a>	
Simple frame that offers color selection	27
<a href="#">BDock</a>	
Container for widget to display in a <a href="#">BDocksZone</a>	30
<a href="#">BDocksZone</a>	31
<a href="#">BDockWidget</a>	
Base class for game-related docks	32
<a href="#">BinaryStateMachine</a>	
Simple QStateMachine with two states	33
<a href="#">BLayout</a>	34
<a href="#">src.world.Cell</a>	35
<a href="#">Cell</a>	
The <a href="#">Cell</a> class	35
<a href="#">CellDock</a>	36
<a href="#">CellType</a>	
The <a href="#">CellType</a> class	37
<a href="#">CellTypeListModel</a>	
The <a href="#">CellTypeListModel</a> class	37
<a href="#">CellTypesDock</a>	38
<a href="#">src.cursescli.CellViewer</a>	38
<a href="#">src.character.Character</a>	39
<a href="#">src.chunk.Chunk</a>	39
<a href="#">src.cache.ChunkCache</a>	40
<a href="#">CICoords</a>	
Describe positions with cell coordinates	41
<a href="#">src.pygamecli.Client</a>	41
<a href="#">src.client.Client</a>	42
<a href="#">src.cursescli.Curses</a>	43
<a href="#">Editor</a>	
Main window of the <a href="#">Editor</a>	44
<a href="#">src.world.Entity</a>	44
<a href="#">Game</a>	
Gather the different parts needed to describe a game	45

<a href="#">GameObject</a>	
Base class for every part of games . . . . .	46
<a href="#">src.cache.GlobalCache</a> . . . . .	52
<a href="#">Image</a>	
Stores an external file in a QImage, and gives each image ressources a unique identifier . . . . .	53
<a href="#">src.cache.ImageCache</a> . . . . .	54
<a href="#">src.interactions.Interaction</a> . . . . .	54
<a href="#">src.printWorld.Interface</a> . . . . .	55
<a href="#">src.interface.Interface</a> . . . . .	55
<a href="#">Intertie</a>	
Provide int that move smoothly from their value to an objective . . . . .	56
<a href="#">src.layer.Layer</a> . . . . .	57
<a href="#">Map</a>	
The <a href="#">Map</a> class . . . . .	57
<a href="#">src.world.Map</a> . . . . .	58
<a href="#">MapDock</a> . . . . .	59
<a href="#">MapPainter</a>	
That can paint a <a href="#">Map</a> using a QPainter . . . . .	59
<a href="#">MapsEditor</a>	
Tab offering map editing facilities . . . . .	69
<a href="#">MapsListModel</a>	
Presentation class for the Qt Model-View framework . . . . .	70
<a href="#">src.cursescli.MapViewer</a> . . . . .	71
<a href="#">src.map.MapViewer</a> . . . . .	71
<a href="#">MapViewer</a>	
Widget to display and edit a <a href="#">Map</a> using a <a href="#">MapPainter</a> . . . . .	72
<a href="#">src.network.NetworkClient</a> . . . . .	74
<a href="#">src.networkudp.NetworkClient</a> . . . . .	75
<a href="#">src.network.NetworkServer</a> . . . . .	76
<a href="#">src.networkudp.NetworkServer</a> . . . . .	77
<a href="#">NewGame</a> . . . . .	78
<a href="#">Object</a>	
The <a href="#">Object</a> class . . . . .	78
<a href="#">ObjectEditor</a> . . . . .	79
<a href="#">ObjectFlagTableModel</a> . . . . .	79
<a href="#">ObjectParamTableModel</a> . . . . .	80
<a href="#">src.world.ObjectType</a> . . . . .	80
<a href="#">Options</a>	
Session-independant options and preferences . . . . .	81
<a href="#">src.orders.Order</a> . . . . .	85
<a href="#">src.orders.OrderDispatcher</a> . . . . .	86
<a href="#">ParamItemDelegate</a> . . . . .	87
<a href="#">src.tools.Perf</a> . . . . .	87
<a href="#">PtCoords</a>	
Describe positions with virtual point coordinates . . . . .	88
<a href="#">PxCoords</a>	
Describe positions with real pixel coordinates . . . . .	89
<a href="#">RlCoords</a>	
Describe positions with relative coordinates . . . . .	90
<a href="#">src.cache.ScaledCache</a> . . . . .	90
<a href="#">SelectionDock</a> . . . . .	91
<a href="#">src.server.Server</a> . . . . .	91
<a href="#">src.network.ServerConnection</a> . . . . .	92
<a href="#">TabAcces</a> . . . . .	93
<a href="#">TabBar</a> . . . . .	93
<a href="#">src.tools.Timer</a> . . . . .	94
<a href="#">src.utils.WalkableGraph</a> . . . . .	95
<a href="#">Welcome</a> . . . . .	95

<a href="#">src.world.World</a> . . . . .	96
<a href="#">World</a>	
The <a href="#">World</a> class . . . . .	96
<a href="#">WorldEditor</a> . . . . .	97
<a href="#">src.printWorld.WorldViewer</a> . . . . .	98
<a href="#">XmlHandler</a> . . . . .	100





## Chapter 8

# File Index

### 8.1 File List

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

src/editor/Game/game.h	
Definition of the <a href="#">Game</a> and <a href="#">World</a> classes	101
src/editor/Game/map.h	
Definition of the <a href="#">Map</a> , <a href="#">Cell</a> and <a href="#">CellType</a> classes	101
src/editor/Game/mappainter.h	
Definition of the <a href="#">MapPainter</a> class and other related classes to render <a href="#">maps</a>	102
src/editor/Game/mapslistmodel.h	
Definition of Model/View presentation classes	103
src/editor/Game/object.h	
Definition of the base class <a href="#">GameObject</a> , and some inherited classes	104
src/editor/Game/xmlhandler.h	
Definition of the <a href="#">XmlHandler</a> class and other related classes to read XML game's files	114
src/editor/GUI/editor.h	??
src/editor/GUI/newgame.h	??
src/editor/GUI/options.h	
Definition of the <a href="#">Options</a> class, and the constants that are used in this class	115
src/editor/GUI/tabaccess.h	??
src/editor/GUI/tabbar.h	??
src/editor/GUI/Tabs/bcolor.h	
Definition of the <a href="#">BColor</a> class	116
src/editor/GUI/Tabs/celldock.h	??
src/editor/GUI/Tabs/celltypesdock.h	??
src/editor/GUI/Tabs/mapdock.h	??
src/editor/GUI/Tabs/mapseeditor.h	??
src/editor/GUI/Tabs/mapviewer.h	??
src/editor/GUI/Tabs/objecteditor.h	??
src/editor/GUI/Tabs/selectiondock.h	??
src/editor/GUI/Tabs/welcome.h	??
src/editor/GUI/Tabs/worldeditor.h	??
src/editor/GUI/Tabs/Docks/bdock.h	
Definition of the <a href="#">BDock</a> class	117
src/editor/GUI/Tabs/Docks/bdockszone.h	??
src/editor/GUI/Tabs/Docks/bdockwidget.h	??
src/editor/GUI/Tabs/Docks/intertie.h	??



## Chapter 9

# Namespace Documentation

### 9.1 src.parsing.map\_parser Namespace Reference

#### Functions

- def [parse\\_cell](#) (cell\_object)
- def [map\\_parser](#) (map\_xml)
- def [get\\_size](#) (tree)
- def **gen\_map** (cells\_specs)

#### 9.1.1 Detailed Description

This module handles xml parsing for maps description files.

#### 9.1.2 Function Documentation

##### 9.1.2.1 [get\\_size\(\)](#)

```
def src.parsing.map_parser.get_size (  
    tree )
```

Gets the size of the map.

##### 9.1.2.2 [map\\_parser\(\)](#)

```
def src.parsing.map_parser.map_parser (  
    map_xml )
```

The main parser for the map xml file.

##### 9.1.2.3 [parse\\_cell\(\)](#)

```
def src.parsing.map_parser.parse_cell (  
    cell_object )
```

Parses a CellType attribute.



## Chapter 10

# Class Documentation

### 10.1 src.path.Archi Class Reference

#### Public Member Functions

- def `__init__` (self, prefix=None)
- def `get_src_file` (self, file\_path, mode='r')
- def `get_static_file` (self, file\_path, mode='r')
- def `get_xml_file` (self, file\_path, mode='r')
- def `list_files` (self, dir\_path)
- def `get_src_dir` (self, dir\_path)
- def `get_static_dir` (self, dir\_path)
- def `get_xml_dir` (self, dir\_path)

#### Public Attributes

- `main_directory`

#### 10.1.1 Detailed Description

This class manages the arhitecture of the project.  
It allows the user to travel in the file system of the game, to get the XML files and others (PNG, configuration files...)  
Moreover, it should be cross-platform compliant

#### 10.1.2 Member Function Documentation

##### 10.1.2.1 `get_src_dir()`

```
def src.path.Archi.get_src_dir (  
    self,  
    dir_path )
```

Gets the given `dir_path` with respect to the `src` folder.

#### 10.1.2.2 get\_src\_file()

```
def src.path.Archi.get_src_file (
    self,
    file_path,
    mode = 'r' )
```

Gets the path of the src directory.  
At least used by the src scripts.

#### 10.1.2.3 get\_static\_dir()

```
def src.path.Archi.get_static_dir (
    self,
    dir_path )
```

Gets the given dir\_path with respect to the static folder.

#### 10.1.2.4 get\_static\_file()

```
def src.path.Archi.get_static_file (
    self,
    file_path,
    mode = 'r' )
```

Gets the path of the static files directory. Static files are basically all graphical files, and a description of the common world

#### 10.1.2.5 get\_xml\_dir()

```
def src.path.Archi.get_xml_dir (
    self,
    dir_path )
```

Gets the given dir\_path with respect to the xml folder.

#### 10.1.2.6 get\_xml\_file()

```
def src.path.Archi.get_xml_file (
    self,
    file_path,
    mode = 'r' )
```

Gets the path of a xml file describing a world, a scenario, or a campaign.

## 10.1.2.7 list\_files()

```
def src.path.Archi.list_files (
    self,
    dir_path )
```

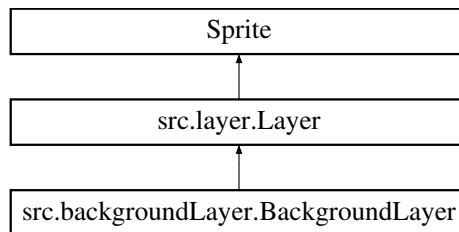
List all the files present in the dir\_path, if it is a dir.  
Else raise a FileNotFoundError.

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

- src/path.py

## 10.2 src.backgroundLayer.BackgroundLayer Class Reference

Inheritance diagram for src.backgroundLayer.BackgroundLayer:



## Public Member Functions

- def **\_\_init\_\_** (self, cell\_ids, size)
- def **init\_layer** (self)
- def **render** (self)
- def **collision\_test** (self, mouse\_pos)
- def **click\_update** (self, mouse\_pos)
- def **get\_grid\_info** (self, cell\_ids)

## Public Attributes

- **cell\_ids**
- **grid\_cell\_ids**
- **g\_width**
- **g\_height**
- **image**
- **rect**
- **mouse\_iso**
- **selected\_cell**

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

- src/backgroundLayer.py

## 10.3 src.world.BaseObject Class Reference

### Public Member Functions

- def `__init__` (self)
- def `__getattr__` (self, attr)
- def `__setattr__` (self, attr, val)
- def `load` (self, data)
- def `contextEval` (self, value)

### Public Attributes

- `params`
- `ident`
- `conditions`

### Static Public Attributes

- int `ident` = 0
- dictionary `ids` = {}

### 10.3.1 Detailed Description

Tout objet du monde

### 10.3.2 Member Function Documentation

#### 10.3.2.1 contextEval()

```
def src.world.BaseObject.contextEval (  
    self,  
    value )
```

Évalue une expression dans le contexte de l'objet pour les ordres

#### 10.3.2.2 load()

```
def src.world.BaseObject.load (  
    self,  
    data )
```

Charge l'objet depuis une structure Xml

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

- `src/world.py`

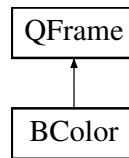


## 10.4 BColor Class Reference

The [BColor](#) class is a simple frame that offers color selection.

```
#include <bcolor.h>
```

Inheritance diagram for BColor:



### Public Slots

- void [setColor](#) (const QColor &c)
- void [setColorQuiet](#) (const QColor &c)
- void [setName](#) (const QString &s)
- void [setNameQuiet](#) (const QString &s)

### Signals

- void [colorChanged](#) (const QColor &)
- void [nameChanged](#) (const QString &)

### Public Member Functions

- [BColor](#) (QWidget \*parent=0)
- [BColor](#) (QColor c, QWidget \*parent=0)
- const QString & [name](#) () const
- const QColor & [color](#) () const

### Properties

- QColor [color](#)
- QString [name](#)

#### 10.4.1 Detailed Description

The [BColor](#) class is a simple frame that offers color selection.

#### 10.4.2 Constructor & Destructor Documentation

##### 10.4.2.1 BColor() [1/2]

```
BColor::BColor (
    QWidget * parent = 0 ) [explicit]
```

Constructs a new [BColor](#) object, with white as current color.

#### 10.4.2.2 BColor() [2/2]

```
BColor::BColor (
    QColor c,
    QWidget * parent = 0 ) [explicit]
```

Constructs a new [BColor](#) object and sets the color to c.

### 10.4.3 Member Function Documentation

#### 10.4.3.1 color()

```
const QColor& BColor::color ( ) const
```

Returns the current color of the selector.

See also

[setColor](#), [setColorQuiet](#), [colorChanged](#)

#### 10.4.3.2 colorChanged

```
void BColor::colorChanged (
    const QColor & ) [signal]
```

This signal is emitted when the color change, both when the user edit it or when [setColor](#) is called.

See also

[color](#), [setColorQuiet](#)

#### 10.4.3.3 name()

```
const QString& BColor::name ( ) const
```

Returns the name of the selector.

See also

[setName](#), [setNameQuiet](#), [nameChanged](#)

#### 10.4.3.4 nameChanged

```
void BColor::nameChanged (
    const QString & ) [signal]
```

This signal is emitted when the name change, when [setColor](#) is called.

See also

[name](#), [setNameQuiet](#)

#### 10.4.3.5 setColor

```
void BColor::setColor (
    const QColor & c ) [slot]
```

Sets the current color.

The signal [colorChanged](#) is emitted.

See also

[setColorQuiet](#), [color](#)

#### 10.4.3.6 setColorQuiet

```
void BColor::setColorQuiet (
    const QColor & c ) [slot]
```

Sets the current color.

The signal [colorChanged](#) is not emitted.

See also

[setColor](#), [color](#)

#### 10.4.3.7 setName

```
void BColor::setName (
    const QString & s ) [slot]
```

Sets the name of the selector.

The signal [nameChanged](#) is emitted.

See also

[setNameQuiet](#), [name](#)

#### 10.4.3.8 setNameQuiet

```
void BColor::setNameQuiet (
    const QString & s ) [slot]
```

Sets the name of the selector.

The signal [nameChanged](#) is not emitted.

See also

[setName](#), [name](#)

## 10.4.4 Property Documentation

### 10.4.4.1 color

```
const QColor & BColor::color [read], [write]
```

The current color that is displayed by the widget

See also

[setColor](#), [setColorQuiet](#), [colorChanged](#).

### 10.4.4.2 name

```
const QString & BColor::name [read], [write]
```

The name that is shown as title for the color chooser dialog used for user color definition purpose.

see also [setName](#), [setNameQuiet](#), and [nameChanged](#).

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

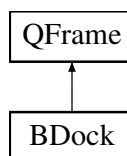
- [src/editor/GUI/Tabs/bcolor.h](#)
- [src/editor/GUI/Tabs/bcolor.cpp](#)

## 10.5 BDock Class Reference

The [BDock](#) class is the container for widget to display in a [BDocksZone](#).

```
#include <bdock.h>
```

Inheritance diagram for BDock:



### Public Slots

- void **setTitle** (QString s)
- void **setUnfold** (bool v)

### Signals

- void **mouseClick** (int i, const QPoint &p)
- void **mouseMove** (int i, const QPoint &p)
- void **mouseRelease** (int i, const QPoint &p)
- void **movementFinished** (int i)

## Public Member Functions

- **BDock** (QString title, [BDockWidget](#) \*dock, QWidget \*parent=0)
- bool **unfold** () const
- int **currentSize** () const
- void **setCurrentSize** (int t)
- void **setIndex** (int i)
- int **index** () const
- void **moveTo** (int i, bool inert=true)
- void **setLength** (int l)

## Properties

- bool **unfold**
- int **currentSize**

### 10.5.1 Detailed Description

The [BDock](#) class is the container for widget to display in a [BDocksZone](#).

A [BDock](#) is composed of a title and a QScrollArea in which a [BDockWidget](#) is displayed.

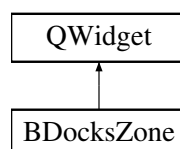
This container is movable within the [BDocksZone](#) it belongs to, and it can be hide.

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

- src/editor/GUI/Tabs/Docks/[bdock.h](#)
- src/editor/GUI/Tabs/Docks/bdock.cpp

## 10.6 BDocksZone Class Reference

Inheritance diagram for BDocksZone:



## Public Slots

- void **swap** (bool anim=true)

## Public Member Functions

- **BDocksZone** (QWidget \*parent=0)
- void **setUnfold** (bool u, bool anim=true)
- const [BinaryStateMachine](#) \* **states** () const
- int **length** () const
- void **setLength** (int t)
- ScrollBarMode **scrollBarMode** () const
- void **setScrollBarMode** (ScrollBarMode m)
- int **currentLength** () const
- void **addDock** (QString title, [BDockWidget](#) \*dock)

## Protected Slots

- void **setCurrentLenght** (int t)

## Properties

- int **length**
- int **currentLength**

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

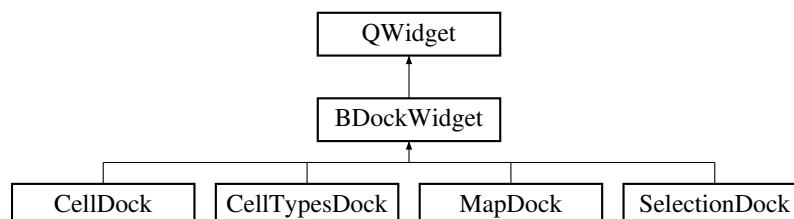
- src/editor/GUI/Tabs/Docks/bdockszone.h
- src/editor/GUI/Tabs/Docks/bdockszone.cpp

## 10.7 BDockWidget Class Reference

The [BDockWidget](#) class is the base class for game-related docks.

```
#include <bdockwidget.h>
```

Inheritance diagram for BDockWidget:



## Public Slots

- virtual void **updateGame** ()

## Signals

- void **gameModified** ()
- void **changeDockName** (QString)

## Public Member Functions

- **BDockWidget** (QWidget \*parent=0)
- void **setGame** (Game \*g)

## Protected Attributes

- Game \* **game**

### 10.7.1 Detailed Description

The [BDockWidget](#) class is the base class for game-related docks.

It provides common functions for set game, update, ...

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

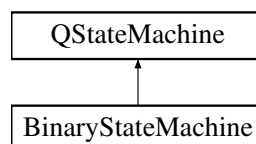
- src/editor/GUI/Tabs/Docks/bdockwidget.h
- src/editor/GUI/Tabs/Docks/bdockwidget.cpp

## 10.8 BinaryStateMachine Class Reference

The [BinaryStateMachine](#) class is a simple QStateMachine with two states.

```
#include <intertie.h>
```

Inheritance diagram for BinaryStateMachine:



## Public Slots

- void **swap** ()
- void **setPositive** (bool p)
- void **setNegative** (bool n)

## Signals

- void **swapped** (bool)
- void **\_\_swap** ()

## Public Member Functions

- **BinaryStateMachine** (QObject \*parent=0)
- void **defineProperty** (QObject \*obj, const char \*prop)
- void **defineProperty** (QObject \*obj, const char \*prop, QVariant yesValue, QVariant noValue)
- bool **isPositive** () const
- bool **isNegative** () const

### 10.8.1 Detailed Description

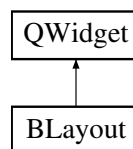
The [BinaryStateMachine](#) class is a simple QStateMachine with two states.

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

- src/editor/GUI/Tabs/Docks/intertie.h
- src/editor/GUI/Tabs/Docks/intertie.cpp

## 10.9 BLayout Class Reference

Inheritance diagram for BLayout:



## Signals

- void **sizeChanged** (int)
- void **showPoint** (int, int)

## Public Member Functions

- **BLayout** (QWidget \*parent=0)
- void **setOrientation** (Qt::Orientation o)
- void **insert** ([BDock](#) \*d, int ind=-1)
- void **setSpacing** (int e)
- void **setLength** (int t)
- int **spacing** () const

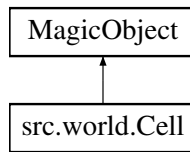
The documentation for this class was generated from the following files:

- src/editor/GUI/Tabs/Docks/bdockszone.h
- src/editor/GUI/Tabs/Docks/bdockszone.cpp



## 10.10 src.world.Cell Class Reference

Inheritance diagram for src.world.Cell:



### Public Member Functions

- `def __init__ (self)`

### Public Attributes

- **entities**
- **objects**

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

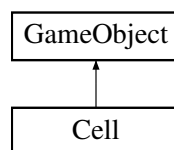
- `src/world.py`

## 10.11 Cell Class Reference

The [Cell](#) class.

```
#include <map.h>
```

Inheritance diagram for Cell:



### Public Member Functions

- **Cell** ([Game](#) \*g=nullptr, [GameObject](#) \*parent=nullptr)
- `bool isSelected () const`
- `void setSelected (bool s=true)`
- `void invertSelected ()`
- `void addSelection ()`
- `bool isPreSelected () const`
- `void confirmPreSelection (bool add=true)`
- `void clearPreSelection ()`

## Public Attributes

- ObjectListD(o, O, bject,, s, [Object](#)) private int **nbSel**
- bool **selectMod**

## Additional Inherited Members

### 10.11.1 Detailed Description

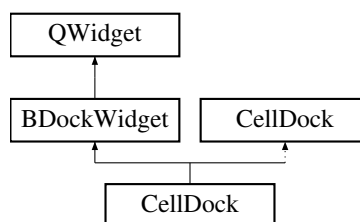
The [Cell](#) class.

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

- src/editor/Game/[map.h](#)
- src/editor/Game/map.cpp

## 10.12 CellDock Class Reference

Inheritance diagram for CellDock:



## Public Slots

- void **updateGame** ()
- void **selectionChanged** ()

## Public Member Functions

- **CellDock** (QWidget \*parent=0)

## Additional Inherited Members

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

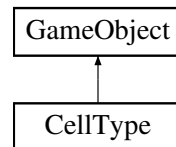
- src/editor/GUI/Tabs/celldock.h
- src/editor/GUI/Tabs/celldock.cpp

## 10.13 CellType Class Reference

The [CellType](#) class.

```
#include <map.h>
```

Inheritance diagram for CellType:



### Public Member Functions

- **CellType** ([Game](#) \*g, [GameObject](#) \*parent)

### Additional Inherited Members

#### 10.13.1 Detailed Description

The [CellType](#) class.

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

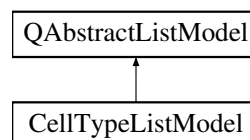
- src/editor/Game/[map.h](#)
- src/editor/Game/map.cpp

## 10.14 CellTypeListModel Class Reference

The [CellTypeListModel](#) class.

```
#include <mapslistmodel.h>
```

Inheritance diagram for CellTypeListModel:



### Public Member Functions

- **CellTypeListModel** ([World](#) \*w, QObject \*parent=0)
- int **rowCount** (const QModelIndex &parent) const Q\_DECL\_OVERRIDE
- QVariant **data** (const QModelIndex &index, int role) const Q\_DECL\_OVERRIDE
- bool **insertRows** (int row, int count, const QModelIndex &parent) Q\_DECL\_OVERRIDE
- bool **removeRows** (int row, int count, const QModelIndex &parent) Q\_DECL\_OVERRIDE

### 10.14.1 Detailed Description

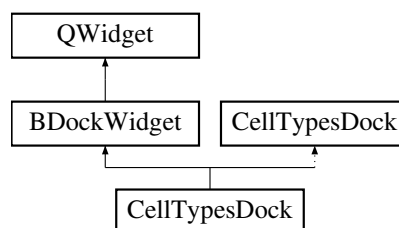
The [CellTypeListModel](#) class.

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

- [src/editor/Game/mapslistmodel.h](#)
- [src/editor/Game/mapslistmodel.cpp](#)

## 10.15 CellTypesDock Class Reference

Inheritance diagram for CellTypesDock:



### Public Member Functions

- **CellTypesDock** (QWidget \*parent=0)
- void **updateGame** ()

### Additional Inherited Members

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

- [src/editor/GUI/Tabs/celltypesdock.h](#)
- [src/editor/GUI/Tabs/celltypesdock.cpp](#)

## 10.16 src.cursescli.CellViewer Class Reference

### Public Member Functions

- def **\_\_init\_\_** (self, cell)
- def **display** (self, win)

### Public Attributes

- **cell**

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

- [src/cursescli.py](#)

## 10.17 src.character.Character Class Reference

### Public Member Functions

- def **\_\_init\_\_** (self, name, start\_pos, skin)
- def **render** (self)
- def **update** (self)
- def **update\_skin** (self)
- def **set\_path** (self, path)
- def **make\_skin** (self, skin)
- def **get\_cell\_pos\_by\_index** (self, index)
- def **move** (self)

### Public Attributes

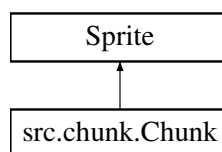
- **skin**
- **name**
- **action**
- **orientation**
- **image**
- **current\_image**
- **game\_frame\_count**
- **anim\_frame\_count**
- **current\_cell**
- **path**
- **pos\_offset**

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

- src/character.py

## 10.18 src.chunk.Chunk Class Reference

Inheritance diagram for src.chunk.Chunk:



### Public Member Functions

- def **\_\_init\_\_** (self, index, cells, map\_size)
- def **init\_chunk** (self)
- def **render** (self)
- def **scale\_chunk** (self, scale)
- def **update** (self, state, mouse\_pos)
- def **click\_trigger** (self, mouse\_pos)
- def **set\_state** (self, state)
- def **get\_state** (self)

## Public Attributes

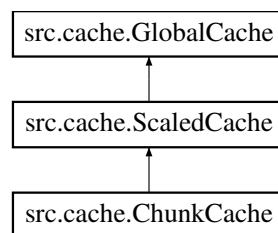
- **index**
- **cells**
- **g\_width**
- **g\_height**
- **g\_map\_height**
- **width**
- **height**
- **pos**
- **base\_rect**
- **layers**
- **image**

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

- `src/chunk.py`

## 10.19 `src.cache.ChunkCache` Class Reference

Inheritance diagram for `src.cache.ChunkCache`:



## Public Member Functions

- **def `init_chunk`** (`cls`, `chunk`)
- **def `init_elts`** (`cls`, `elts`)
- **def `init_chunks`** (`cls`, `chunks`)
- **def `get_chunk`** (`cls`, `chunk_index`, `scale=1`)
- **def `add_scaled`** (`cls`, `chunk_index`, `scale=1`)

## Static Public Attributes

- dictionary **`cache`** = {}

## Additional Inherited Members

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

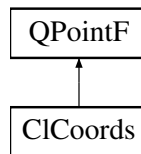
- `src/cache.py`

## 10.20 CCoords Class Reference

The [CCoords](#) class describe positions with cell coordinates.

```
#include <mappainter.h>
```

Inheritance diagram for CCoords:



### Public Member Functions

- **CCoords** (qreal x, qreal y)
- **CCoords** (const QPointF &p)

#### 10.20.1 Detailed Description

The [CCoords](#) class describe positions with cell coordinates.

Theses coordinates describe each point relatively to the cell grid. They correspond to the isometric 3D world.

See also

[RlCoords](#), [PtCoords](#), [PxCoords](#)

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

- src/editor/Game/[mappainter.h](#)

## 10.21 src.pygamecli.Client Class Reference

### Public Member Functions

- def **\_\_init\_\_** (self, path)
- def **\_\_del\_\_** (self)
- def **run** (self)
- def **frame\_counter** (self, n)
- def **update\_view** (self, mouse\_pos, mov\_speed\_x, mov\_speed\_y, deltat)
- def **get\_conf\_file** (self, path)
- def **get\_conf** (self, conf, type=str)
- def **init\_cache** (self)
- def **handleOrder** (self, ident, order)

### Public Attributes

- **net**
- **screen\_size**
- **screen**
- **world**
- **interface**
- **interactions**
- **perso**
- **orderDispatcher**
- **background**
- **conf**

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

- `src/pygamecli.py`

## 10.22 `src.client.Client` Class Reference

### Public Member Functions

- `def __init__ (self, path, Interface)`
- `def __del__ (self)`
- `def run (self)`
- `def getEntity (self)`
- `def main (self)`
- `def handleOrder (self, ident, order)`

### Public Attributes

- **loop**
- **net**
- **world**
- **interface**
- **interactions**
- **perso**
- **orderDispatcher**
- **netTask**

### 10.22.1 Detailed Description

Main class of the client process, gathering interface, world and networking



## 10.22.2 Constructor & Destructor Documentation

### 10.22.2.1 `__del__()`

```
def src.client.Client.__del__ (
    self )
```

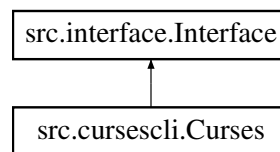
Kill network and interface

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

- `src/client.py`

## 10.23 src.cursescli.Curses Class Reference

Inheritance diagram for `src.cursescli.Curses`:



### Public Member Functions

- `def __init__ (self, w)`
- `def repaint (self)`
- `def end (self)`
- `def getEvent (self)`

### Public Attributes

- `win`
- `mv`

### 10.23.1 Detailed Description

ncurses-based UI

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

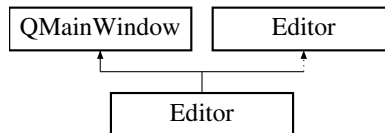
- `src/cursescli.py`

## 10.24 Editor Class Reference

The [Editor](#) class is the main window of the [Editor](#).

```
#include <editor.h>
```

Inheritance diagram for Editor:



### Public Member Functions

- **Editor** (QStringList args, QWidget \*parent=0)

#### 10.24.1 Detailed Description

The [Editor](#) class is the main window of the [Editor](#).

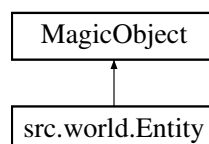
It is composed of tabs that offer editing facilities.

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

- src/editor/GUI/editor.h
- src/editor/GUI/editor.cpp

## 10.25 src.world.Entity Class Reference

Inheritance diagram for src.world.Entity:



### Public Member Functions

- def **\_\_init\_\_** (self)

## Public Attributes

- **quests**
- **inventory**
- **user**

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

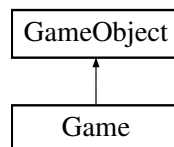
- `src/world.py`

## 10.26 Game Class Reference

The `Game` class gather the different parts needed to describe a game.

```
#include <game.h>
```

Inheritance diagram for `Game`:



## Public Member Functions

- `int newIdent ()`
- `World * world ()`
- `Map * currentMap ()`
- `void setCurrentMap (Map *m)`
- `void addImage (Image *im)`

## Additional Inherited Members

### 10.26.1 Detailed Description

The `Game` class gather the different parts needed to describe a game.

It contains mainly the `World`, and the resources used by it (images and strings)

For editing purposes, it contains also the active map (the one being editing)

## 10.26.2 Member Function Documentation

### 10.26.2.1 newIdent()

```
int Game::newIdent ( ) [inline]
```

Returns a new unused identifiers

#### Note

It should only be used by [GameObject](#) methods [GameObject::init](#) and [GameObject::GameObject](#).

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

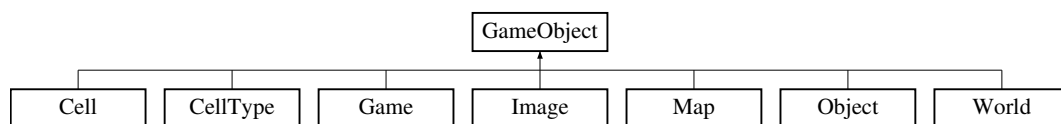
- [src/editor/Game/game.h](#)
- [src/editor/Game/game.cpp](#)

## 10.27 GameObject Class Reference

The [GameObject](#) class is the base class for every part of games.

```
#include <object.h>
```

Inheritance diagram for [GameObject](#):



### Public Member Functions

- [GameObject](#) ([Game](#) \*g=nullptr, [GameObject](#) \*parent=nullptr)
- void [init](#) ([Game](#) \*g, [GameObject](#) \*p)
- virtual bool [isValid](#) () const
- int [ident](#) () const
- const QDateTime & [lastInternalEdition](#) () const
- const QDateTime & [lastChildrenEdition](#) () const
- const QDateTime & [lastEdition](#) () const
- int [getParam](#) (const QString &p) const
- void [setParam](#) (const QString &p, int v)
- bool [hasParam](#) (const QString &p) const
- QList< QString > [params](#) () const
- bool [getFlag](#) (const QString &f) const
- void [setFlag](#) (const QString &f, bool v)
- bool [hasFlag](#) (const QString &f) const
- QList< QString > [flags](#) () const
- void [touch](#) ()
- void [addReference](#) ()
- void [removeReference](#) ()
- void [setParent](#) ([GameObject](#) \*p)

## Protected Member Functions

- void **addChild** ([GameObject](#) \*c)
- void **removeChild** ([GameObject](#) \*c)
- void **childrenTouched** (const QDateTime &d)

## Protected Attributes

- [GameObject](#) \* **parent**
- QMap< int, [GameObject](#) \* > **children**
- [Game](#) \* **game**
- int **id**
- int **nbRef**
- QMap< QString, int > **aParams**
- QMap< QString, bool > **aFlags**
- QString **fileName**
- QDateTime **lastEdit**
- QDateTime **lastChildEdit**

### 10.27.1 Detailed Description

The [GameObject](#) class is the base class for every part of games.

Each instance is identified by a game-wide unique identifier.

#### Object edition notification mechanism

To make the edition easier, each [GameObject](#) contains two QDateTime values :

- The most recent edition time, which is updated by the [touch](#) method
- The most recent child edition time, also updated by the [touch](#) method

#### Note

If the changes that are made in the object have to be detected by display/edition widgets, the [touch](#) function should be called.

To prevent the notification chain to be broken, the existing objects should always have a parent (except for the root object). This can be achieved using the [init](#) or [setParent](#) method, when the parent have not been given in the constructor. (see [object.h](#) for details)

#### References count

#### Todo

## 10.27.2 Constructor & Destructor Documentation

### 10.27.2.1 GameObject()

```
GameObject::GameObject (
    Game * g = nullptr,
    GameObject * parent = nullptr )
```

Constructs a new [GameObject](#) with parent `parent` and the reference to the game `g`.

#### Note

If these objects cannot be given to the constructor (case of an array of objects), the [init](#) method must be called after the creation to make the [GameObject](#) valid.

## 10.27.3 Member Function Documentation

### 10.27.3.1 flags()

```
QList<QString> GameObject::flags ( ) const [inline]
```

Returns the list of the registered flags

#### See also

[getFlag](#), [setFlag](#), [params](#)

### 10.27.3.2 getFlag()

```
bool GameObject::getFlag (
    const QString & f ) const [inline]
```

Returns the value of the `f` flag.

#### Note

If the requested parameter does not exists, a `false` value is returned, and the flags map stay unchanged

#### See also

[flags](#), [hasFlag](#), [setFlag](#), [getParam](#)

### 10.27.3.3 `getParam()`

```
int GameObject::getParam (
    const QString & p ) const [inline]
```

Returns the value of the `p` parameter.

#### Note

If the requested parameter does not exists, a null value is returned, and the parameters map stay unchanged

#### See also

[params](#), [hasParam](#), [setParam](#), [getFlag](#)

### 10.27.3.4 `hasFlag()`

```
bool GameObject::hasFlag (
    const QString & f ) const [inline]
```

Returns true if the flag `f` is register in the object's flags.

#### See also

[getFlag](#), [setFlag](#), [hasParam](#)

### 10.27.3.5 `hasParam()`

```
bool GameObject::hasParam (
    const QString & p ) const [inline]
```

Returns true if the parameter `is` register in the object's parameters.

#### See also

[getParam](#), [setParam](#), [hasFlag](#)

### 10.27.3.6 `ident()`

```
int GameObject::ident ( ) const [inline]
```

Returns the name wide unique identifier of the object.

#### See also

[init](#), [GameObject](#)

### 10.27.3.7 init()

```
void GameObject::init (
    Game * g,
    GameObject * p )
```

Initialises the object in case it had been construct with a NULL pointer (array of objects)

See also

[isValid](#), [GameObject](#)

### 10.27.3.8 isValid()

```
virtual bool GameObject::isValid ( ) const [inline], [virtual]
```

Returns true if the object has been initialised

See also

[init](#), [GameObject](#)

Reimplemented in [Image](#).

### 10.27.3.9 lastChildrenEdition()

```
const QDateTime& GameObject::lastChildrenEdition ( ) const [inline]
```

Returns the last time one of the object's children has been modified.

See also

[lastEdition](#), [lastInternalEdition](#)

### 10.27.3.10 lastEdition()

```
const QDateTime& GameObject::lastEdition ( ) const [inline]
```

Returns the last time a modification was made on the object or one of its children.

See also

[lastInternalEdition](#), [lastChildrenEdition](#)



### 10.27.3.11 lastInternalEdition()

```
const QDateTime& GameObject::lastInternalEdition ( ) const [inline]
```

Returns the last edition time.

See also

[lastEdition](#), [lastChildrenEdition](#)

### 10.27.3.12 params()

```
QList<QString> GameObject::params ( ) const [inline]
```

Returns the list of the registered paramters

See also

[getParam](#), [setParam](#), [flags](#)

### 10.27.3.13 setFlag()

```
void GameObject::setFlag (
    const QString & f,
    bool v ) [inline]
```

Set the value of the *f* flag.

Note

If the requested flag does not exists, it is created.

See also

[flags](#), [hasFlag](#), [getFlag](#), [setParam](#)

### 10.27.3.14 setParam()

```
void GameObject::setParam (
    const QString & p,
    int v ) [inline]
```

Set the value of the *p* parameter.

Note

If the requested parameter does not exists, it is created.

See also

[params](#), [hasParam](#), [getParam](#), [setFlag](#)

### 10.27.3.15 touch()

```
void GameObject::touch ( )
```

Notify the object and its parent that it has been modified.

#### See also

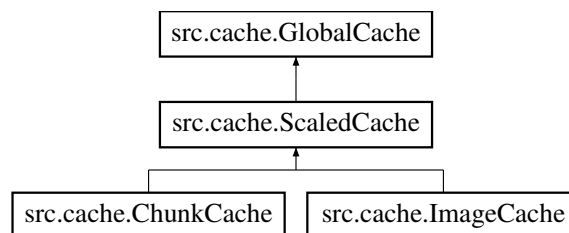
[lastInternalEdition](#), [lastChildrenEdition](#), [lastEdition](#).

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

- [src/editor/Game/object.h](#)
- [src/editor/Game/object.cpp](#)

## 10.28 src.cache.GlobalCache Class Reference

Inheritance diagram for src.cache.GlobalCache:



### Public Member Functions

- def **\_\_init\_\_** (self)
- def **set** (cls, key, value)
- def **get** (cls, key)
- def **clear** (cls)
- def **keys** (cls)
- def **show** (cls)

### Public Attributes

- **cache**

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

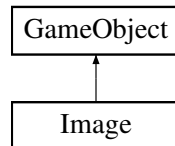
- [src/cache.py](#)

## 10.29 Image Class Reference

The [Image](#) class stores an external file in a QImage, and gives each image resources a unique identifier.

```
#include <object.h>
```

Inheritance diagram for Image:



### Public Member Functions

- **Image** ([Game](#) \*g, [GameObject](#) \*parent, const QString &fileName)
- bool **isValid** () const
- const QImage & **image** () const
- const QSize **size** () const
- void **update** ()

### Additional Inherited Members

#### 10.29.1 Detailed Description

The [Image](#) class stores an external file in a QImage, and gives each image resources a unique identifier.

#### 10.29.2 Member Function Documentation

##### 10.29.2.1 isValid()

```
bool Image::isValid ( ) const [inline], [virtual]
```

Returns true if the object has been initialised

See also

[init](#), [GameObject](#)

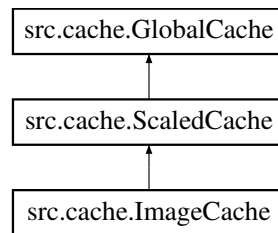
Reimplemented from [GameObject](#).

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

- src/editor/Game/[object.h](#)
- src/editor/Game/[object.cpp](#)

## 10.30 `src.cache.ImageCache` Class Reference

Inheritance diagram for `src.cache.ImageCache`:



### Public Member Functions

- def **init\_image\_from\_file** (cls, image)
- def **init\_image\_from\_surface** (cls, key, image)
- def **get\_image** (cls, image, scale=1)
- def **init\_elts** (cls, elts)
- def **init\_images** (cls, images)
- def **add\_scaled** (cls, image, scale=1)

### Static Public Attributes

- dictionary **cache** = {}

### Additional Inherited Members

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

- `src/cache.py`

## 10.31 `src.interactions.Interaction` Class Reference

### Public Member Functions

- def **\_\_init\_\_** (self)
- def **load** (self, dat)

### Public Attributes

- **target**
- **type**
- **key**
- **event**

### 10.31.1 Detailed Description

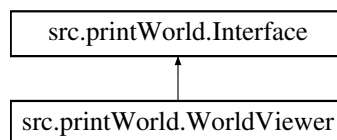
Représente une connection entre une entrée utilisateur  
et un événement à déclencher sur une cible

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

- src/interactions.py

## 10.32 src.printWorld.Interface Class Reference

Inheritance diagram for src.printWorld.Interface:

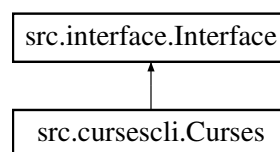


The documentation for this class was generated from the following file:

- src/printWorld.py

## 10.33 src.interface.Interface Class Reference

Inheritance diagram for src.interface.Interface:



### Public Member Functions

- def **\_\_init\_\_** (self, w)
- def **update** (self)
- def **repaint** (self)
- def **end** (self)
- def **getEvent** (self)

### Public Attributes

- **world**
- **lastUpdate**

### 10.33.1 Detailed Description

UI base-class, can be used as a dummy interface

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

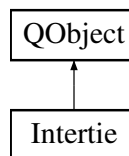
- `src/interface.py`

## 10.34 Intertie Class Reference

The [Intertie](#) class provide int that move smoothly from their value to an objective.

```
#include <intertie.h>
```

Inheritance diagram for Intertie:



### Public Slots

- void **setValue** (int v, bool inert=true)
- void **setMaximumSpeed** (int vM)
- void **setAcceleration** (int a)
- void **setUpdateInterval** (int d)

### Signals

- void **modificationFinished** (int)
- void **valueChanged** (int)

### Public Member Functions

- **Intertie** (QObject \*parent=0)
- int **value** () const
- void **link** (QObject \*obj, const char \*prop)

### 10.34.1 Detailed Description

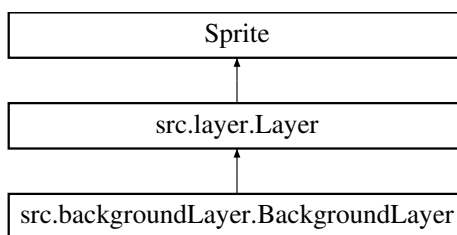
The [Intertie](#) class provide int that move smoothly from their value to an objective.

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

- `src/editor/GUI/Tabs/Docks/intertie.h`
- `src/editor/GUI/Tabs/Docks/intertie.cpp`

## 10.35 src.layer.Layer Class Reference

Inheritance diagram for src.layer.Layer:



### Public Member Functions

- def **\_\_init\_\_** (self, size)
- def **render** (self, surf)
- def **update** (self)
- def **get\_cell\_pos** (self, c\_line, c\_col, size\_image)
- def **make\_grid** (self, img\_set, cell\_ids)

### Public Attributes

- **cells**
- **g\_width**
- **g\_height**
- **size**

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

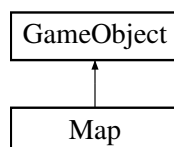
- src/layer.py

## 10.36 Map Class Reference

The [Map](#) class.

```
#include <map.h>
```

Inheritance diagram for Map:



## Public Member Functions

- **Map** ([Game](#) \*g, [GameObject](#) \*parent)
- **ParamGetter** (width) ParamGetter(height) QSize size() const
- void **setWidth** (int w)
- void **setHeight** (int h)
- void **resize** (int w, int h)
- [Cell](#) & **cell** (int i, int j) const
- [Cell](#) & **cell** (const QPoint &p) const
- void **selectAll** ()
- void **unselectAll** ()
- void **confirmPreSelection** (bool add=true)
- void **clearPreSelection** ()

## Additional Inherited Members

### 10.36.1 Detailed Description

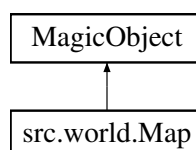
The [Map](#) class.

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

- src/editor/Game/[map.h](#)
- src/editor/Game/map.cpp

## 10.37 src.world.Map Class Reference

Inheritance diagram for src.world.Map:



## Public Member Functions

- def **\_\_init\_\_** (self)
- def **fill** (self)

## Public Attributes

- **cells**
- **cellsGrid**



### 10.37.1 Member Function Documentation

#### 10.37.1.1 fill()

```
def src.world.Map.fill (
    self )
```

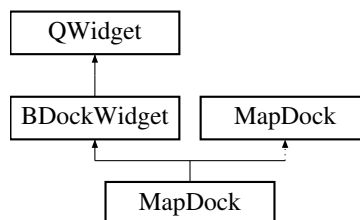
Complète les cases par défaut

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

- src/world.py

## 10.38 MapDock Class Reference

Inheritance diagram for MapDock:



### Public Member Functions

- **MapDock** (QWidget \*parent=0)
- void **updateGame** ()

### Additional Inherited Members

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

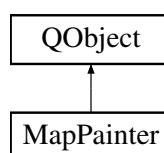
- src/editor/GUI/Tabs/mapdock.h
- src/editor/GUI/Tabs/mapdock.cpp

## 10.39 MapPainter Class Reference

The [MapPainter](#) class that can paint a [Map](#) using a QPainter.

```
#include <mappainter.h>
```

Inheritance diagram for MapPainter:



## Public Types

- enum `Element` {  
`Nothing` = 0, `CellBackground` = 1, `Grid` = 2, `CellSelection` = 4,  
`CellHighlighting` = 8, `Objects` = 16, `All` = 31 }

*The Element enum describes the different elements that can be render.*

## Signals

- void `mapSizeChanged` (QSize)
- void `viewCenterChanged` (QPoint)

## Public Member Functions

- `MapPainter` (QObject \*parent=0)
- `MapPainter` (Map \*m, QObject \*parent=0)
- void `setPaintedElement` (`Element` e, bool painted=true)
- void `setPaintedElements` (`Element` e)
- void `setMap` (Map \*m)
- void `paint` (QPainter &p)
- const QImage & `render` ()
- `RICoords` `viewCenter` () const
- void `setViewCenter` (`RICoords` relativeCenter)
- void `setViewCenter` (double relativeCenterX, double relativeCenterY)
- void `setViewCenterQuiet` (double x, double y)
- double `scale` () const
- void `setScale` (double `scale`)
- void `setScaleDomain` (double scaleMin, double scaleMax)
- bool `setHighlightedCell` (const `CICoords` &p)
- bool `setHighlightedCell` (int i, int j)
- QPoint `highlightedCell` () const
- bool `hasHighlightedCell` () const
- bool `isCell` (const `CICoords` &c) const
- void `resize` (QSize s)
- void `resize` (int wi, int he)
- QSize `size` () const
- void `zoom` (double factor, QPointF fixedPoint)
- QPair< bool, bool > `move` (`PxCoords` delta, QPointF center)
- QSize `virtualSize` () const
- `PxCoords` `ptToPxI` (`PtCoords` p) const
- `PtCoords` `pxItoPt` (`PxCoords` p) const
- `PtCoords` `cooToPt` (`CICoords` p) const
- `CICoords` `ptToCoo` (`PtCoords` p) const
- `PxCoords` `cooToPxI` (`CICoords` p) const
- `CICoords` `pxItoCoo` (`PxCoords` p) const
- `PtCoords` `indToPt` (int i, int j) const
- const QColor & `selectedCellColor` () const
- const QColor & `preSelectedCellColor` () const
- void `setSelectedCellColor` (const QColor &c)
- void `setPreSelectedCellColor` (const QColor &c)

### 10.39.1 Detailed Description

The [MapPainter](#) class that can paint a [Map](#) using a QPainter.

The class take charge of the different ratios of the [map](#) rendering and the area in which it will be rendered.

#### Note

The view is kept updated with the associated [map](#) at each [paint](#) or [render](#) call. It is thus just needed to call one of these functions to update the view after a modification.

To ensure a type checking security about the different types of coordinates that are used, four different types that inherit from QPointF are used : [RICoords](#), [CICoords](#), [PtCoords](#) and [PxCoords](#)

### 10.39.2 Member Enumeration Documentation

#### 10.39.2.1 Element

```
enum MapPainter::Element
```

The Element enum discribes the different elements that can be render.

This includes both map's objects and user interaction and editing elements.

Element value can be used as flags using the operators `|`, `&`, `^`.

#### See also

[Cell](#), [CellType](#)

#### Enumerator

Nothing	Represent no elements
CellBackground	The bacground associated to the <a href="#">cell type</a>
Grid	A thin grid that separate <a href="#">cells</a>
CellSelection	Graphical information about the selection state
CellHighlighting	Graphical visualisation of the <a href="#">cells</a> the mouse is over
Objects	The objects that lay on the <a href="#">cells</a>
All	Represent all elements

### 10.39.3 Constructor & Destructor Documentation

#### 10.39.3.1 MapPainter() [1/2]

```
MapPainter::MapPainter (
    QObject * parent = 0 )
```

Constructs a new [MapPainter](#) with a default size of (42,42).

### 10.39.3.2 MapPainter() [2/2]

```
MapPainter::MapPainter (
    Map * m,
    QObject * parent = 0 )
```

Constructs a new [MapPainter](#) with a default size of (42,42), and loads the [map](#) m.

## 10.39.4 Member Function Documentation

### 10.39.4.1 cooToPt()

```
PtCoords MapPainter::cooToPt (
    ClCoords p ) const
```

Converts cells indice to virtual point coordinates

### 10.39.4.2 cooToPxI()

```
PxCoords MapPainter::cooToPxI (
    ClCoords p ) const
```

Convenient function equivalent to [ptToPxI\(cooToPt\(p\)\)](#)

### 10.39.4.3 hasHighlightedCell()

```
bool MapPainter::hasHighlightedCell ( ) const
```

Returns true if a [cell](#) is highligthed.

See also

[highlightedCell](#), [setHighlightedCell](#)

### 10.39.4.4 highlightedCell()

```
QPoint MapPainter::highlightedCell ( ) const
```

Returns the integer index of the [cell](#) the is highlighted.

See also

[setHighlightedCell](#), [hasHighlightedCell](#)

#### 10.39.4.5 indToPt()

```
PtCoords MapPainter::indToPt (
    int i,
    int j ) const
```

Converts to coordinates

#### 10.39.4.6 isCell()

```
bool MapPainter::isCell (
    const ClCoords & c ) const
```

Returns true if the coordinate `c` correspond to a [cell](#).

#### 10.39.4.7 mapSizeChanged

```
void MapPainter::mapSizeChanged (
    QSize ) [signal]
```

This signal is emitted when the total size of the [map](#)'s view change.

It appends mainly during scale change and modification on the [map](#) (resize, angles setting, ...).

#### 10.39.4.8 move()

```
QPair< bool, bool > MapPainter::move (
    PxCoords delta,
    QPointF center )
```

Change the center position from the given center and a pixel difference.

The return value indicate if the expected center was valid (regarding x or y coordinate).

See also

[setViewCenter](#)

#### 10.39.4.9 paint()

```
void MapPainter::paint (
    QPainter & p )
```

Draws the map in the QPaintDevice.

See also

[render](#)

#### 10.39.4.10 preSelectedCellColor()

```
const QColor & MapPainter::preSelectedCellColor ( ) const
```

Returns the color of the filter that is applied to pre-selected cells.

See also

[setPreSelectedCellColor](#), [selectedCellColor](#)

#### 10.39.4.11 ptToCoo()

```
ClCoords MapPainter::ptToCoo (
    PtCoords p ) const
```

Converts virtual point to cell indice

#### 10.39.4.12 ptToPxl()

```
PxCoords MapPainter::ptToPxl (
    PtCoords p ) const
```

Converts virtual point to real pixel coordinates

#### 10.39.4.13 pxlToCoo()

```
ClCoords MapPainter::pxlToCoo (
    PxCoords p ) const
```

Convenient function equivalent to [ptToCoo\(pxToPt\(p\)\)](#)

#### 10.39.4.14 pxlToPt()

```
PtCoords MapPainter::pxlToPt (
    PxCoords p ) const
```

Converts real pixel to virtual point coordinates

#### 10.39.4.15 render()

```
const QImage & MapPainter::render ( )
```

Provides a QImage with a view of the map.

See also

[paint](#)

**10.39.4.16** `resize()` [1/2]

```
void MapPainter::resize (
    QSize s )
```

Change the size of the view, *ie* the rectangle in which the map will be render.

See also

[size](#)

**10.39.4.17** `resize()` [2/2]

```
void MapPainter::resize (
    int wi,
    int he )
```

This is an overload function, see [resize](#)

**10.39.4.18** `scale()`

```
double MapPainter::scale ( ) const
```

Returns the current scale of the view.

See also

[setScale](#)

**10.39.4.19** `selectedCellColor()`

```
const QColor & MapPainter::selectedCellColor ( ) const
```

Returns the color of the filter that is applied to selected cells.

See also

[setSelectedCellColor](#), [preSelectedCellColor](#)

**10.39.4.20** `setHighlightedCell()` [1/2]

```
bool MapPainter::setHighlightedCell (
    const ClCoords & p )
```

Sets the highlighted [cell](#) to the one at the [ClCoords](#) p

See also

[highlightedCell](#), [hasHighlightedCell](#)

**10.39.4.21 setHighlightedCell()** [2/2]

```
bool MapPainter::setHighlightedCell (
    int i,
    int j )
```

This is an overload function, see [setViewCenter](#).

**10.39.4.22 setMap()**

```
void MapPainter::setMap (
    Map * m )
```

Loads the [map](#), computing the new size of the view area.

**10.39.4.23 setPaintedElement()**

```
void MapPainter::setPaintedElement (
    MapPainter::Element e,
    bool painted = true )
```

Enables or disables the render of an [element](#).

See also

[setPaintedElements](#)

**10.39.4.24 setPaintedElements()**

```
void MapPainter::setPaintedElements (
    Element e )
```

Sets the rendered [elements](#).

See also

[setPaintedElement](#)

**10.39.4.25 setPreSelectedCellColor()**

```
void MapPainter::setPreSelectedCellColor (
    const QColor & c )
```

Sets the color of the filter that is applied to pre-selected cells.

See also

[preSelectedCellColor](#), [setSelectedCellColor](#)



#### 10.39.4.26 `setScale()`

```
void MapPainter::setScale (
    double scale )
```

Sets the current view scale. This closest value in the scale domain will be used.

See also

[scale](#), [setScaleDomain](#)

#### 10.39.4.27 `setScaleDomain()`

```
void MapPainter::setScaleDomain (
    double scaleMin,
    double scaleMax )
```

Sets the valid values for the scale.

See also

[scale](#), [setScale](#)

#### 10.39.4.28 `setSelectedCellColor()`

```
void MapPainter::setSelectedCellColor (
    const QColor & c )
```

Sets the color of the filter that is applied to selected cells.

See also

[selectedCellColor](#), [setPreSelectedCellColor](#)

#### 10.39.4.29 `setViewCenter()` [1/2]

```
void MapPainter::setViewCenter (
    RlCoords relativeCenter )
```

Change the view center, using relative coordinates.

If the new center is invalid (the view exceed the map area), the closest valid center is used.

See also

[viewCenter](#)

#### 10.39.4.30 `setViewCenter()` [2/2]

```
void MapPainter::setViewCenter (
    double relativeCenterX,
    double relativeCenterY )
```

This is an overload function, see [setViewCenter](#).

#### 10.39.4.31 `setViewCenterQuiet()`

```
void MapPainter::setViewCenterQuiet (
    double x,
    double y )
```

does the same as [setViewCenter](#), without emitting the signal `viewCenterChanged` to avoid event loop.

#### 10.39.4.32 `size()`

```
QSize MapPainter::size ( ) const
```

Return the size of the rectangle in which the map is render. This is also the size of the image returned by [render](#).

See also

[resize](#)

#### 10.39.4.33 `viewCenter()`

```
RlCoords MapPainter::viewCenter ( ) const
```

Return the relative coordinates of the current view center.

See also

[setViewCenter](#)

#### 10.39.4.34 `viewCenterChanged`

```
void MapPainter::viewCenterChanged (
    QPoint ) [signal]
```

This signal is emitted when the center of the [map](#) change.

It appends mainly during moving on the view and zooming.

## 10.39.4.35 virtualSize()

```
QSize MapPainter::virtualSize ( ) const
```

Computes the total size of the image of the map

## 10.39.4.36 zoom()

```
void MapPainter::zoom (
    double factor,
    QPointF fixedPoint )
```

Multiplying the scale of the view by factor, trying to leave the point center at the same position.

## Note

It is not always possible to keep this point fixed, in particularity when the view is resulting view would exceed the map region. In that case, the center is adapt to minimise the difference.

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

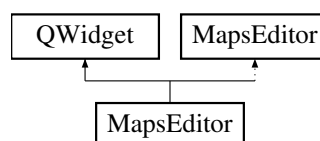
- [src/editor/Game/mappainter.h](#)
- [src/editor/Game/mappainter.cpp](#)

## 10.40 MapsEditor Class Reference

The [MapsEditor](#) class is the tab offering map editing facilities.

```
#include <mapseditor.h>
```

Inheritance diagram for MapsEditor:



## Public Slots

- void **updateGame** ()

## Public Member Functions

- **MapsEditor** (QWidget \*parent=0)
- void **setGame** ([Game](#) \*g)

### 10.40.1 Detailed Description

The [MapsEditor](#) class is the tab offering map editing facilities.

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

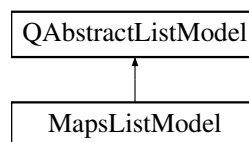
- `src/editor/GUI/Tabs/mapseditor.h`
- `src/editor/GUI/Tabs/mapseditor.cpp`

## 10.41 MapsListModel Class Reference

The [MapsListModel](#) class provides a presentation class for the Qt Model-View framework.

```
#include <mapslistmodel.h>
```

Inheritance diagram for MapsListModel:



### Public Slots

- `void update ()`

### Public Member Functions

- **MapsListModel** ([World](#) \*w, QObject \*parent=0)
- `int rowCount (const QModelIndex &parent) const Q_DECL_OVERRIDE`
- `QVariant data (const QModelIndex &index, int role) const Q_DECL_OVERRIDE`
- `bool insertRows (int row, int count, const QModelIndex &parent) Q_DECL_OVERRIDE`
- `bool removeRows (int row, int count, const QModelIndex &parent) Q_DECL_OVERRIDE`

### 10.41.1 Detailed Description

The [MapsListModel](#) class provides a presentation class for the Qt Model-View framework.

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

- `src/editor/Game/mapslistmodel.h`
- `src/editor/Game/mapslistmodel.cpp`

## 10.42 src.cursescli.MapViewer Class Reference

### Public Member Functions

- def **\_\_init\_\_** (self, m, w)
- def **display** (self, win)

### Public Attributes

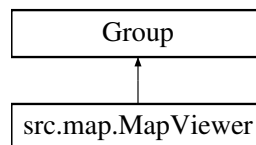
- **map**
- **world**
- **cellViews**

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

- src/cursescli.py

## 10.43 src.map.MapViewer Class Reference

Inheritance diagram for src.map.MapViewer:



### Public Member Functions

- def **\_\_init\_\_** (self, currentMap, w)
- def **load\_chunks** (self, bg)
- def **make\_walkables** (self, bg)
- def **zoom** (self, dz)
- def **move** (self, dx, dy)
- def **render** (self)
- def **onscreen\_chunks** (self)
- def **update** (self)
- def **update\_chunks** (self)
- def **propagate\_trigger** (self, event)
- def **compute\_path** (self, start\_pos, end\_pos)
- def **load\_bg** (self, path)

## Public Attributes

- **map**
- **world**
- **cm\_height**
- **width**
- **height**
- **screen\_cgwidth**
- **screen\_cgheight**
- **walkablesGraph**
- **last\_curr\_chunk**
- **current\_chunk**
- **pos\_offset**
- **chunk\_pos**
- **chunks\_state**

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

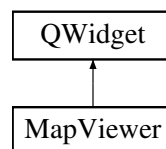
- `src/map.py`

## 10.44 MapViewer Class Reference

The [MapViewer](#) class provides a widget to display and edit a [Map](#) using a [MapPainter](#).

```
#include <mapviewer.h>
```

Inheritance diagram for MapViewer:



## Public Types

- enum [SelectionMode](#) { [PencilSelection](#), [RectangleSelection](#), [RegionSelection](#) }  
*The SelectionMode enum describes the different behaviour the cell selection can have.*

## Public Slots

- void **updateRequest** ()

## Signals

- void **viewSizeChanged** (QSize)
- void **selectionChanged** ()

## Public Member Functions

- **MapView** (QWidget \*parent=0)
- void **setMap** ([Map](#) \*m)
- void **updateMap** ()
- [MapPainter](#) & **mapPainter** ()
- void **setSelectionMode** ([SelectionMode](#) m)
- [SelectionMode](#) **selectionMode** () const

### 10.44.1 Detailed Description

The [MapView](#) class provides a widget to display and edit a [Map](#) using a [MapPainter](#).

Several [selection modes](#) are available. Combined with the `Ctrl` and `Shift` modifiers, a colossal amount of selection possibilities is offered. See [SelectionMode](#) for more information.

### 10.44.2 Member Enumeration Documentation

#### 10.44.2.1 SelectionMode

```
enum MapView::SelectionMode
```

The `SelectionMode` enum describes the different behaviour the cell selection can have.

The selection's behaviour is based on two parameters :

- The keyboard modifiers that are pressed during selection.
- The current selection mode

If the `Ctrl` modifier is pressed, the past selected cells stay selected otherwise, they are all unselected

If the `Shift` modifier is pressed, the selection is inverted.

Three modes of selection exist :

#### Enumerator

<code>PencilSelection</code>	The <a href="#">cells</a> under the cursor are selected
<code>RectangleSelection</code>	The <a href="#">cells</a> inside the rectangle defined by the clicked cell and the cell under the cursor are selected
<code>RegionSelection</code>	The <a href="#">cells</a> inside the region drawn by cursor's moves are selected

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

- `src/editor/GUI/Tabs/mapviewer.h`
- `src/editor/GUI/Tabs/mapviewer.cpp`

## 10.45 src.network.NetworkClient Class Reference

### Public Member Functions

- def **\_\_init\_\_** (self, handle)
- def **askEntity** (self, ent)
- def **connect** (self)
- def **run** (self)
- def **send** (self, m)
- def **sendEvent** (self, obj, event)
- def **kill** (self)

### Public Attributes

- **handle**
- **alive**
- **writer**

### 10.45.1 Detailed Description

This class manages the network activities of the client.  
It allows the client to send messages (describing events) to the server.

### 10.45.2 Member Function Documentation

#### 10.45.2.1 askEntity()

```
def src.network.NetworkClient.askEntity (
    self,
    ent )
```

Ask for an entity

#### 10.45.2.2 run()

```
def src.network.NetworkClient.run (
    self )
```

Main loop of the client's network task. After connecting the socket to the server, wait for orders and handle them immediately until connection ends.



### 10.45.2.3 sendEvent()

```
def src.network.NetworkClient.sendEvent (
    self,
    obj,
    event )
```

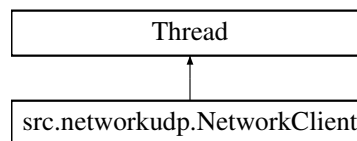
Send an event to the server in a formatted way, specifying the id of the object to affect.

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

- src/network.py

## 10.46 src.networkudp.NetworkClient Class Reference

Inheritance diagram for src.networkudp.NetworkClient:



### Public Member Functions

- def **\_\_init\_\_** (self, handle)
- def **run** (self)
- def **send** (self, m)
- def **sendEvent** (self, obj, eve)
- def **kill** (self)

### Public Attributes

- **handle**
- **soc**
- **alive**

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

- src/networkudp.py

## 10.47 src.network.NetworkServer Class Reference

### Public Member Functions

- def `__init__` (self, handle, loop)
- def `waitForClients` (self, n)
- def `connect` (self, reader, writer)
- def `run` (self)
- def `sendOrder` (self, ident, order)
- def `broadcast` (self, m)

### Public Attributes

- `handle`
- `loop`
- `alive`
- `connections`
- `server`

### 10.47.1 Detailed Description

This class is the task of the server that manages clients connections on his port. It keeps the list of connected clients and provides method to broadcast messages to all the clients.

### 10.47.2 Constructor & Destructor Documentation

#### 10.47.2.1 `__init__()`

```
def src.network.NetworkServer.__init__ (
    self,
    handle,
    loop )
```

The server listen on the port specified in `const.py`.

### 10.47.3 Member Function Documentation

#### 10.47.3.1 `run()`

```
def src.network.NetworkServer.run (
    self )
```

When a client tries to connect, the server adds him to the list and creates a new task managing communications with this client.

## 10.47.3.2 sendOrder()

```
def src.network.NetworkServer.sendOrder (
    self,
    ident,
    order )
```

Send an order to all connected clients by broadcasting messages to all threads in the list.

## 10.47.3.3 waitForClients()

```
def src.network.NetworkServer.waitForClients (
    self,
    n )
```

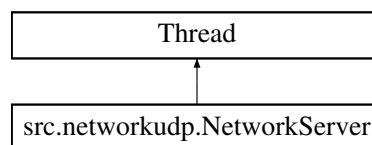
Block until n clients are connected

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

- src/network.py

## 10.48 src.networkudp.NetworkServer Class Reference

Inheritance diagram for src.networkudp.NetworkServer:



## Public Member Functions

- def **\_\_init\_\_** (self, handle)
- def **waitForClients** (self, n)
- def **run** (self)
- def **sendOrder** (self, ident, order)
- def **broadcast** (self, m)
- def **kill** (self)

## Public Attributes

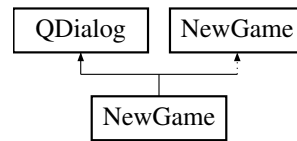
- **handle**
- **soc**
- **alive**
- **addr**

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

- src/networkudp.py

## 10.49 NewGame Class Reference

Inheritance diagram for NewGame:



### Public Member Functions

- **NewGame** (QWidget \*parent=0)
- QString **name** () const
- QString **folder** () const
- bool **createFolder** () const

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

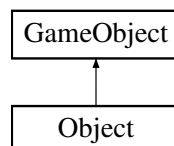
- src/editor/GUI/newgame.h
- src/editor/GUI/newgame.cpp

## 10.50 Object Class Reference

The [Object](#) class.

```
#include <object.h>
```

Inheritance diagram for Object:



### Public Member Functions

- **Object** ([Game](#) \*g, [GameObject](#) \*parent)

### Additional Inherited Members

#### 10.50.1 Detailed Description

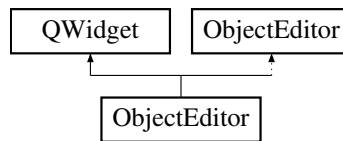
The [Object](#) class.

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

- src/editor/Game/[object.h](#)
- src/editor/Game/object.cpp

## 10.51 ObjectEditor Class Reference

Inheritance diagram for ObjectEditor:



### Public Member Functions

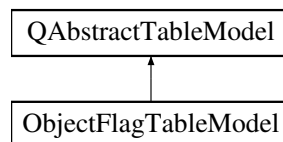
- **ObjectEditor** (QWidget \*parent=0)
- void **setGame** (Game \*g)

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

- src/editor/GUI/Tabs/objecteditor.h
- src/editor/GUI/Tabs/objecteditor.cpp

## 10.52 ObjectFlagTableModel Class Reference

Inheritance diagram for ObjectFlagTableModel:



### Public Member Functions

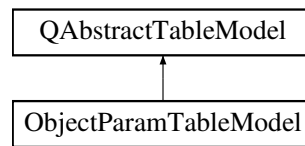
- **ObjectFlagTableModel** (GameObject \*obj, QObject \*parent=0)
- int **rowCount** (const QModelIndex &parent) const Q\_DECL\_OVERRIDE
- int **columnCount** (const QModelIndex &parent) const Q\_DECL\_OVERRIDE
- QVariant **data** (const QModelIndex &index, int role) const Q\_DECL\_OVERRIDE
- Qt::ItemFlags **flags** (const QModelIndex &index) const
- bool **setData** (const QModelIndex &index, const QVariant &value, int role)
- QVariant **headerData** (int section, Qt::Orientation orientation, int role) const

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

- src/editor/Game/mapslistmodel.h
- src/editor/Game/mapslistmodel.cpp

## 10.53 ObjectParamTableModel Class Reference

Inheritance diagram for ObjectParamTableModel:



### Public Member Functions

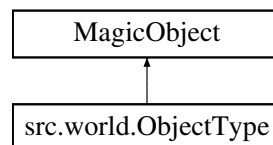
- **ObjectParamTableModel** ([GameObject](#) \*obj, QObject \*parent=0)
- int **rowCount** (const QModelIndex &parent) const Q\_DECL\_OVERRIDE
- int **columnCount** (const QModelIndex &parent) const Q\_DECL\_OVERRIDE
- QVariant **data** (const QModelIndex &index, int role) const Q\_DECL\_OVERRIDE
- Qt::ItemFlags **flags** (const QModelIndex &index) const
- bool **setData** (const QModelIndex &index, const QVariant &value, int role)
- QVariant **headerData** (int section, Qt::Orientation orientation, int role) const

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

- src/editor/Game/[mapslistmodel.h](#)
- src/editor/Game/mapslistmodel.cpp

## 10.54 src.world.ObjectType Class Reference

Inheritance diagram for src.world.ObjectType:



### Public Member Functions

- def **\_\_init\_\_** (self, typ=MagicObject)
- def [create](#) (self)
- def **\_\_str\_\_** (self)

### Public Attributes

- **type**

### 10.54.1 Detailed Description

Les types d'objets (au sens informatique)

### 10.54.2 Member Function Documentation

#### 10.54.2.1 create()

```
def src.world.ObjectType.create (
    self )
```

Instantiation d'un objet à partir du type

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

- src/world.py

## 10.55 Options Struct Reference

The [Options](#) class provides session-independant options and preferences.

```
#include <options.h>
```

### Public Member Functions

- `template<class T >`  
T [load](#) (QString group, QString opt)
- `template<class T >`  
void [save](#) (QString group, QString opt, T val)
- `template<class T >`  
void [setDefault](#) (QString group, QString opt, T val)
- bool [isAdjustable](#) (QString group, QString opt, bool adjust=true)
- void [setAdjustable](#) (QString group, QString opt, bool adjust)
- void [reinitialise](#) (QString group="")

### Static Public Member Functions

- static [Options](#) & [options](#) ()

### 10.55.1 Detailed Description

The [Options](#) class provides session-independant options and preferences.

## Features

The [Options](#) class aims at storing global options, that are available at any place in the entire application. The preferences are permanently stored and remain between the separate sessions and windows.

Two sorts of options exist :

- The adjustable ones : the value of the option change when [save](#) is called.
- The non-adjustable ones : the value of the option doesn't change is [save](#) is called, the option must be modified with [setAdjustable](#).

The sort of option can be set with the [setAdjustable](#) function.

## Design

The [Options](#) class is designed following the **Singleton design pattern**. The constructor is thus private, and the only [Options](#) instance is created at the first call of [options](#).

QSetting is used internally, see Qt's documentation for details about the storing mechanisms.

## Reading and writing existing options

To read or write options, the [Options](#) instance must be retrieved, using the [options](#) function, then the [load](#) and [save](#) functions can be called.

## Adding options

To add a new option, it is only needed to add a default hard coded value, using the [Default](#) and [DefaultF](#) macros in the [Options](#) constructor.

It is strongly advised to use macro to define new options group (see [WIN](#), for an example).

## Note

To use [Options](#) with custom types (other than C++ standard), the defining header of the type must be included at the top of the [options.h](#) file, in order to be used in the default value declaration.

## Warning

Pointer objects are not supported, and the result of the use of the [Options](#) class with such values is undefined.

## See also

[options.h](#)



## 10.55.2 Member Function Documentation

### 10.55.2.1 isAdjustable()

```
bool Options::isAdjustable (
    QString group,
    QString opt,
    bool adjust = true )
```

Returns `true` if the option defined by its group and name is adjustable, `false` elsewhere.

See also

[setAdjustable](#)

### 10.55.2.2 load()

```
template<class T >
T Options::load (
    QString group,
    QString opt ) [inline]
```

Reads an option defined by its group and name.

Note

The template argument must be precised since it can't be deduced from arguments' types.

Warning

If the option type and the reading type mismatch, an default null value is returned.

See also

[save](#)

### 10.55.2.3 options()

```
Options & Options::options ( ) [static]
```

Returns the unique [Options](#) instance.

### 10.55.2.4 reinitialise()

```
void Options::reinitialise (
    QString group = "" )
```

Clear all options from the group. If `group == ""`, all entries are deleted.

#### 10.55.2.5 save()

```
template<class T >
void Options::save (
    QString group,
    QString opt,
    T val ) [inline]
```

Writes the new value of the options defined by its group and name, if the option is adjustable. See [Options](#) for details about options types.

##### Note

The template argument can be omitted since it would be deduced from the value argument.

##### See also

[setDefault](#), [load](#)

#### 10.55.2.6 setAdjustable()

```
void Options::setAdjustable (
    QString group,
    QString opt,
    bool adjust )
```

Sets if the option defined by its group and name is adjustable.

##### See also

[isAdjustable](#)

#### 10.55.2.7 setDefault()

```
template<class T >
void Options::setDefault (
    QString group,
    QString opt,
    T val ) [inline]
```

Writes the new value of the options defined by its group and name, whatever the option type is. See [Options](#) for details about options types.

##### Note

The template argument can be omitted since it would be deduced from the value argument.

##### See also

[save](#), [load](#)

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

- [src/editor/GUI/options.h](#)
- [src/editor/GUI/options.cpp](#)

## 10.56 src.orders.Order Class Reference

### Public Member Functions

- def **\_\_init\_\_** (self)
- def **\_\_getattr\_\_** (self, attr)
- def **\_\_setattr\_\_** (self, attr, val)
- def **copy** (self)
- def **load** (self, dat, named)
- def **toBytes** (self)
- def **fromBytes** (self, b)

### Public Attributes

- **type**
- **args**

### Static Public Attributes

- list **params** = [None] \* (len(OrderType)+1)

### 10.56.1 Detailed Description

Représente une modification à apporter au monde

### 10.56.2 Member Function Documentation

#### 10.56.2.1 fromBytes()

```
def src.orders.Order.fromBytes (  
    self,  
    b )
```

Récupère l'ordre à partir d'un bytes réseau

#### 10.56.2.2 load()

```
def src.orders.Order.load (  
    self,  
    dat,  
    named )
```

Initialise l'ordre avec une structure provenant d'un Xml

### 10.56.2.3 toBytes()

```
def src.orders.Order.toBytes (
    self )
```

Bytes pour envoyer l'ordre sur le réseau

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

- `src/orders.py`

## 10.57 src.orders.OrderDispatcher Class Reference

### Public Member Functions

- `def __init__` (self, world, handle, timer)
- `def treat` (self, emitter, order)

### Public Attributes

- **world**
- **handle**
- **timer**

### 10.57.1 Detailed Description

Traite les ordres pour le client ou le serveur

### 10.57.2 Member Function Documentation

#### 10.57.2.1 treat()

```
def src.orders.OrderDispatcher.treat (
    self,
    emitter,
    order )
```

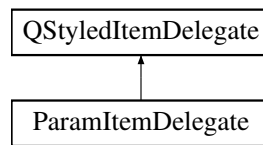
Traite un ordre et renvoie l'éventuel ordre à retransmettre

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

- `src/orders.py`

## 10.58 ParamItemDelegate Class Reference

Inheritance diagram for ParamItemDelegate:



### Public Member Functions

- **ParamItemDelegate** (QObject \*parent=nullptr)
- QWidget \* **createEditor** (QWidget \*parent, const QStyleOptionViewItem &option, const QModelIndex &index) const
- void **setEditorData** (QWidget \*editor, const QModelIndex &index) const
- void **updateEditorGeometry** (QWidget \*editor, const QStyleOptionViewItem &option, const QModelIndex &index) const
- void **setModelData** (QWidget \*editor, QAbstractItemModel \*model, const QModelIndex &index) const

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

- src/editor/GUI/Tabs/objecteditor.h
- src/editor/GUI/Tabs/objecteditor.cpp

## 10.59 src.tools.Perf Class Reference

### Public Member Functions

- def **\_\_init\_\_** (self)
- def **tic** (self)
- def **toc** (self)
- def **show** (self)

### Public Attributes

- **num**
- **avg**
- **min**
- **max**
- **t**

### 10.59.1 Detailed Description

Calcule les performances d'un morceau de code

## 10.59.2 Member Function Documentation

### 10.59.2.1 show()

```
def src.tools.Perf.show (  
    self )
```

Affiche le rapport

### 10.59.2.2 tic()

```
def src.tools.Perf.tic (  
    self )
```

À lancer avant la fonction

### 10.59.2.3 toc()

```
def src.tools.Perf.toc (  
    self )
```

À lancer après la fonction

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

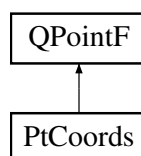
- src/tools.py

## 10.60 PtCoords Class Reference

The [PtCoords](#) class describe positions with virtual point coordinates.

```
#include <mappainter.h>
```

Inheritance diagram for PtCoords:



### Public Member Functions

- **PtCoords** (qreal x, qreal y)
- **PtCoords** (const QPointF &p)

### 10.60.1 Detailed Description

The [PtCoords](#) class describe positions with virtual point coordinates.

Theses coordinates describe each point relatively to the view. They correspond to a point in the image containing the entire map.

See also

[RlCoords](#), [ClCoords](#), [PxCoords](#)

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

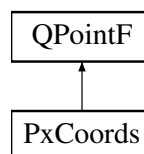
- `src/editor/Game/mappainter.h`

## 10.61 PxCoords Class Reference

The [PxCoords](#) class describe positions with real pixel coordinates.

```
#include <mappainter.h>
```

Inheritance diagram for PxCoords:



### Public Member Functions

- **PxCoords** (qreal x, qreal y)
- **PxCoords** (const QPointF &p)
- **PxCoords** (const QPoint &p)
- **PxCoords** (int x, int y)

### 10.61.1 Detailed Description

The [PxCoords](#) class describe positions with real pixel coordinates.

Theses coordinates describe the pixel position.

See also

[RlCoords](#), [ClCoords](#), [PtCoords](#)

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

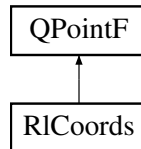
- `src/editor/Game/mappainter.h`

## 10.62 RCoords Class Reference

The [RCoords](#) class describe positions with relative coordinates.

```
#include <mappainter.h>
```

Inheritance diagram for RCoords:



### Public Member Functions

- **RCoords** (qreal x, qreal y)
- **RCoords** (const QPointF &p)

### 10.62.1 Detailed Description

The [RCoords](#) class describe positions with relative coordinates.

Theses coordinates have values in  $[0, 1]$ , for every point in the view.

See also

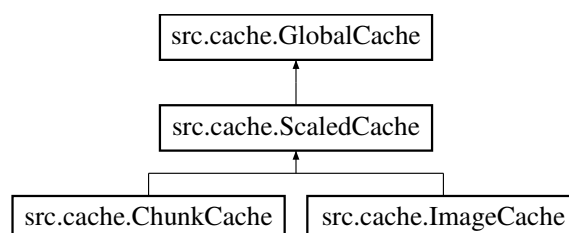
[CIRCoords](#) [PtCoords](#), [PxCoords](#)

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

- `src/editor/Game/mappainter.h`

## 10.63 src.cache.ScaledCache Class Reference

Inheritance diagram for src.cache.ScaledCache:





### Public Member Functions

- def **add\_scaled** (cls, elt, scale=1)
- def **remove** (cls, elt, scale=1)
- def **get\_elt** (cls, elt, scale=1)
- def **free\_cache** (cls)

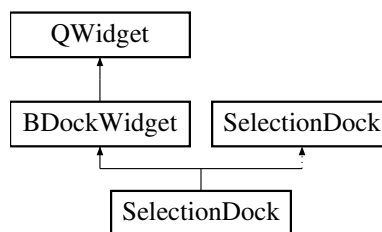
### Additional Inherited Members

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

- src/cache.py

## 10.64 SelectionDock Class Reference

Inheritance diagram for SelectionDock:



### Public Member Functions

- **SelectionDock** ([MapView](#) \*mv, QWidget \*parent=0)

### Additional Inherited Members

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

- src/editor/GUI/Tabs/selectiondock.h
- src/editor/GUI/Tabs/selectiondock.cpp

## 10.65 src.server.Server Class Reference

### Public Member Functions

- def **\_\_init\_\_** (self, path)
- def **\_\_del\_\_** (self)
- def **run** (self)
- def **main** (self)
- def **handleEvent** (self, emitter, event)

## Public Attributes

- **loop**
- **net**
- **world**
- **actions**
- **timer**
- **orderDispatcher**
- **events**
- **pause**
- **coEntities**

### 10.65.1 Detailed Description

Classe principale du processus serveur, concilie réseau, monde, actions et timer

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

- `src/server.py`

## 10.66 `src.network.ServerConnection` Class Reference

### Public Member Functions

- `def __init__(self, reader, writer, handle, parent)`
- `def run(self)`
- `def send(self, m)`
- `def end(self)`

### Public Attributes

- **reader**
- **writer**
- **handle**
- **entity**
- **server**

### 10.66.1 Detailed Description

This thread manages the communications with one particular client (one thread is created by client).

## 10.66.2 Member Function Documentation

### 10.66.2.1 run()

```
def src.network.ServerConnection.run (
    self )
```

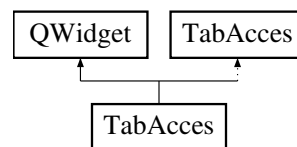
Wait for messages from the client and handle them immediately.

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

- src/network.py

## 10.67 TabAcces Class Reference

Inheritance diagram for TabAcces:



### Signals

- void **activated** (int i)

### Public Member Functions

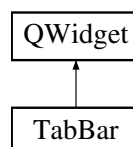
- **TabAcces** (int i, const QString &n, const QPixmap &p, QWidget \*parent=0)
- void **setActive** (bool a)

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

- src/editor/GUI/tabaccres.h
- src/editor/GUI/tabaccres.cpp

## 10.68 TabBar Class Reference

Inheritance diagram for TabBar:



## Public Slots

- void **setCurrentTab** (int t)

## Signals

- void **currentTabChanged** (int)

## Public Member Functions

- **TabBar** (QWidget \*parent=0)
- void **addTabAcces** (const QString &n, const QPixmap &p)
- int **currentTab** () const
- void **setTabsEnabled** (bool e)

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

- src/editor/GUI/tabbar.h
- src/editor/GUI/tabbar.cpp

## 10.69 src.tools.Timer Class Reference

### Public Member Functions

- def **\_\_init\_\_** (self, timeFunc=time)
- def **add** (self, time, func, args)
- def **run** (self)

### Public Attributes

- **dt**
- **step**
- **heap**
- **pause**
- **count**
- **time**

### 10.69.1 Detailed Description

Déclenche des appels différés de couroutine

## 10.69.2 Member Function Documentation

### 10.69.2.1 add()

```
def src.tools.Timer.add (
    self,
    time,
    func,
    args )
```

Inscrit l'appel de func avec les arguments args

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

- src/tools.py

## 10.70 src.utils.WalkableGraph Class Reference

### Public Member Functions

- def **\_\_init\_\_** (self, walkables)
- def **get\_neighbors** (self, index)
- def **dist** (self, u, v)
- def **get\_path** (self, source, dest)

### Public Attributes

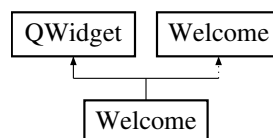
- **walkables**

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

- src/utils.py

## 10.71 Welcome Class Reference

Inheritance diagram for Welcome:



## Public Member Functions

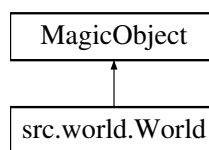
- **Welcome** (QWidget \*parent=0)

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

- src/editor/GUI/Tabs/welcome.h
- src/editor/GUI/Tabs/welcome.cpp

## 10.72 src.world.World Class Reference

Inheritance diagram for src.world.World:



## Public Member Functions

- `def __init__ (self)`

## Public Attributes

- **maps**
- **entities**
- **objects**

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

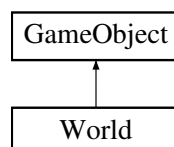
- src/world.py

## 10.73 World Class Reference

The [World](#) class.

```
#include <game.h>
```

Inheritance diagram for World:



## Public Member Functions

- **World** ([Game](#) \*g, [GameObject](#) \*parent)
- **ObjectListD** (m, M, ap,, s, [Map](#)) ObjectListD(o

## Public Attributes

- **O**
- **bject**
- **s**

## Additional Inherited Members

### 10.73.1 Detailed Description

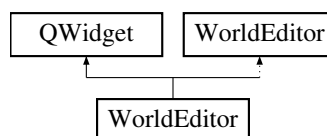
The [World](#) class.

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

- src/editor/Game/[game.h](#)
- src/editor/Game/game.cpp

## 10.74 WorldEditor Class Reference

Inheritance diagram for WorldEditor:



## Public Member Functions

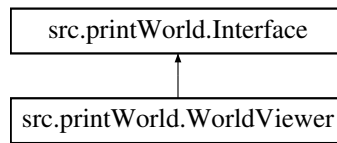
- **WorldEditor** (QWidget \*parent=0)
- void **setGame** ([Game](#) \*g)

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

- src/editor/GUI/Tabs/worldeditor.h
- src/editor/GUI/Tabs/worldeditor.cpp

## 10.75 src.printWorld.WorldViewer Class Reference

Inheritance diagram for src.printWorld.WorldViewer:



### Public Member Functions

- def `__init__` (self, w)
- def `get_event` (self)
- def `end` (self)
- def `update` (self)
- def `render` (self)
- def `move` (self, dx, dy)
- def `move_char` (self, ident, end\_pos)
- def `propagate_trigger` (self, event)

### Public Attributes

- `screen_size`
- `current_map`
- `main_char`
- `characters`

### 10.75.1 Constructor & Destructor Documentation

#### 10.75.1.1 `__init__()`

```
def src.printWorld.WorldViewer.__init__ (
    self,
    w )
```

Initialize the pygame Interface

### 10.75.2 Member Function Documentation

#### 10.75.2.1 `end()`

```
def src.printWorld.WorldViewer.end (
    self )
```

End the display



### 10.75.2.2 get\_event()

```
def src.printWorld.WorldViewer.get_event (
    self )
```

Return current pygame events stack

### 10.75.2.3 move()

```
def src.printWorld.WorldViewer.move (
    self,
    dx,
    dy )
```

Move the current map by dx,dy, return the rectangle of modifications

### 10.75.2.4 move\_char()

```
def src.printWorld.WorldViewer.move_char (
    self,
    ident,
    end_pos )
```

Move the entitie ident to cell end\_pos

### 10.75.2.5 propagate\_trigger()

```
def src.printWorld.WorldViewer.propagate_trigger (
    self,
    event )
```

Propagate event to next objects in the arborescence, here its the current map

### 10.75.2.6 render()

```
def src.printWorld.WorldViewer.render (
    self )
```

Return the new image to show on screen

### 10.75.2.7 update()

```
def src.printWorld.WorldViewer.update (
    self )
```

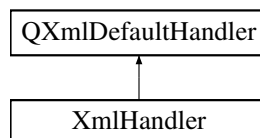
Update the view

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

- [src/printWorld.py](#)

## 10.76 XmlHandler Class Reference

Inheritance diagram for XmlHandler:



### Public Member Functions

- **XmlHandler** ([Game](#) \*g)
- bool **startElement** (const QString &, const QString &localName, const QString &, const QXmlAttributes &atts)
- bool **endElement** (const QString &, const QString &localName, const QString &)

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

- [src/editor/Game/xmlhandler.h](#)
- [src/editor/Game/xmlhandler.cpp](#)

# Chapter 11

## File Documentation

### 11.1 src/editor/Game/game.h File Reference

Definition of the [Game](#) and [World](#) classes.

```
#include "object.h"
#include "map.h"
```

#### Classes

- class [World](#)  
*The [World](#) class.*
- class [Game](#)  
*The [Game](#) class gather the differents parts needed to describe a game.*

#### 11.1.1 Detailed Description

Definition of the [Game](#) and [World](#) classes.

### 11.2 src/editor/Game/map.h File Reference

Definition of the [Map](#), [Cell](#) and [CellType](#) classes.

```
#include "object.h"
```

#### Classes

- class [CellType](#)  
*The [CellType](#) class.*
- class [Cell](#)  
*The [Cell](#) class.*
- class [Map](#)  
*The [Map](#) class.*

## Macros

- `#define forCells(i) int nbCell = width()*height(); for(int i(0); i<nbCell; ++i)`

### 11.2.1 Detailed Description

Definition of the [Map](#), [Cell](#) and [CellType](#) classes.

### 11.2.2 Macro Definition Documentation

#### 11.2.2.1 [forCells](#)

```
#define forCells(
    i ) int nbCell = width()*height(); for(int i(0); i<nbCell; ++i)
```

Usefull macro to set up a for on the cells

## 11.3 [src/editor/Game/mappainter.h](#) File Reference

Definition of the [MapPainter](#) class and other related classes to render [maps](#).

```
#include "map.h"
#include "GUI/options.h"
```

## Classes

- class [RlCoords](#)  
*The [RlCoords](#) class describe positions with relative coordinates.*
- class [ClCoords](#)  
*The [ClCoords](#) class describe positions with cell coordinates.*
- class [PtCoords](#)  
*The [PtCoords](#) class describe positions with virtual point coordinates.*
- class [PxCoords](#)  
*The [PxCoords](#) class describe positions with real pixel coordinates.*
- class [MapPainter](#)  
*The [MapPainter](#) class that can paint a [Map](#) using a [QPainter](#).*

## Macros

- `#define MINMAX(a, x, b) std::min(std::max(a,x),b)`

## Functions

- [MapPainter::Element operator|](#) ([MapPainter::Element](#) a, [MapPainter::Element](#) b)  
*The operator | is the flag OR operation.*
- [MapPainter::Element operator &](#) ([MapPainter::Element](#) a, [MapPainter::Element](#) b)  
*The operator & is the flag AND operation.*
- [MapPainter::Element operator^](#) ([MapPainter::Element](#) a, [MapPainter::Element](#) b)  
*The operator ^ is the flag substraction operation.*

### 11.3.1 Detailed Description

Definition of the [MapPainter](#) class and other related classes to render [maps](#).

This file defines four types of coordinates : [RiCoords](#), [CiCoords](#), [PtCoords](#) and [PxCoords](#). They all inherit from QPointF, and give a static type checking for the consistency of the coordinates which are used.

#### Author

Baptiste Pauget

### 11.3.2 Function Documentation

#### 11.3.2.1 operator^()

```
MapPainter::Element operator^ (
    MapPainter::Element a,
    MapPainter::Element b ) [inline]
```

The operator ^ is the flag substraction operation.

#### Warning

This is not a XOR operation, it corresponds to a&!b

## 11.4 src/editor/Game/mapslistmodel.h File Reference

Definition of Model/View presentation classes.

```
#include <QAbstractListModel>
#include <QAbstractTableModel>
#include "game.h"
#include "mappainter.h"
```

## Classes

- class [MapsListModel](#)  
*The [MapsListModel](#) class provides a presentation class for the Qt Model-View framework.*
- class [CellTypeListModel](#)  
*The [CellTypeListModel](#) class.*
- class [ObjectParamTableModel](#)
- class [ObjectFlagTableModel](#)

### 11.4.1 Detailed Description

Definition of Model/View presentation classes.

## 11.5 src/editor/Game/object.h File Reference

Definition of the base class [GameObject](#), and some inherited classes.

```
#include <QtCore>
#include <QtGui>
#include <assert.h>
```

### Classes

- class [GameObject](#)  
The [GameObject](#) class is the base class for every part of games.
- class [Image](#)  
The [Image](#) class stores an external file in a QImage, and gives each image ressources a unique identifier.
- class [Object](#)  
The [Object](#) class.

### Macros

- #define [ObjectsMapC](#)(name, names, Type, Types, pref, arg)
- #define [ObjectsMap](#)(pref, ini, Ini, body, sg, pl) [ObjectsMapC](#)(ini##body##sg, ini##body##pl, Ini##body##sg, Ini##body##pl, pref,ini)
- #define [ObjectListDef](#)(Objects, Type) private: QMap<int, Type\*> a##Objects; public:
- #define [ObjectListAdd](#)([Object](#), Objects, Type) void add##[Object](#)(Type\* new##[Object](#)){a##Objects[new##[Object](#)->ident()] = new##[Object](#); touch();}
- #define [ObjectListTake](#)([Object](#), Objects, Type) Type\* take##[Object](#)(int id){touch(); return a##Objects.<← take(id);}
- #define [ObjectListGetter](#)(object, Objects, Type) inline Type\* object(int id) const{return a##Objects.value(id, nullptr);}
- #define [ObjectListValues](#)(objects, Objects, Type) inline QList<Type\*> objects() const{return a##Objects.<← values();}
- #define [ObjectListGetters](#)(object, [Object](#), objects, Objects, Type) [ObjectListGetter](#)(object,Objects, Type) [ObjectListValues](#)(objects,Objects, Type)
- #define [ObjectListModifiers](#)([Object](#), Objects, Type) [ObjectListAdd](#)([Object](#),Objects, Type) [ObjectList](#)<← Take([Object](#), Objects, Type)
- #define [ObjectList](#)(object, [Object](#), objects, Objects, Type) [ObjectListDef](#)(Objects,Type) [ObjectList](#)<← Getters(object,[Object](#),objects,Objects,Type) [ObjectListModifiers](#)([Object](#), Objects, Type)
- #define [ObjectListD](#)(init, Init, body, sg, pl, Type) [ObjectList](#)(init##body##sg,Init##body##sg,init##body##pl,Init##body##pl,Type)
- #define [C](#)(Macro, init, Init, body, ...) Macro(init##body, Init##body, ##\_\_VA\_ARGS\_\_)
- #define [C0](#)(Macro, init, Init, body) Macro(init##body, Init##body)
- #define [C1](#)(Macro, init, Init, body, arg) Macro(init##body, Init##body, arg)
- #define [SetFlag](#)(flag, value) aFlags[#flag] = value
- #define [FlagGetter](#)(flag, [Flag](#)) inline bool is##[Flag](#)() const{return aFlags[#flag];}
- #define [FlagSetter](#)(flag, [Flag](#)) inline void set##[Flag](#)(bool flag){[SetFlag](#)(flag,flag); touch();}
- #define [Flag](#)(flag, [Flag](#)) [FlagGetter](#)(flag, [Flag](#)) [FlagSetter](#)(flag, [Flag](#))

- `#define SetParam(param, value) aParams[#param] = value`
- `#define ParamGetter(param) inline int param() const{return aParams[#param];}`
- `#define ParamSetter(param, Param) inline void set##Param(int param##Value){SetParam(param,param##Value); touch();}`
- `#define Param(param, Param) ParamGetter(param) ParamSetter(param, Param)`
- `#define AttrGetter(attr, Attr, Type) inline Type* attr() const{return a##Attr;}`
- `#define AttrFree(Attr) if(a##Attr) a##Attr->removeReference();`
- `#define AttrLink(Attr) if(a##Attr) a##Attr->addReference();`
- `#define AttrSetter(attr, Attr, Type) inline void set##Attr(Type* new##Attr){AttrFree(Attr); a##Attr = new##Attr; AttrLink(Attr); touch();}`
- `#define AttrDef(Attr, Type) private: Type* a##Attr = nullptr; public:`
- `#define Attr(attr, Attr, Type) AttrDef(Attr, Type) AttrGetter(attr,Attr,Type) AttrSetter(attr, Attr, Type)`
- `#define AttrT(type, Type) Attr(type, Type, Type)`

### 11.5.1 Detailed Description

Definition of the base class [GameObject](#), and some inherited classes.

The objects structure

Objects destructors

The Macro System

To add conveniently attributes and flags to [GameObject](#) subclassed objects, a set of macro is provided.

Name conventions

For a attribute named `attr`, the following conventions are observed :

- `attr()` is the getter method
- `setAttr()` is the setter method
- `aAttr` is the name of the attribut (if any)

A specific convention is applied for flags (boolean attributes) :

- `isAttr()` is the getter method

Macros

To define a new attribute, a global macro can be used in the class declaration. The provided basic implementations keep the object edition synchronization.

If a cleverer process is needed, custom getter or setter can be implemented, and the getter and setter macros can be used separately to define the obvious methods

**Provided macros**

Attribute Type	Complete declaration	Getter	Setter
Flags (bool)	<a href="#">Flag</a>	<a href="#">FlagGetter</a>	<a href="#">FlagSetter</a>
Parameters (int)	<a href="#">Param</a>	<a href="#">ParamGetter</a>	<a href="#">ParamSetter</a>
<a href="#">GameObject</a> based Attributes	<a href="#">Attr</a>	<a href="#">AttrGetter</a>	<a href="#">AttrSetter</a>

## The case of attributes

An additionnal [AttrT](#) macro is provided, that deduce a default name from the type.

### Name tools

To make the definition easier and avoid the name repetition that is introduced by the name convention, a [C](#) macro is provided to construct the names with lower and upper initial letter from theses letter and the end of the name.

### Author

Baptiste Pauget

## 11.5.2 Macro Definition Documentation

### 11.5.2.1 Attr

```
#define Attr(
    attr,
    Attr,
    Type ) AttrDef(Attr, Type) AttrGetter(attr,Attr,Type) AttrSetter(attr, Attr,
Type)
```

The `Attr` macro defines a new `<aAttr>` named attribute of type `Type`, with its generic getter and setter methods.

With respect to the [name convention](#), this macro needs the parameter's name with lower and upper initial letter case.

### Example

```
Attr(parent,Parent, GameObject)
-->
private:
    GameObject *aParent;
public:
    inline GameObject* parent() const{return aParent;}
    inline void setParent(GameObject* parentObject){aParent = parentObject; touch();}
```

### See also

[AttrT](#), [AttrDef](#), [AttrSetter](#), [AttrGetter](#), [C](#)



### 11.5.2.2 AttrDef

```
#define AttrDef(  
    Attr,  
    Type ) private:  Type* a##Attr = nullptr; public:
```

The AttrDef macro defines a private attribute name <aAttr>.

#### Note

To avoid redefinition error, no attribute or method name <aAttr > must exist.

#### Warning

This macro is designed to be used in a public part of the class. Please note that inserting this macro in a private or protected part will change the visibility of the next declaration to public.

#### Example

```
Attr(Parent,GameObject)  
-->  
private:  
    GameObject *aParent;  
public:
```

#### See also

[Attr](#)

### 11.5.2.3 AttrGetter

```
#define AttrGetter(  
    attr,  
    Attr,  
    Type ) inline Type* attr() const{return a##Attr;}
```

The AttrGetter macro defines a generic getter method for the attribute named attr of type Type.

With respect to the [name convention](#), this macro needs the attribute's name with lower and upper initial letter case.

#### Example

```
AttrGetter(parent,Parent,GameObject)  
--> inline GameObject* parent() const{return aParent;}
```

#### See also

[Attr](#), [AttrSetter](#), [C](#)

#### 11.5.2.4 AttrSetter

```
#define AttrSetter(
    attr,
    Attr,
    Type ) inline void set##Attr(Type* new##Attr) {AttrFree(Attr); a##Attr = new##Attr;
AttrLink(Attr); touch();}
```

The AttrSetter macro defines a generic setter method for the attribute named `attr` of type `Type`.

With respect to the [name convention](#), this macro needs the attribute's name with lower and upper initial letter case.

#### Example

```
AttrSetter(parent, Parent, GameObject)
--> inline void setParent(GameObject* &parentObject) {aParent = parentObject; touch();}
```

#### See also

[Attr](#), [AttrGetter](#), [C](#)

#### 11.5.2.5 AttrT

```
#define AttrT(
    type,
    Type ) Attr(type, Type, Type)
```

The AttrT macro defines a new attribute of type `Type`, named after the type name, with its generic getter and setter methods.

With respect to the [name convention](#), this macro needs the parameter's type with lower and upper initial letter case.

#### Example

```
AttrT(cellType, CellType)
-->
private:
    CellType *aCellType;
public:
    inline CellType* cellType() const{return aCellType;}
    inline void setCellType(CellType* cellTypeObject){aCellType = cellTypeObject; touch();}
```

#### See also

[Attr](#), [AttrDef](#), [AttrSetter](#), [AttrGetter](#), [C](#)

### 11.5.2.6 C

```
#define C(
    Macro,
    init,
    Init,
    body,
    ... ) Macro(init##body, Init##body, ##__VA_ARGS__)
```

The C macro calls the Macro argument with argument tokens formed by the concatenation of init and body, and Init and body.

This enables to call a macro with the same argument, with the initial letter in lower and upper case.

A custom number of arguments can be added after the body one.

#### Note

This use of variadic arguments follow the [gcc specification](#), but can be not supported by some compilers.

As some IDE does not fully support variadic macro expansion, the [C0](#) and [C1](#) macros can be used to avoid some inconvenience due to uncomplete code undersanding.

#### Example

```
C(Param,w,W,idth)
--> Param(width, Width)

C(Attr, p,P,arent, GameObject)
--> Attr(parent, Parent, GameObject)
```

#### See also

[object.h](#)

### 11.5.2.7 C0

```
#define C0(
    Macro,
    init,
    Init,
    body ) Macro(init##body, Init##body)
```

The C0 macro is equivalent to the [C](#) macro, with no additional argument.

This macro is provided to avoid the use of variadic arguments that are currently not totally supported by some IDE.

#### Example

```
C(Flag, v,V,isible)
--> Flag(visible, Visible)
```

#### See also

[C1](#)

### 11.5.2.8 C1

```
#define C1(  
    Macro,  
    init,  
    Init,  
    body,  
    arg ) Macro(init##body, Init##body, arg)
```

The C1 macro is equivalent to the [C](#) macro, with one additional argument.

This macro is provided to avoid the use of variadic arguments that are currently not totally supported by some IDE.

#### Example

```
C(Attr, p,P,arent, GameObject)  
--> Attr(parent, Parent, GameObject)
```

See also

[C0](#)

### 11.5.2.9 Flag

```
#define Flag(  
    flag,  
    Flag ) FlagGetter(flag, Flag) FlagSetter(flag, Flag)
```

The Flag macro defines generic getter and setter methods for the flag named `flag`.

With respect to the [name convention](#), this macro needs the flag's name with lower and upper initial letter case.

#### Example

```
Flag(visible,Visible)  
--> inline bool isVisible() const{return aFlags["visible"];}  
    inline void setVisible(bool visible){aFlags["visible"] = visible; touch() }
```

See also

[FlagGetter](#), [FlagSetter](#), [C](#)

### 11.5.2.10 FlagGetter

```
#define FlagGetter(  
    flag,  
    Flag ) inline bool is##Flag() const{return aFlags[#flag];}
```

The FlagGetter macro defines a generic getter method for the flag named `flag`.

With respect to the [name convention](#), this macro needs the flag's name with lower and upper initial letter case.

#### Warning

The default getter method does not check whether the `flag` named flag really exist. To avoid runtime access error, it is strongly advice to initialize the flag in the object's constructor, using the setter method or the [SetFlag](#) macro.

#### Example

```
FlagGetter(visible,Visible)  
--> inline bool isVisible() const{return aFlags["visible"];}
```

See also

[Flag](#), [FlagSetter](#), [C](#)

### 11.5.2.11 FlagSetter

```
#define FlagSetter(
    flag,
    Flag ) inline void set##Flag(bool flag){SetFlag(flag,flag); touch();}
```

The FlagSetter macro defines a generic setter method for the flag named `flag`.

With respect to the [name convention](#), this macro needs the flag's name with lower and upper initial letter case.

#### Example

```
FlagSetter(visible,Visible)
--> inline void setVisible(bool visible){aFlags["visible"] = visible; touch()}
```

#### See also

[Flag](#), [FlagGetter](#), [C](#)

### 11.5.2.12 ObjectsMap

```
#define ObjectsMap(
    pref,
    ini,
    Ini,
    body,
    sg,
    pl ) ObjectsMapC(ini##body##sg, ini##body##pl, Ini##body##sg, Ini##body##pl,
pref,ini)
```

#### Deprecated

### 11.5.2.13 ObjectsMapC

```
#define ObjectsMapC(
    name,
    names,
    Type,
    Types,
    pref,
    arg )
```

#### Value:

```
private: \
    QMap<int, Type*> pref##Types; \
public: \
    void add##Type(Type* arg){pref##Types[arg->ident()] = arg; touch();} \
    void remove##Type(Type* arg){if(pref##Types.contains(arg->ident()))pref##Types.remove(arg->ident()); \
    touch();} \
    inline Type* name(int id) const{return pref##Types.value(id, nullptr);} \
    inline QList<Type*> names() const{return pref##Types.values();}
```

#### Deprecated

#### 11.5.2.14 Param

```
#define Param(  
    param,  
    Param ) ParamGetter(param) ParamSetter(param, Param)
```

The Param macro defines generic getter and setter methods for the parameter named `param`.

With respect to the [name convention](#), this macro needs the parameter's name with lower and upper initial letter case.

#### Example

```
Param(width,Width)  
--> inline int width() const{return aParams["width"];}  
    inline void setWidth(int widthValue){aParams["width"] = widthValue; touch();}
```

#### See also

[ParamGetter](#), [ParamSetter](#), [C](#)

#### 11.5.2.15 ParamGetter

```
#define ParamGetter(  
    param ) inline int param() const{return aParams[#param];}
```

The ParamGetter macro defines a generic getter method for the parameter named `param`.

#### Warning

The default getter method does not check whether the `param` named parameter really exist. To avoid runtime access error, it is strongly advice to initialize the parameter in the object's constructor, using the setter method or the [SetParam](#) macro.

#### Example

```
ParamGetter(width)  
--> inline int width() const{return aParams["width"];}
```

#### See also

[Param](#), [ParamSetter](#)

### 11.5.2.16 ParamSetter

```
#define ParamSetter(  
    param,  
    Param ) inline void set##Param(int param##Value){SetParam(param,param##Value);  
touch();}
```

The ParamSetter macro defines a generic setter method for the parameter named `param`.

With respect to the [name convention](#), this macro needs the parameter's name with lower and upper initial letter case.

#### Example

```
ParamSetter(width,Width)  
--> inline void setWidth(bool widthValue){aParams["width"] = widthValue; touch();}
```

#### See also

[Param](#), [ParamGetter](#), [C](#)

### 11.5.2.17 SetFlag

```
#define SetFlag(  
    flag,  
    value ) aFlags[#flag] = value
```

Convenient macro to set a flag directly.

This is usefull in custom setters, to avoid call loops.

#### Warning

The [touch](#) function isn't called. After this macro use, the [object](#) is no longer synchronised.

#### Example

```
SetFlag(visible, false)  
--> aFlags["visible"] = false
```

#### See also

[Flag](#), [FlagSetter](#), [object.h](#)

### 11.5.2.18 SetParam

```
#define SetParam(  
    param,  
    value ) aParams[#param] = value
```

Convenient macro to set a param directly.

This is usefull in custom setters, to avoid call loops.

#### Warning

The [touch](#) function isn't called. After this macro use, the [object](#) is no longer synchronised.

#### Example

```
SetParam(width, 42)  
--> aParams["width"] = 42
```

#### See also

[Param](#), [ParamSetter](#), [object.h](#)

## 11.6 src/editor/Game/xmlhandler.h File Reference

Definition og the [XmlHandler](#) class and other related classes to read XML game's files.

```
#include <QtXml>  
#include "game.h"
```

### Classes

- class [XmlHandler](#)

### Typedefs

- typedef std::pair< QString, FileContent > **Asso**

### Enumerations

- enum **FileContent** {  
    **FCUnknown**, **FCGame**, **FCRessources**, **FCWorld**,  
    **FCMap**, **FCEntity**, **FCObject** }

### Functions

- const QMap< QString, FileContent > **overHead** ({ Asso("Game", FCGame), Asso("Ressources", FC↵  
Ressources), Asso("World", FCWorld), Asso("Map", FCMap), Asso("Entity", FCEntity), Asso("Object", F↵  
CObject) })



### 11.6.1 Detailed Description

Definition of the [XmlHandler](#) class and other related classes to read XML game's files.

## 11.7 src/editor/GUI/options.h File Reference

Definition of the [Options](#) class, and the constants that are used in this class.

```
#include <QSettings>
#include <QDir>
#include <QSize>
#include <QPoint>
#include <QColor>
```

### Classes

- struct [Options](#)

*The [Options](#) class provides session-independent options and preferences.*

### Macros

- #define [WIN](#) "Window"
- #define [DIR](#) "Directories"
- #define [MAP](#) "MapsEditor"
- #define [DefaultF](#)(group, opt, val) defaultValues[group][opt] = QPair<QVariant, bool>(val, false)
- #define [Default](#)(group, opt, val) defaultValues[group][opt] = QPair<QVariant, bool>(val, true)

### Variables

- const QString **ADAPT** = "Adjustable"
- const QString **VAL** = "Value"

### 11.7.1 Detailed Description

Definition of the [Options](#) class, and the constants that are used in this class.

The headers of types which are used in the application must be include here. See [Options](#) for details.

### Author

Baptiste Pauget

## 11.7.2 Macro Definition Documentation

### 11.7.2.1 Default

```
#define Default(  
    group,  
    opt,  
    val ) defaultValues[group][opt] = QPair<QVariant, bool>(val, true)
```

This macro defines a new adaptati option identified by its group and name.

### 11.7.2.2 DefaultF

```
#define DefaultF(  
    group,  
    opt,  
    val ) defaultValues[group][opt] = QPair<QVariant, bool>(val, false)
```

This macro defines a new unadaptati option identified by its group and name.

### 11.7.2.3 DIR

```
#define DIR "Directories"
```

Group of paths options.

### 11.7.2.4 MAP

```
#define MAP "MapsEditor"
```

Group of [MapsEditor](#) related options.

### 11.7.2.5 WIN

```
#define WIN "Window"
```

Group of window related options.

## 11.8 src/editor/GUI/Tabs/bcolor.h File Reference

Definition of the [BColor](#) class.

```
#include <QtWidgets>
```

## Classes

- class [BColor](#)

*The [BColor](#) class is a simple frame that offers color selection.*

### 11.8.1 Detailed Description

Definition of the [BColor](#) class.

#### Author

Baptiste Pauget

## 11.9 src/editor/GUI/Tabs/Docks/bdock.h File Reference

Definition of the [BDock](#) class.

```
#include <QtWidgets>
#include "intertie.h"
#include "bdockwidget.h"
```

## Classes

- class [BDock](#)

*The [BDock](#) class is the container for widget to display in a [BDocksZone](#).*

### 11.9.1 Detailed Description

Definition of the [BDock](#) class.



# Index

- `__del__`
    - `src::client::Client`, [43](#)
  - `__init__`
    - `src::network::NetworkServer`, [76](#)
    - `src::printWorld::WorldViewer`, [98](#)
- `add`
  - `src::tools::Timer`, [95](#)
- `askEntity`
  - `src::network::NetworkClient`, [74](#)
- `Attr`
  - `object.h`, [106](#)
- `AttrDef`
  - `object.h`, [106](#)
- `AttrGetter`
  - `object.h`, [107](#)
- `AttrSetter`
  - `object.h`, [107](#)
- `AttrT`
  - `object.h`, [108](#)
- `BColor`, [27](#)
  - `BColor`, [27](#)
  - `color`, [28](#), [30](#)
  - `colorChanged`, [28](#)
  - `name`, [28](#), [30](#)
  - `nameChanged`, [28](#)
  - `setColor`, [28](#)
  - `setColorQuiet`, [29](#)
  - `setName`, [29](#)
  - `setNameQuiet`, [29](#)
- `BDock`, [30](#)
- `BDockWidget`, [32](#)
- `BDocksZone`, [31](#)
- `BLayout`, [34](#)
- `BinaryStateMachine`, [33](#)
- `C`
  - `object.h`, [108](#)
- `C0`
  - `object.h`, [109](#)
- `C1`
  - `object.h`, [109](#)
- `Cell`, [35](#)
- `CellDock`, [36](#)
- `CellType`, [37](#)
- `CellTypeListModel`, [37](#)
- `CellTypesDock`, [38](#)
- `CICoords`, [41](#)
- `color`
  - `BColor`, [28](#), [30](#)
- `colorChanged`
  - `BColor`, [28](#)
- `contextEval`
  - `src::world::BaseObject`, [26](#)
- `cooToPt`
  - `MapPainter`, [62](#)
- `cooToPxl`
  - `MapPainter`, [62](#)
- `create`
  - `src::world::ObjectType`, [81](#)
- `DIR`
  - `options.h`, [116](#)
- `Default`
  - `options.h`, [116](#)
- `DefaultF`
  - `options.h`, [116](#)
- `Editor`, [44](#)
- `Element`
  - `MapPainter`, [61](#)
- `end`
  - `src::printWorld::WorldViewer`, [98](#)
- `fill`
  - `src::world::Map`, [59](#)
- `Flag`
  - `object.h`, [110](#)
- `FlagGetter`
  - `object.h`, [110](#)
- `FlagSetter`
  - `object.h`, [110](#)
- `flags`
  - `GameObject`, [48](#)
- `forCells`
  - `map.h`, [102](#)
- `fromBytes`
  - `src::orders::Order`, [85](#)
- `Game`, [45](#)
  - `newIdent`, [46](#)
- `GameObject`, [46](#)
  - `flags`, [48](#)
  - `GameObject`, [48](#)
  - `getFlag`, [48](#)
  - `getParam`, [48](#)
  - `hasFlag`, [49](#)
  - `hasParam`, [49](#)
  - `ident`, [49](#)

- init, [49](#)
- isValid, [50](#)
- lastChildrenEdition, [50](#)
- lastEdition, [50](#)
- lastInternalEdition, [50](#)
- params, [51](#)
- setFlag, [51](#)
- setParam, [51](#)
- touch, [51](#)
- get\_event
  - src::printWorld::WorldViewer, [98](#)
- get\_size
  - src::parsing::map\_parser, [21](#)
- get\_src\_dir
  - src::path::Archi, [23](#)
- get\_src\_file
  - src::path::Archi, [23](#)
- get\_static\_dir
  - src::path::Archi, [24](#)
- get\_static\_file
  - src::path::Archi, [24](#)
- get\_xml\_dir
  - src::path::Archi, [24](#)
- get\_xml\_file
  - src::path::Archi, [24](#)
- getFlag
  - GameObject, [48](#)
- getParam
  - GameObject, [48](#)
- hasFlag
  - GameObject, [49](#)
- hasHighlightedCell
  - MapPainter, [62](#)
- hasParam
  - GameObject, [49](#)
- highlightedCell
  - MapPainter, [62](#)
- ident
  - GameObject, [49](#)
- Image, [53](#)
  - isValid, [53](#)
- indToPt
  - MapPainter, [62](#)
- init
  - GameObject, [49](#)
- Intertie, [56](#)
- isAdjustable
  - Options, [83](#)
- isCell
  - MapPainter, [63](#)
- isValid
  - GameObject, [50](#)
  - Image, [53](#)
- lastChildrenEdition
  - GameObject, [50](#)
- lastEdition
  - GameObject, [50](#)
- lastInternalEdition
  - GameObject, [50](#)
- list\_files
  - src::path::Archi, [24](#)
- load
  - Options, [83](#)
  - src::orders::Order, [85](#)
  - src::world::BaseObject, [26](#)
- MAP
  - options.h, [116](#)
- Map, [57](#)
- map.h
  - forCells, [102](#)
- map\_parser
  - src::parsing::map\_parser, [21](#)
- MapDock, [59](#)
- MapPainter, [59](#)
  - cooToPt, [62](#)
  - cooToPxl, [62](#)
  - Element, [61](#)
  - hasHighlightedCell, [62](#)
  - highlightedCell, [62](#)
  - indToPt, [62](#)
  - isCell, [63](#)
  - MapPainter, [61](#)
  - mapSizeChanged, [63](#)
  - move, [63](#)
  - paint, [63](#)
  - preSelectedCellColor, [63](#)
  - ptToCoo, [64](#)
  - ptToPxl, [64](#)
  - pxlToCoo, [64](#)
  - pxlToPt, [64](#)
  - render, [64](#)
  - resize, [64](#), [65](#)
  - scale, [65](#)
  - selectedCellColor, [65](#)
  - setHighlightedCell, [65](#)
  - setMap, [66](#)
  - setPaintedElement, [66](#)
  - setPaintedElements, [66](#)
  - setPreSelectedCellColor, [66](#)
  - setScale, [66](#)
  - setScaleDomain, [67](#)
  - setSelectedCellColor, [67](#)
  - setViewCenter, [67](#)
  - setViewCenterQuiet, [68](#)
  - size, [68](#)
  - viewCenter, [68](#)
  - viewCenterChanged, [68](#)
  - virtualSize, [68](#)
  - zoom, [69](#)
- mapSizeChanged
  - MapPainter, [63](#)
- MapView, [72](#)
  - SelectionMode, [73](#)
- mappainter.h

- operator<sup>^</sup>, 103
- MapsEditor, 69
- MapsListModel, 70
- move
  - MapPainter, 63
  - src::printWorld::WorldViewer, 99
- move\_char
  - src::printWorld::WorldViewer, 99
- name
  - BColor, 28, 30
- nameChanged
  - BColor, 28
- NewGame, 78
- newIdent
  - Game, 46
- Object, 78
- object.h
  - Attr, 106
  - AttrDef, 106
  - AttrGetter, 107
  - AttrSetter, 107
  - AttrT, 108
  - C, 108
  - C0, 109
  - C1, 109
  - Flag, 110
  - FlagGetter, 110
  - FlagSetter, 110
  - ObjectsMap, 111
  - ObjectsMapC, 111
  - Param, 111
  - ParamGetter, 112
  - ParamSetter, 112
  - SetFlag, 113
  - SetParam, 113
- ObjectEditor, 79
- ObjectFlagTableModel, 79
- ObjectParamTableModel, 80
- ObjectsMap
  - object.h, 111
- ObjectsMapC
  - object.h, 111
- operator<sup>^</sup>
  - mappainter.h, 103
- Options, 81
  - isAdjustable, 83
  - load, 83
  - options, 83
  - reinitialise, 83
  - save, 83
  - setAdjustable, 84
  - setDefault, 84
- options
  - Options, 83
- options.h
  - DIR, 116
  - Default, 116
  - DefaultF, 116
  - MAP, 116
  - WIN, 116
- paint
  - MapPainter, 63
- Param
  - object.h, 111
- ParamGetter
  - object.h, 112
- ParamItemDelegate, 87
- ParamSetter
  - object.h, 112
- params
  - GameObject, 51
- parse\_cell
  - src::parsing::map\_parser, 21
- preSelectedCellColor
  - MapPainter, 63
- propagate\_trigger
  - src::printWorld::WorldViewer, 99
- PtCoords, 88
- ptToCoo
  - MapPainter, 64
- ptToPxl
  - MapPainter, 64
- PxCoords, 89
- pxlToCoo
  - MapPainter, 64
- pxlToPt
  - MapPainter, 64
- reinitialise
  - Options, 83
- render
  - MapPainter, 64
  - src::printWorld::WorldViewer, 99
- resize
  - MapPainter, 64, 65
- RICoords, 90
- run
  - src::network::NetworkClient, 74
  - src::network::NetworkServer, 76
  - src::network::ServerConnection, 93
- save
  - Options, 83
- scale
  - MapPainter, 65
- selectedCellColor
  - MapPainter, 65
- SelectionDock, 91
- SelectionMode
  - MapView, 73
- sendEvent
  - src::network::NetworkClient, 74
- sendOrder
  - src::network::NetworkServer, 76
- setAdjustable

- Options, 84
- setColor
  - BColor, 28
- setColorQuiet
  - BColor, 29
- setDefault
  - Options, 84
- SetFlag
  - object.h, 113
- setFlag
  - GameObject, 51
- setHighlightedCell
  - MapPainter, 65
- setMap
  - MapPainter, 66
- setName
  - BColor, 29
- setNameQuiet
  - BColor, 29
- setPaintedElement
  - MapPainter, 66
- setPaintedElements
  - MapPainter, 66
- SetParam
  - object.h, 113
- setParam
  - GameObject, 51
- setPreSelectedCellColor
  - MapPainter, 66
- setScale
  - MapPainter, 66
- setScaleDomain
  - MapPainter, 67
- setSelectedCellColor
  - MapPainter, 67
- setViewCenter
  - MapPainter, 67
- setViewCenterQuiet
  - MapPainter, 68
- show
  - src::tools::Perf, 88
- size
  - MapPainter, 68
- src.backgroundLayer.BackgroundLayer, 25
- src.cache.ChunkCache, 40
- src.cache.GlobalCache, 52
- src.cache.ImageCache, 54
- src.cache.ScaledCache, 90
- src.character.Character, 39
- src.chunk.Chunk, 39
- src.client.Client, 42
- src.cursescli.CellViewer, 38
- src.cursescli.Curses, 43
- src.cursescli.MapViewer, 71
- src.interactions.Interaction, 54
- src.interface.Interface, 55
- src.layer.Layer, 57
- src.map.MapViewer, 71
- src.network.NetworkClient, 74
- src.network.NetworkServer, 76
- src.network.ServerConnection, 92
- src.networkudp.NetworkClient, 75
- src.networkudp.NetworkServer, 77
- src.orders.Order, 85
- src.orders.OrderDispatcher, 86
- src.parsing.map\_parser, 21
- src.path.Archi, 23
- src.printWorld.Interface, 55
- src.printWorld.WorldViewer, 98
- src.pygamecli.Client, 41
- src.server.Server, 91
- src.tools.Perf, 87
- src.tools.Timer, 94
- src.utils.WalkableGraph, 95
- src.world.BaseObject, 26
- src.world.Cell, 35
- src.world.Entity, 44
- src.world.Map, 58
- src.world.ObjectType, 80
- src.world.World, 96
- src/editor/GUI/Tabs/Docks/bdock.h, 117
- src/editor/GUI/Tabs/bcolor.h, 116
- src/editor/GUI/options.h, 115
- src/editor/Game/game.h, 101
- src/editor/Game/map.h, 101
- src/editor/Game/mappainter.h, 102
- src/editor/Game/mapslistmodel.h, 103
- src/editor/Game/object.h, 104
- src/editor/Game/xmlhandler.h, 114
- src::client::Client
  - \_\_del\_\_, 43
- src::network::NetworkClient
  - askEntity, 74
  - run, 74
  - sendEvent, 74
- src::network::NetworkServer
  - \_\_init\_\_, 76
  - run, 76
  - sendOrder, 76
  - waitForClients, 77
- src::network::ServerConnection
  - run, 93
- src::orders::Order
  - fromBytes, 85
  - load, 85
  - toBytes, 85
- src::orders::OrderDispatcher
  - treat, 86
- src::parsing::map\_parser
  - get\_size, 21
  - map\_parser, 21
  - parse\_cell, 21
- src::path::Archi
  - get\_src\_dir, 23
  - get\_src\_file, 23
  - get\_static\_dir, 24



- [get\\_static\\_file, 24](#)
  - [get\\_xml\\_dir, 24](#)
  - [get\\_xml\\_file, 24](#)
  - [list\\_files, 24](#)
- [src::printWorld::WorldViewer](#)
  - [\\_\\_init\\_\\_, 98](#)
  - [end, 98](#)
  - [get\\_event, 98](#)
  - [move, 99](#)
  - [move\\_char, 99](#)
  - [propagate\\_trigger, 99](#)
  - [render, 99](#)
  - [update, 99](#)
- [src::tools::Perf](#)
  - [show, 88](#)
  - [tic, 88](#)
  - [toc, 88](#)
- [src::tools::Timer](#)
  - [add, 95](#)
- [src::world::BaseObject](#)
  - [contextEval, 26](#)
  - [load, 26](#)
- [src::world::Map](#)
  - [fill, 59](#)
- [src::world::ObjectType](#)
  - [create, 81](#)
- [TabAcces, 93](#)
- [TabBar, 93](#)
- [tic](#)
  - [src::tools::Perf, 88](#)
- [toBytes](#)
  - [src::orders::Order, 85](#)
- [toc](#)
  - [src::tools::Perf, 88](#)
- [touch](#)
  - [GameObject, 51](#)
- [treat](#)
  - [src::orders::OrderDispatcher, 86](#)
- [update](#)
  - [src::printWorld::WorldViewer, 99](#)
- [viewCenter](#)
  - [MapPainter, 68](#)
- [viewCenterChanged](#)
  - [MapPainter, 68](#)
- [virtualSize](#)
  - [MapPainter, 68](#)
- [WIN](#)
  - [options.h, 116](#)
- [waitForClients](#)
  - [src::network::NetworkServer, 77](#)
- [Welcome, 95](#)
- [World, 96](#)
- [WorldEditor, 97](#)
- [XmlHandler, 100](#)
- [zoom](#)
  - [MapPainter, 69](#)