

Indie Studio

Generated by Doxygen 1.8.13



# Contents

<b>1</b>	<b>Hierarchical Index</b>	<b>1</b>
1.1	Class Hierarchy . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>File Index</b>	<b>5</b>
3.1	File List . . . . .	5
<b>4</b>	<b>Class Documentation</b>	<b>9</b>
4.1	Model3d::Animation Struct Reference . . . . .	9
4.1.1	Detailed Description . . . . .	9
4.2	AObject Class Reference . . . . .	9
4.2.1	Detailed Description . . . . .	10
4.2.2	Constructor & Destructor Documentation . . . . .	10
4.2.2.1	~AObject() . . . . .	10
4.2.3	Member Function Documentation . . . . .	10
4.2.3.1	getId() . . . . .	10
4.2.3.2	getModel() . . . . .	11
4.2.3.3	getType() . . . . .	11
4.2.3.4	print() . . . . .	11
4.2.3.5	setPosition() . . . . .	11
4.2.3.6	setRotation() . . . . .	11
4.3	APickable Class Reference . . . . .	12
4.3.1	Detailed Description . . . . .	12

4.3.2	Constructor & Destructor Documentation . . . . .	12
4.3.2.1	APickable() . . . . .	12
4.3.2.2	~APickable() . . . . .	12
4.4	AScene Class Reference . . . . .	13
4.4.1	Detailed Description . . . . .	14
4.4.2	Constructor & Destructor Documentation . . . . .	14
4.4.2.1	AScene() . . . . .	14
4.4.2.2	~AScene() . . . . .	14
4.4.3	Member Function Documentation . . . . .	14
4.4.3.1	addWidget() . . . . .	14
4.4.3.2	delWidget() . . . . .	15
4.4.3.3	getEventReceiver() . . . . .	15
4.4.3.4	print() . . . . .	15
4.4.3.5	updateCamera() . . . . .	16
4.5	Button Class Reference . . . . .	16
4.5.1	Detailed Description . . . . .	16
4.5.2	Constructor & Destructor Documentation . . . . .	17
4.5.2.1	Button() . . . . .	17
4.5.2.2	~Button() . . . . .	17
4.5.3	Member Function Documentation . . . . .	17
4.5.3.1	print() . . . . .	17
4.6	Camera Class Reference . . . . .	18
4.6.1	Detailed Description . . . . .	18
4.6.2	Constructor & Destructor Documentation . . . . .	18
4.6.2.1	Camera() . . . . .	18
4.6.3	Member Function Documentation . . . . .	19
4.6.3.1	changePosition() . . . . .	19
4.6.3.2	changeSight() . . . . .	19
4.7	ConfigManager Class Reference . . . . .	19
4.7.1	Detailed Description . . . . .	20

4.7.2	Constructor & Destructor Documentation . . . . .	20
4.7.2.1	ConfigManager() . . . . .	20
4.7.3	Member Function Documentation . . . . .	20
4.7.3.1	createFile() . . . . .	20
4.7.3.2	loadConfig() . . . . .	21
4.7.4	Member Data Documentation . . . . .	21
4.7.4.1	sound . . . . .	21
4.7.4.2	videoConfig . . . . .	21
4.8	CoreManager Class Reference . . . . .	21
4.8.1	Detailed Description . . . . .	22
4.8.2	Constructor & Destructor Documentation . . . . .	22
4.8.2.1	CoreManager() . . . . .	22
4.8.2.2	~CoreManager() . . . . .	22
4.9	error::Exception Class Reference . . . . .	22
4.9.1	Detailed Description . . . . .	23
4.9.2	Constructor & Destructor Documentation . . . . .	23
4.9.2.1	Exception() . . . . .	23
4.9.2.2	~Exception() . . . . .	23
4.9.3	Member Function Documentation . . . . .	23
4.9.3.1	getSeverity() . . . . .	24
4.9.3.2	what() . . . . .	24
4.10	Game Class Reference . . . . .	24
4.10.1	Detailed Description . . . . .	24
4.10.2	Constructor & Destructor Documentation . . . . .	24
4.10.2.1	Game() . . . . .	24
4.10.2.2	~Game() . . . . .	25
4.10.3	Member Function Documentation . . . . .	25
4.10.3.1	getLevel() . . . . .	25
4.10.3.2	setLevel() . . . . .	25
4.11	GameLoadScene Class Reference . . . . .	26

4.11.1 Detailed Description . . . . .	26
4.11.2 Constructor & Destructor Documentation . . . . .	26
4.11.2.1 GameLoadScene() . . . . .	27
4.11.2.2 ~GameLoadScene() . . . . .	27
4.11.3 Member Function Documentation . . . . .	27
4.11.3.1 getEventReceiver() . . . . .	27
4.12 GameMenuReceiver Class Reference . . . . .	27
4.12.1 Detailed Description . . . . .	28
4.12.2 Constructor & Destructor Documentation . . . . .	28
4.12.2.1 GameMenuReceiver() . . . . .	28
4.12.2.2 ~GameMenuReceiver() . . . . .	28
4.12.3 Member Function Documentation . . . . .	28
4.12.3.1 OnEvent() . . . . .	28
4.13 GameMenuScene Class Reference . . . . .	29
4.13.1 Detailed Description . . . . .	29
4.13.2 Constructor & Destructor Documentation . . . . .	29
4.13.2.1 GameMenuScene() . . . . .	29
4.13.2.2 ~GameMenuScene() . . . . .	29
4.13.3 Member Function Documentation . . . . .	30
4.13.3.1 getEventReceiver() . . . . .	30
4.14 GameNewReceiver Class Reference . . . . .	30
4.14.1 Detailed Description . . . . .	30
4.14.2 Constructor & Destructor Documentation . . . . .	30
4.14.2.1 GameNewReceiver() . . . . .	31
4.14.2.2 ~GameNewReceiver() . . . . .	31
4.14.3 Member Function Documentation . . . . .	31
4.14.3.1 OnEvent() . . . . .	31
4.15 GameNewScene Class Reference . . . . .	31
4.15.1 Detailed Description . . . . .	32
4.15.2 Constructor & Destructor Documentation . . . . .	32

4.15.2.1	<a href="#">GameNewScene()</a>	32
4.15.2.2	<a href="#">~GameNewScene()</a>	32
4.15.3	<a href="#">Member Function Documentation</a>	32
4.15.3.1	<a href="#">getEventReceiver()</a>	32
4.16	<a href="#">GameOnlineScene Class Reference</a>	33
4.16.1	<a href="#">Detailed Description</a>	33
4.16.2	<a href="#">Constructor &amp; Destructor Documentation</a>	33
4.16.2.1	<a href="#">GameOnlineScene()</a>	33
4.16.2.2	<a href="#">~GameOnlineScene()</a>	33
4.16.3	<a href="#">Member Function Documentation</a>	34
4.16.3.1	<a href="#">getEventReceiver()</a>	34
4.17	<a href="#">GameSimulationScene Class Reference</a>	34
4.17.1	<a href="#">Detailed Description</a>	34
4.17.2	<a href="#">Constructor &amp; Destructor Documentation</a>	35
4.17.2.1	<a href="#">GameSimulationScene()</a>	35
4.17.2.2	<a href="#">~GameSimulationScene()</a>	35
4.17.3	<a href="#">Member Function Documentation</a>	35
4.17.3.1	<a href="#">getEventReceiver()</a>	35
4.17.3.2	<a href="#">updateCamera()</a>	35
4.18	<a href="#">Image Class Reference</a>	36
4.18.1	<a href="#">Detailed Description</a>	36
4.18.2	<a href="#">Constructor &amp; Destructor Documentation</a>	36
4.18.2.1	<a href="#">Image()</a>	36
4.18.2.2	<a href="#">~Image()</a>	37
4.18.3	<a href="#">Member Function Documentation</a>	37
4.18.3.1	<a href="#">print()</a>	37
4.19	<a href="#">IntroScene Class Reference</a>	37
4.19.1	<a href="#">Detailed Description</a>	38
4.19.2	<a href="#">Constructor &amp; Destructor Documentation</a>	38
4.19.2.1	<a href="#">IntroScene()</a>	38

4.19.2.2	<a href="#">~IntroScene()</a>	38
4.19.3	<a href="#">Member Function Documentation</a>	38
4.19.3.1	<a href="#">getEventReceiver()</a>	38
4.20	<a href="#">IWidget Class Reference</a>	39
4.20.1	<a href="#">Detailed Description</a>	39
4.20.2	<a href="#">Constructor &amp; Destructor Documentation</a>	39
4.20.2.1	<a href="#">~IWidget()</a>	39
4.20.3	<a href="#">Member Function Documentation</a>	39
4.20.3.1	<a href="#">print()</a>	39
4.21	<a href="#">LegendScene Class Reference</a>	40
4.21.1	<a href="#">Detailed Description</a>	40
4.21.2	<a href="#">Constructor &amp; Destructor Documentation</a>	40
4.21.2.1	<a href="#">LegendScene()</a>	40
4.21.2.2	<a href="#">~LegendScene()</a>	41
4.21.3	<a href="#">Member Function Documentation</a>	41
4.21.3.1	<a href="#">getEventReceiver()</a>	41
4.22	<a href="#">Level Class Reference</a>	41
4.22.1	<a href="#">Detailed Description</a>	42
4.22.2	<a href="#">Constructor &amp; Destructor Documentation</a>	42
4.22.2.1	<a href="#">Level()</a>	42
4.22.2.2	<a href="#">~Level()</a>	42
4.22.3	<a href="#">Member Function Documentation</a>	42
4.22.3.1	<a href="#">loadLevel()</a>	42
4.22.3.2	<a href="#">movePlayer()</a>	42
4.22.3.3	<a href="#">print()</a>	43
4.23	<a href="#">MainScene Class Reference</a>	43
4.23.1	<a href="#">Detailed Description</a>	44
4.23.2	<a href="#">Constructor &amp; Destructor Documentation</a>	44
4.23.2.1	<a href="#">MainScene()</a>	44
4.23.2.2	<a href="#">~MainScene()</a>	44



4.23.3 Member Function Documentation . . . . .	44
4.23.3.1 getEventReceiver() . . . . .	44
4.24 Map Class Reference . . . . .	45
4.24.1 Detailed Description . . . . .	45
4.24.2 Constructor & Destructor Documentation . . . . .	45
4.24.2.1 Map() . . . . .	45
4.24.3 Member Function Documentation . . . . .	46
4.24.3.1 loadMap() . . . . .	46
4.24.3.2 loadSceneNodes() . . . . .	46
4.25 MapFactory Class Reference . . . . .	46
4.25.1 Detailed Description . . . . .	46
4.26 Model3d Class Reference . . . . .	47
4.26.1 Detailed Description . . . . .	47
4.26.2 Constructor & Destructor Documentation . . . . .	47
4.26.2.1 Model3d() . . . . .	47
4.26.2.2 ~Model3d() . . . . .	48
4.26.3 Member Function Documentation . . . . .	48
4.26.3.1 print() . . . . .	48
4.26.3.2 setCurrentAnim() . . . . .	48
4.27 ModelsManager Class Reference . . . . .	49
4.27.1 Detailed Description . . . . .	49
4.27.2 Constructor & Destructor Documentation . . . . .	49
4.27.2.1 ModelsManager() . . . . .	49
4.27.3 Member Function Documentation . . . . .	49
4.27.3.1 getModel() . . . . .	49
4.27.3.2 loadModel() . . . . .	50
4.28 OptionControl Class Reference . . . . .	50
4.28.1 Detailed Description . . . . .	50
4.29 OptionScene Class Reference . . . . .	51
4.29.1 Detailed Description . . . . .	51

4.29.2	Constructor & Destructor Documentation . . . . .	51
4.29.2.1	OptionScene() . . . . .	51
4.29.2.2	~OptionScene() . . . . .	51
4.29.3	Member Function Documentation . . . . .	52
4.29.3.1	getEventReceiver() . . . . .	52
4.30	OptionSoundReceiver Class Reference . . . . .	52
4.30.1	Detailed Description . . . . .	52
4.30.2	Constructor & Destructor Documentation . . . . .	52
4.30.2.1	OptionSoundReceiver() . . . . .	53
4.30.2.2	~OptionSoundReceiver() . . . . .	53
4.30.3	Member Function Documentation . . . . .	53
4.30.3.1	OnEvent() . . . . .	53
4.31	OptionSoundScene Class Reference . . . . .	53
4.31.1	Detailed Description . . . . .	54
4.31.2	Constructor & Destructor Documentation . . . . .	54
4.31.2.1	OptionSoundScene() . . . . .	54
4.31.2.2	~OptionSoundScene() . . . . .	54
4.31.3	Member Function Documentation . . . . .	54
4.31.3.1	getEventReceiver() . . . . .	54
4.32	OptionVideoReceiver Class Reference . . . . .	55
4.32.1	Detailed Description . . . . .	55
4.32.2	Constructor & Destructor Documentation . . . . .	55
4.32.2.1	OptionVideoReceiver() . . . . .	55
4.32.2.2	~OptionVideoReceiver() . . . . .	55
4.32.3	Member Function Documentation . . . . .	55
4.32.3.1	OnEvent() . . . . .	56
4.33	OptionVideoScene Class Reference . . . . .	56
4.33.1	Detailed Description . . . . .	56
4.33.2	Constructor & Destructor Documentation . . . . .	56
4.33.2.1	OptionVideoScene() . . . . .	57

4.33.2.2	<a href="#">~OptionVideoScene()</a> . . . . .	57
4.33.3	<a href="#">Member Function Documentation</a> . . . . .	57
4.33.3.1	<a href="#">getEventReceiver()</a> . . . . .	57
4.34	<a href="#">Potion Class Reference</a> . . . . .	57
4.34.1	<a href="#">Detailed Description</a> . . . . .	58
4.34.2	<a href="#">Constructor &amp; Destructor Documentation</a> . . . . .	58
4.34.2.1	<a href="#">Potion()</a> . . . . .	58
4.34.2.2	<a href="#">~Potion()</a> . . . . .	58
4.35	<a href="#">SaveManager Class Reference</a> . . . . .	58
4.35.1	<a href="#">Detailed Description</a> . . . . .	59
4.35.2	<a href="#">Member Function Documentation</a> . . . . .	59
4.35.2.1	<a href="#">loadLevel()</a> . . . . .	59
4.36	<a href="#">ScoreManager Class Reference</a> . . . . .	59
4.36.1	<a href="#">Detailed Description</a> . . . . .	60
4.36.2	<a href="#">Constructor &amp; Destructor Documentation</a> . . . . .	60
4.36.2.1	<a href="#">ScoreManager()</a> . . . . .	60
4.36.3	<a href="#">Member Function Documentation</a> . . . . .	60
4.36.3.1	<a href="#">addScore()</a> . . . . .	60
4.36.3.2	<a href="#">changeScore()</a> . . . . .	62
4.36.3.3	<a href="#">createFile()</a> . . . . .	62
4.36.3.4	<a href="#">getPlayerScore()</a> . . . . .	62
4.36.3.5	<a href="#">loadScores()</a> . . . . .	63
4.37	<a href="#">ScoreScene Class Reference</a> . . . . .	63
4.37.1	<a href="#">Detailed Description</a> . . . . .	63
4.37.2	<a href="#">Constructor &amp; Destructor Documentation</a> . . . . .	64
4.37.2.1	<a href="#">ScoreScene()</a> . . . . .	64
4.37.2.2	<a href="#">~ScoreScene()</a> . . . . .	64
4.37.3	<a href="#">Member Function Documentation</a> . . . . .	64
4.37.3.1	<a href="#">getEventReceiver()</a> . . . . .	64
4.37.3.2	<a href="#">getScores()</a> . . . . .	64

4.37.3.3	to_WString()	65
4.38	Text Class Reference	65
4.38.1	Detailed Description	65
4.38.2	Constructor & Destructor Documentation	65
4.38.2.1	Text()	65
4.38.3	Member Function Documentation	66
4.38.3.1	print()	66
4.39	Window Class Reference	66
4.39.1	Detailed Description	67
4.39.2	Constructor & Destructor Documentation	67
4.39.2.1	Window()	67
4.39.2.2	~Window()	67
4.39.3	Member Function Documentation	67
4.39.3.1	addScene()	68
4.39.3.2	deleteScene()	68
4.39.3.3	isOpen()	68
4.39.3.4	loop()	68
4.39.3.5	printScene()	69
4.39.3.6	setCurrentScene()	69
<b>5</b>	<b>File Documentation</b>	<b>71</b>
5.1	inc/exception/Exception.hh File Reference	71
5.1.1	Detailed Description	71
5.2	inc/game/Game.hh File Reference	71
5.2.1	Detailed Description	72
5.3	inc/game/level/Level.hh File Reference	72
5.3.1	Detailed Description	72
5.4	inc/game/level/Map.hh File Reference	73
5.4.1	Detailed Description	73
5.5	inc/game/level/MapFactory.hh File Reference	73
5.5.1	Detailed Description	74

5.6	<a href="#">inc/game/object/AObject.hh File Reference</a>	74
5.6.1	<a href="#">Detailed Description</a>	74
5.7	<a href="#">inc/game/object/APickable.hh File Reference</a>	74
5.7.1	<a href="#">Detailed Description</a>	75
5.8	<a href="#">inc/graphics/AScene.hh File Reference</a>	75
5.8.1	<a href="#">Detailed Description</a>	75
5.9	<a href="#">inc/graphics/Camera.hh File Reference</a>	76
5.9.1	<a href="#">Detailed Description</a>	76
5.10	<a href="#">inc/graphics/scenes/events/GameMenuReceiver.hh File Reference</a>	76
5.10.1	<a href="#">Detailed Description</a>	76
5.11	<a href="#">inc/graphics/scenes/events/GameNewReceiver.hh File Reference</a>	77
5.11.1	<a href="#">Detailed Description</a>	77
5.12	<a href="#">inc/graphics/scenes/events/OptionSoundReceiver.hh File Reference</a>	77
5.12.1	<a href="#">Detailed Description</a>	77
5.13	<a href="#">inc/graphics/scenes/events/OptionVideoReceiver.hh File Reference</a>	78
5.13.1	<a href="#">Detailed Description</a>	78
5.14	<a href="#">inc/graphics/scenes/GameLoadScene.hh File Reference</a>	78
5.14.1	<a href="#">Detailed Description</a>	78
5.15	<a href="#">inc/graphics/scenes/GameMenuScene.hh File Reference</a>	79
5.15.1	<a href="#">Detailed Description</a>	79
5.16	<a href="#">inc/graphics/scenes/GameNewScene.hh File Reference</a>	79
5.16.1	<a href="#">Detailed Description</a>	79
5.17	<a href="#">inc/graphics/scenes/GameOnlineScene.hh File Reference</a>	80
5.17.1	<a href="#">Detailed Description</a>	80
5.18	<a href="#">inc/graphics/scenes/GameScene.hh File Reference</a>	80
5.18.1	<a href="#">Detailed Description</a>	80
5.19	<a href="#">inc/graphics/scenes/GameSimulationScene.hh File Reference</a>	81
5.19.1	<a href="#">Detailed Description</a>	81
5.20	<a href="#">inc/graphics/scenes/IntroScene.hh File Reference</a>	81
5.20.1	<a href="#">Detailed Description</a>	81

5.21	<a href="#">inc/graphics/scenes/LegendScene.hh File Reference</a>	82
5.21.1	<a href="#">Detailed Description</a>	82
5.22	<a href="#">inc/graphics/scenes/MainScene.hh File Reference</a>	82
5.22.1	<a href="#">Detailed Description</a>	82
5.23	<a href="#">inc/graphics/scenes/OptionControlScene.hh File Reference</a>	83
5.23.1	<a href="#">Detailed Description</a>	83
5.24	<a href="#">inc/graphics/scenes/OptionScene.hh File Reference</a>	83
5.24.1	<a href="#">Detailed Description</a>	83
5.25	<a href="#">inc/graphics/scenes/OptionSoundScene.hh File Reference</a>	84
5.25.1	<a href="#">Detailed Description</a>	84
5.26	<a href="#">inc/graphics/scenes/OptionVideoScene.hh File Reference</a>	84
5.26.1	<a href="#">Detailed Description</a>	84
5.27	<a href="#">inc/graphics/scenes/ScoreScene.hh File Reference</a>	85
5.27.1	<a href="#">Detailed Description</a>	85
5.28	<a href="#">inc/graphics/widgets/Button.hh File Reference</a>	85
5.28.1	<a href="#">Detailed Description</a>	86
5.29	<a href="#">inc/graphics/widgets/Image.hh File Reference</a>	86
5.29.1	<a href="#">Detailed Description</a>	86
5.30	<a href="#">inc/graphics/widgets/IWidget.hh File Reference</a>	86
5.30.1	<a href="#">Detailed Description</a>	87
5.31	<a href="#">inc/graphics/widgets/Model3d.hh File Reference</a>	87
5.31.1	<a href="#">Detailed Description</a>	87
5.32	<a href="#">inc/graphics/widgets/Text.hh File Reference</a>	88
5.32.1	<a href="#">Detailed Description</a>	88
5.33	<a href="#">inc/graphics/Window.hh File Reference</a>	88
5.33.1	<a href="#">Detailed Description</a>	89
5.34	<a href="#">inc/managers/config/Model.hh File Reference</a>	89
5.34.1	<a href="#">Detailed Description</a>	89
5.35	<a href="#">inc/managers/ConfigManager.hh File Reference</a>	89
5.35.1	<a href="#">Detailed Description</a>	90
5.36	<a href="#">inc/managers/CoreManager.hh File Reference</a>	90
5.36.1	<a href="#">Detailed Description</a>	90
5.37	<a href="#">inc/managers/LevelManager.hh File Reference</a>	91
5.37.1	<a href="#">Detailed Description</a>	91
5.38	<a href="#">inc/managers/ModelsManager.hh File Reference</a>	91
5.38.1	<a href="#">Detailed Description</a>	92
5.39	<a href="#">inc/managers/SaveManager.hh File Reference</a>	92
5.39.1	<a href="#">Detailed Description</a>	92
5.40	<a href="#">inc/managers/ScoreManager.hh File Reference</a>	92
5.40.1	<a href="#">Detailed Description</a>	93

# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

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

Model3d::Animation . . . . .	9
AObject . . . . .	9
APickable . . . . .	12
Potion . . . . .	57
AScene . . . . .	13
GameLoadScene . . . . .	26
GameMenuScene . . . . .	29
GameNewScene . . . . .	31
GameOnlineScene . . . . .	33
GameSimulationScene . . . . .	34
IntroScene . . . . .	37
LegendScene . . . . .	40
MainScene . . . . .	43
OptionScene . . . . .	51
OptionSoundScene . . . . .	53
OptionVideoScene . . . . .	56
ScoreScene . . . . .	63
Camera . . . . .	18
ConfigManager . . . . .	19
CoreManager . . . . .	21
error::Exception . . . . .	22
Game . . . . .	24
GameMenuReceiver . . . . .	27
GameNewReceiver . . . . .	30
IWidget . . . . .	39
Button . . . . .	16
Image . . . . .	36
Level . . . . .	41
Map . . . . .	45
Model3d . . . . .	47
Text . . . . .	65
MapFactory . . . . .	46
ModelsManager . . . . .	49
OptionControl . . . . .	50

OptionSoundReceiver . . . . .	52
OptionVideoReceiver . . . . .	55
SaveManager . . . . .	58
ScoreManager . . . . .	59
Window . . . . .	66



## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">Model3d::Animation</a>	Represent a model's animation . . . . .	9
<a href="#">AObject</a>	An <a href="#">AObject</a> instance has his own <a href="#">Model3d</a> attribute, and can modify it. . . . .	9
<a href="#">APickable</a>	Create a pickable <a href="#">AObject</a> . . . . .	12
<a href="#">AScene</a>	Represent the window's scene that will print some widget/object of the scene : buttons, images, etc.. . . . .	13
<a href="#">Button</a>	Represent a button in the user interface . . . . .	16
<a href="#">Camera</a>	Where cameras behaviour is defined . . . . .	18
<a href="#">ConfigManager</a>	This class contain all the methode to manipulate the configuration . . . . .	19
<a href="#">CoreManager</a>	<a href="#">CoreManager</a> is the main manager who regroupes all the other manager . . . . .	21
<a href="#">error::Exception</a>	This class allow to throw some execption and handle it finely . . . . .	22
<a href="#">Game</a>	The <a href="#">Game</a> class . . . . .	24
<a href="#">GameLoadScene</a>	<a href="#">Button</a> interface of GameLoad scene . . . . .	26
<a href="#">GameMenuReceiver</a>	Manager of event of game scene . . . . .	27
<a href="#">GameMenuScene</a>	<a href="#">Button</a> interface of menu setting scene . . . . .	29
<a href="#">GameNewReceiver</a>	Manager of event of new game scene . . . . .	30
<a href="#">GameNewScene</a>	<a href="#">Button</a> interface of new game setting scene . . . . .	31
<a href="#">GameOnlineScene</a>	<a href="#">Button</a> interface of GameOnlinescene . . . . .	33
<a href="#">GameSimulationScene</a>	<a href="#">Button</a> interface of GameSimulationscene . . . . .	34

<a href="#">Image</a>	Represent a image in the user interface . . . . .	36
<a href="#">IntroScene</a>	<a href="#">Button</a> interface of Introcene . . . . .	37
<a href="#">IWidget</a>	Represent a generic object in the user interface . . . . .	39
<a href="#">LegendScene</a>	<a href="#">Button</a> interface of <a href="#">LegendScene</a> . . . . .	40
<a href="#">Level</a>	Represent a game's level . . . . .	41
<a href="#">MainScene</a>	<a href="#">Button</a> interface of main scene . . . . .	43
<a href="#">Map</a>	Represent a level's map . . . . .	45
<a href="#">MapFactory</a>	Represent a maps' factory . . . . .	46
<a href="#">Model3d</a>	Represent a model 3d in the user interface . . . . .	47
<a href="#">ModelsManager</a>	The class to load all the information about a model . . . . .	49
<a href="#">OptionControl</a>	<a href="#">Button</a> interface of OptionControlscene . . . . .	50
<a href="#">OptionScene</a>	<a href="#">Button</a> interface of option scene . . . . .	51
<a href="#">OptionSoundReceiver</a>	Manager of event of Sound scene setting . . . . .	52
<a href="#">OptionSoundScene</a>	<a href="#">Button</a> interface of sound setting scene . . . . .	53
<a href="#">OptionVideoReceiver</a>	Manager of event of video setting scene . . . . .	55
<a href="#">OptionVideoScene</a>	<a href="#">Button</a> interface of video setting scene . . . . .	56
<a href="#">Potion</a>	Create a <a href="#">APickable AObject</a> key . . . . .	57
<a href="#">SaveManager</a>	The <a href="#">SaveManager</a> class . . . . .	58
<a href="#">ScoreManager</a>	The <a href="#">ScoreManager</a> class . . . . .	59
<a href="#">ScoreScene</a>	<a href="#">Button</a> interface of <a href="#">ScoreScene</a> . . . . .	63
<a href="#">Text</a>	Inherited from a <a href="#">IWidget</a> . . . . .	65
<a href="#">Window</a>	Represent the window that will print some scene (view of the application) . . . . .	66

## Chapter 3

# File Index

### 3.1 File List

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

inc/ <b>graphics_header.hh</b> . . . . .	??
inc/ <b>type_compatibility.hh</b> . . . . .	??
inc/ <b>VideoController.hh</b> . . . . .	??
inc/event/ <b>Controller.hh</b> . . . . .	??
inc/event/ <b>ControllerManager.hh</b> . . . . .	??
inc/event/ <b>ControllerType.hh</b> . . . . .	??
inc/event/ <b>Event.hh</b> . . . . .	??
inc/exception/ <b>Exception.hh</b>	
Exception class . . . . .	71
inc/game/ <b>Game.hh</b>	
Contain all the information about the game . . . . .	71
inc/game/character/ <b>ACharacter.hh</b> . . . . .	??
inc/game/character/heros/ <b>AHero.hh</b> . . . . .	??
inc/game/character/heros/classtype/ <b>Archer.hh</b> . . . . .	??
inc/game/character/heros/classtype/ <b>Monk.hh</b> . . . . .	??
inc/game/character/heros/classtype/ <b>Ninja.hh</b> . . . . .	??
inc/game/character/heros/classtype/ <b>Warrior.hh</b> . . . . .	??
inc/game/character/IA/ <b>IACloseRange.hh</b> . . . . .	??
inc/game/character/IA/ <b>IAHero.hh</b> . . . . .	??
inc/game/character/IA/ <b>IALongRange.hh</b> . . . . .	??
inc/game/character/IA/ <b>IIA.hh</b> . . . . .	??
inc/game/character/monster/ <b>AMonster.hh</b> . . . . .	??
inc/game/character/monster/ <b>Spawner.hpp</b> . . . . .	??
inc/game/character/monster/CrawlingDog/ <b>CrawlingDog.hh</b> . . . . .	??
inc/game/character/monster/ForestGuard/ <b>ForestGuard.hh</b> . . . . .	??
inc/game/character/monster/MuscledMutant/ <b>MuscledMutant.hh</b> . . . . .	??
inc/game/level/ <b>Level.hh</b>	
Use to create a level of the game . . . . .	72
inc/game/level/ <b>Map.hh</b>	
Defines the map that is related to a level. The process to create a map is relatively simple. First of all, you've got to load a map with <b>Map::loadMap()</b> In a second time, you've got to retrieve all the nodes with a triangle selector container (meta) with <b>loadSceneNodes</b> Then, you can create collision response thanks to these nodes by calling <b>initCollisionResponseAnimator()</b> Finally, you can draw the map (that should be achieved with <b>drawAll()</b> bugs: If meshes/models arent loading properly. Double check the paths in the .irr file . . . . .	73

inc/game/level/ <a href="#">MapFactory.hh</a>	
Maps' factory use to build the maps objects	73
inc/game/object/ <a href="#">AObject.hh</a>	
Use to create object	74
inc/game/object/ <a href="#">APickable.hh</a>	
Use to create pickable object	74
inc/game/object/ <a href="#">Door.hh</a>	??
inc/game/object/ <a href="#">Inventory.hh</a>	??
inc/game/object/ <a href="#">Key.hh</a>	??
inc/game/object/ <a href="#">Potion.hh</a>	??
inc/game/object/ <a href="#">Treasure.hh</a>	??
inc/game/object/ <a href="#">Wall.hh</a>	??
inc/game/object/weapon/ <a href="#">AWeapon.hh</a>	??
inc/game/object/weapon/ <a href="#">Bow.hh</a>	??
inc/game/object/weapon/ <a href="#">DualDagger.hh</a>	??
inc/game/object/weapon/ <a href="#">Fist.hh</a>	??
inc/game/object/weapon/ <a href="#">Katana.hh</a>	??
inc/graphics/ <a href="#">AScene.hh</a>	
Define a window's scene	75
inc/graphics/ <a href="#">Camera.hh</a>	
Defines the camera, its type and behaviour bugs: Actually you have to give the constructor the camera. That may change soon with integration on indie_studio This way, cam will be created inside cosntructor with the g_Window->sceneManager that would be such accessible	76
inc/graphics/ <a href="#">Window.hh</a>	
Define a window	88
inc/graphics/scenes/ <a href="#">CreditScene.hh</a>	??
inc/graphics/scenes/ <a href="#">GameLoadScene.hh</a>	
Use to display the gameload scene	78
inc/graphics/scenes/ <a href="#">GameMenuScene.hh</a>	
Use to choose between new game and load game	79
inc/graphics/scenes/ <a href="#">GameNewScene.hh</a>	
Use to choose characters for each player	79
inc/graphics/scenes/ <a href="#">GameOnlineScene.hh</a>	
Use to display the GameOnline scene	80
inc/graphics/scenes/ <a href="#">GameScene.hh</a>	
Use to display the <a href="#">Game</a> scene	80
inc/graphics/scenes/ <a href="#">GameSimulationScene.hh</a>	
Use to display the GameOnline scene	81
inc/graphics/scenes/ <a href="#">IntroScene.hh</a>	
Use to display the <a href="#">IntroScene</a> scene	81
inc/graphics/scenes/ <a href="#">LegendScene.hh</a>	
Use to display the Legend scene	82
inc/graphics/scenes/ <a href="#">MainScene.hh</a>	
Main scene who lead to each menus	82
inc/graphics/scenes/ <a href="#">OptionControlScene.hh</a>	
Use to display the <a href="#">OptionControl</a> scene	83
inc/graphics/scenes/ <a href="#">OptionScene.hh</a>	
Use to display the option menu	83
inc/graphics/scenes/ <a href="#">OptionSoundScene.hh</a>	
Use to create the button interface of the sound setting scene	84
inc/graphics/scenes/ <a href="#">OptionVideoScene.hh</a>	
Use to create the button interface of the video setting scene	84
inc/graphics/scenes/ <a href="#">ScoreScene.hh</a>	
Use to display the Score scene	85
inc/graphics/scenes/events/ <a href="#">CreditReceiver.hh</a>	??
inc/graphics/scenes/events/ <a href="#">GameLoadReceiver.hh</a>	??
inc/graphics/scenes/events/ <a href="#">GameMenuReceiver.hh</a>	
Use to manage event of the game (new game / load) scene	76

inc/graphics/scenes/events/ <a href="#">GameNewReceiver.hh</a>	
Use to manage event of the new game (character selection) scene	77
inc/graphics/scenes/events/ <b>GameOnlineReceiver.hh</b>	??
inc/graphics/scenes/events/ <b>GameReceiver.hh</b>	??
inc/graphics/scenes/events/ <b>GameSimulationReceiver.hh</b>	??
inc/graphics/scenes/events/ <b>IntroReceiver.hh</b>	??
inc/graphics/scenes/events/ <b>LegendReceiver.hh</b>	??
inc/graphics/scenes/events/ <b>MainReceiver.hh</b>	??
inc/graphics/scenes/events/ <b>OptionControlReceiver.hh</b>	??
inc/graphics/scenes/events/ <b>OptionReceiver.hh</b>	??
inc/graphics/scenes/events/ <a href="#">OptionSoundReceiver.hh</a>	
Allow to manage event of Sound scene setting	77
inc/graphics/scenes/events/ <a href="#">OptionVideoReceiver.hh</a>	
Use to manage event of the video setting scene	78
inc/graphics/scenes/events/ <b>ScoreReceiver.hh</b>	??
inc/graphics/widgets/ <a href="#">Button.hh</a>	
Allow to add some button in the window	85
inc/graphics/widgets/ <a href="#">Image.hh</a>	
Allow to add some image in the window	86
inc/graphics/widgets/ <a href="#">IWidget.hh</a>	
Give a interface for all the class that could be print on the window	86
inc/graphics/widgets/ <a href="#">Model3d.hh</a>	
Allow to add some model 3d in the window	87
inc/graphics/widgets/ <a href="#">Text.hh</a>	
Allow you to create a text object	88
inc/indie/ <b>GUIButtonId.hh</b>	??
inc/managers/ <a href="#">ConfigManager.hh</a>	
Allow you to load, modify and save the configuration	89
inc/managers/ <a href="#">CoreManager.hh</a>	
<a href="#">CoreManager</a> regroupes all the application's manager	90
inc/managers/ <b>EventManager.hh</b>	??
inc/managers/ <a href="#">LevelManager.hh</a>	
Load level configuration out of XML format	91
inc/managers/ <a href="#">ModelsManager.hh</a>	
The Models manager will allow you to get all the information to load a model	91
inc/managers/ <a href="#">SaveManager.hh</a>	
The save manager will allow you to save and restore a game	92
inc/managers/ <a href="#">ScoreManager.hh</a>	
The score manager will allow you to save, load and restore game scores	92
inc/managers/ <b>SoundManager.hh</b>	??
inc/managers/config/ <b>Control.hh</b>	??
inc/managers/config/ <b>Devices.hh</b>	??
inc/managers/config/ <b>IControl.hh</b>	??
inc/managers/config/ <b>LevelDefinition.hh</b>	??
inc/managers/config/ <a href="#">Model.hh</a>	
The Class Model allow you to manipulate model's informations	89
inc/managers/config/ <b>Score.hh</b>	??
inc/managers/config/ <b>Sound.hh</b>	??
inc/managers/config/ <b>Video.hh</b>	??
inc/tools/ <b>Maths.hh</b>	??
inc/tools/ <b>Transform.hh</b>	??



## Chapter 4

# Class Documentation

### 4.1 Model3d::Animation Struct Reference

Represent a model's animation.

```
#include <Model3d.hh>
```

#### 4.1.1 Detailed Description

Represent a model's animation.

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

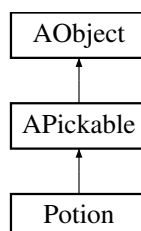
- inc/graphics/widgets/[Model3d.hh](#)

### 4.2 AObject Class Reference

An [AObject](#) instance has his own [Model3d](#) attribute, and can modify it..

```
#include <AObject.hh>
```

Inheritance diagram for AObject:



## Public Member Functions

- [AObject](#) (std::shared\_ptr< [Model3d](#) >, indie::objectType=indie::objectType::UNKNOWN, irr::u32 id=0)  
*Constructor Build the class first param is a shared\_ptr on a [Model3d](#), second the type of object (CHEST, POTION, ...), third his id.*
- [~AObject](#) ()  
*Destructor.*
- indie::objectType [getType](#) () const  
*getType*
- std::shared\_ptr< [Model3d](#) > [getModel](#) () const  
*getModel*
- void [setPosition](#) (Vector3d)  
*setPosition*
- void [setRotation](#) (Vector3d)  
*setRotation*
- void [print](#) () const  
*print*
- irr::u32 [getId](#) () const  
*getId*

### 4.2.1 Detailed Description

An [AObject](#) instance has his own [Model3d](#) attribute, and can modify it..

### 4.2.2 Constructor & Destructor Documentation

#### 4.2.2.1 ~AObject()

```
AObject::~~AObject ( )
```

Destructor.

Destroy the class

### 4.2.3 Member Function Documentation

#### 4.2.3.1 getId()

```
irr::u32 AObject::getId ( ) const
```

getId

return the id of the [AObject](#)



#### 4.2.3.2 getModel()

```
std::shared_ptr< Model3d > AObject::getModel ( ) const
```

getModel

return the [Model3d](#) of the [AObject](#)

#### 4.2.3.3 getType()

```
indie::objectType AObject::getType ( ) const
```

getType

return the type of the [AObject](#)

#### 4.2.3.4 print()

```
void AObject::print ( ) const
```

print

print the object on window

#### 4.2.3.5 setPosition()

```
void AObject::setPosition (
    Vector3d pos )
```

setPosition

change the vector position of the [Model3d](#) of the [AObject](#)

#### 4.2.3.6 setRotation()

```
void AObject::setRotation (
    Vector3d rot )
```

setRotation

change the vector rotation of the [Model3d](#) of the [AObject](#)

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

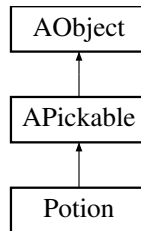
- [inc/game/object/AObject.hh](#)
- [src/game/object/AObject.cpp](#)

### 4.3 APickable Class Reference

Create a pickable [AObject](#).

```
#include <APickable.hh>
```

Inheritance diagram for APickable:



#### Public Member Functions

- [APickable](#) (indie::objectType type)  
*Constructor.*
- [~APickable](#) ()  
*Destructor.*

#### 4.3.1 Detailed Description

Create a pickable [AObject](#).

#### 4.3.2 Constructor & Destructor Documentation

##### 4.3.2.1 APickable()

```
APickable::APickable (
    indie::objectType type )
```

Constructor.

Build the class

##### 4.3.2.2 ~APickable()

```
APickable::~~APickable ( )
```

Destructor.

Destroy the class

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

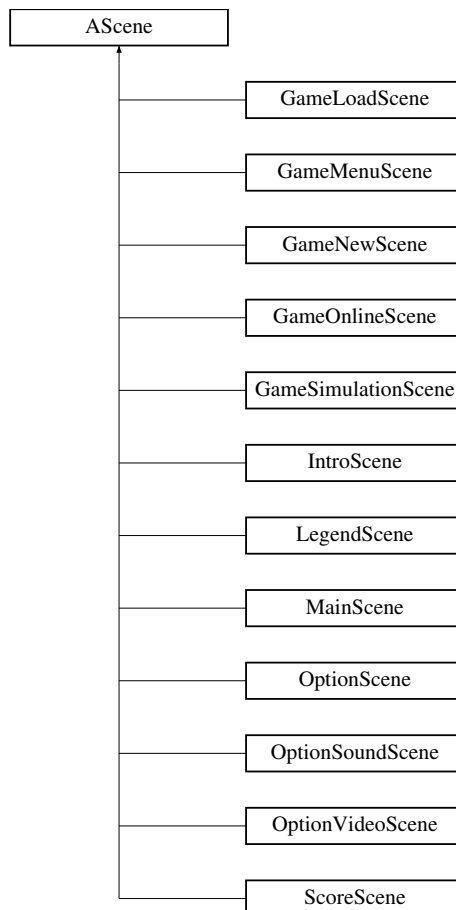
- inc/game/object/[APickable.hh](#)
- src/game/object/APickable.cpp

## 4.4 AScene Class Reference

Represent the window's scene that will print some widget/object of the scene : buttons, images, etc...

```
#include <AScene.hh>
```

Inheritance diagram for AScene:



### Public Member Functions

- [AScene](#) ()  
*Constructor.*
- [~AScene](#) ()  
*Destructor.*
- void [print](#) ([Window](#) \*win)  
*Print all the scene's widgets into the [Window](#).*
- virtual void [updateCamera](#) ([Window](#) \*win)  
*Print all the scene's widgets into the [Window](#).*
- void [addWidget](#) (const std::shared\_ptr< [IWidget](#) > &widget, Int id=-1)  
*Allow the user to add texts, images, buttons, checkboxes or 3d models into the scene.*
- void [delWidget](#) (Int id)  
*Allow the user to delete texts, images, buttons, checkboxes or 3d models into the scene.*
- virtual irr::IEventReceiver \* [getEventReceiver](#) () const =0  
*Allow the user to implemente the event callback class for each scene.*

#### 4.4.1 Detailed Description

Represent the window's scene that will print some widget/object of the scene : buttons, images, etc...

#### 4.4.2 Constructor & Destructor Documentation

##### 4.4.2.1 AScene()

```
AScene::AScene ( )
```

Constructor.

Build the class.

###### Parameters

None	
------	--

##### 4.4.2.2 ~AScene()

```
AScene::~~AScene ( )
```

Destructor.

Destroy the class.

#### 4.4.3 Member Function Documentation

##### 4.4.3.1 addWidget()

```
void AScene::addWidget (
    const std::shared_ptr< IWidget > & widget,
    Int id = -1 )
```

Allow the user to add texts, images, buttons, checkboxes or 3d models into the scene.

###### Parameters

'widget'	the widget to add in the scene.
----------	---------------------------------

#### 4.4.3.2 delWidget()

```
void AScene::delWidget (
    Int id )
```

Allow the user to delete texts, images, buttons, checkboxes or 3d models into the scene.

Example: scene->addWidget(widget, 2); scene->delWidget(2);

##### Parameters

'id'	the widget's id to del in the scene.
------	--------------------------------------

#### 4.4.3.3 getEventReceiver()

```
virtual irr::IEventReceiver* AScene::getEventReceiver ( ) const [pure virtual]
```

Allow the user to implemente the event callback class for each scene.

##### Parameters

None.	
-------	--

##### Returns

The events' receiver of the scene.

Implemented in [MainScene](#), [GameOnlineScene](#), [IntroScene](#), [LegendScene](#), [OptionScene](#), [OptionVideoScene](#), [GameSimulationScene](#), [OptionSoundScene](#), [GameLoadScene](#), [GameMenuScene](#), [GameNewScene](#), and [ScoreScene](#).

#### 4.4.3.4 print()

```
void AScene::print (
    Window * win )
```

Print all the scene's widgets into the [Window](#).

##### Parameters

'win'	the window where the scene will be print.
-------	---

#### 4.4.3.5 updateCamera()

```
virtual void AScene::updateCamera (
    Window * win ) [inline], [virtual]
```

Print all the scene's widgets into the [Window](#).

##### Parameters

<code>'win'</code>	the window where the scene will be print.
--------------------	---

Reimplemented in [GameSimulationScene](#).

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

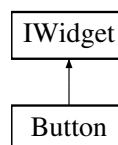
- inc/graphics/[AScene.hh](#)
- src/graphics/[AScene.cpp](#)

## 4.5 Button Class Reference

Represent a button in the user interface.

```
#include <Button.hh>
```

Inheritance diagram for Button:



### Public Member Functions

- [Button](#) (const Rect &rect, const wchar\_t \*text, const wchar\_t \*tooltext, enum indie::GUIButtonId id, const wchar\_t \*background=L"assets/menus/transparent.png")  
*Constructor.*
- [~Button](#) ()  
*Destructor.*
- void [print](#) ([Window](#) \*win)  
*Print the button into the window.*

#### 4.5.1 Detailed Description

Represent a button in the user interface.

## 4.5.2 Constructor & Destructor Documentation

### 4.5.2.1 Button()

```
Button::Button (
    const Rect & rect,
    const wchar_t * text,
    const wchar_t * tooltext,
    enum indie::GUIButtonId id,
    const wchar_t * background = L"assets/menus/transparent.png" )
```

Constructor.

Build the class.

#### Parameters

- |    |  |
|----|--|
| 1) | 'rect' the rectangle who represent the button (x1, y1, x2, y2). 2) 'background' the name of the image, that will be put on the background of the button. 3) 'text' the text that will be write on the button. 3) 'tooltext' the text displayed in the tooltip. 4) 'id' the id throw the user click on the button . |
|----|--|

### 4.5.2.2 ~Button()

```
Button::~~Button ( )
```

Destructor.

Destroy the class.

## 4.5.3 Member Function Documentation

### 4.5.3.1 print()

```
void Button::print (
    Window * win ) [virtual]
```

Print the button into the window.

#### Parameters

- |          |  |
|----------|--|
| 'window' | the window where the button will be print. |
|----------|--|

Implements [IWidget](#).

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

- inc/graphics/widgets/[Button.hh](#)
- src/graphics/widgets/Button.cpp

## 4.6 Camera Class Reference

where cameras behaviour is defined

```
#include <Camera.hh>
```

### Public Member Functions

- [Camera](#) (irr::scene::ICameraSceneNode \*const cam)  
*instanciate [Camera](#)*
- [~Camera](#) ()  
*Destructor...*
- void [changePosition](#) (irr::core::vector3df move)  
*change absolute camera position*
- void [changeSight](#) (irr::core::vector3df newTarget)  
*change camera sight*

### 4.6.1 Detailed Description

where cameras behaviour is defined

### 4.6.2 Constructor & Destructor Documentation

#### 4.6.2.1 Camera()

```
Camera::Camera (
    irr::scene::ICameraSceneNode *const cam )
```

instanciate [Camera](#)

#### Parameters

<i>cam</i>	: the irrlicht static camera
------------	------------------------------



### 4.6.3 Member Function Documentation

#### 4.6.3.1 changePosition()

```
void Camera::changePosition (
    irr::core::vector3df move )
```

change absolute camera position

##### Parameters

<i>move</i>	: irrlicht 3D vector to positions camera
-------------	--

#### 4.6.3.2 changeSight()

```
void Camera::changeSight (
    irr::core::vector3df newTarget )
```

change camera sight

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

- [inc/graphics/Camera.hh](#)
- [src/graphics/Camera.cpp](#)

## 4.7 ConfigManager Class Reference

This class contain all the methode to manipulate the configuration.

```
#include <ConfigManager.hh>
```

### Public Member Functions

- [ConfigManager](#) ()  
*Constructor.*
- void [loadConfig](#) (std::string)  
*Load the configuration.*
- void [saveConfig](#) () const  
*Savethe object in the file.*
- void [createFile](#) (std::string)  
*create an empty configuration file*

## Public Attributes

- config::Video [videoConfig](#)  
*The object videoConfig is a structure that contain all the setting for the video.*
- config::Sound [sound](#)  
*The object sound is a structure that constain all the setting for the sound.*
- config::Control [control](#)  
*The structure control give all the informations about the control keys.*

### 4.7.1 Detailed Description

This class contain all the methode to manipulate the configuration.

### 4.7.2 Constructor & Destructor Documentation

#### 4.7.2.1 ConfigManager()

```
ConfigManager::ConfigManager ( )
```

Constructor.

Build the class

##### Parameters

<i>nothing</i>	
----------------	--

### 4.7.3 Member Function Documentation

#### 4.7.3.1 createFile()

```
void ConfigManager::createFile (
    std::string fileName )
```

create an empty configuration file

##### Parameters

<i>path</i>	to the new configuration file
-------------	-------------------------------

#### 4.7.3.2 loadConfig()

```
void ConfigManager::loadConfig (
    std::string fileName )
```

Load the configuration.

##### Parameters

<i>path</i>	to the configuration file
-------------	---------------------------

##### Returns

void

### 4.7.4 Member Data Documentation

#### 4.7.4.1 sound

```
config::Sound ConfigManager::sound
```

The object sound is a structure that constain all the setting for the sound.

The sructure is on the config namespace

#### 4.7.4.2 videoConfig

```
config::Video ConfigManager::videoConfig
```

The object videoConfig is a structure that contain all the setting for the video.

The sructure is on the config namespace

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

- inc/managers/[ConfigManager.hh](#)
- src/managers/ConfigManager.cpp

## 4.8 CoreManager Class Reference

[CoreManager](#) is the main manager who regroupes all the other manager.

```
#include <CoreManager.hh>
```

## Public Member Functions

- [CoreManager](#) ()  
*Constructor.*
- [~CoreManager](#) ()  
*Destructor.*

### 4.8.1 Detailed Description

[CoreManager](#) is the main manager who regroupes all the other manager.

### 4.8.2 Constructor & Destructor Documentation

#### 4.8.2.1 CoreManager()

```
CoreManager::CoreManager ( )
```

Constructor.

Build the class.

##### Parameters

<i>None</i>	
-------------	--

#### 4.8.2.2 ~CoreManager()

```
CoreManager::~~CoreManager ( )
```

Destructor.

Destroy the class.

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

- inc/managers/[CoreManager.hh](#)
- src/managers/CoreManager.cpp

## 4.9 error::Exception Class Reference

This class allow to throw some exception and handle it finely.

```
#include <Exception.hh>
```

Inherits exception.

## Public Member Functions

- [Exception](#) (const std::string &msg, const error::severity &severity, const char \*file, const char \*func, int line)  
*/brief Constructor : Build the class.*
- [~Exception](#) () throw ()  
*/brief Destructor : Destroy the class.*
- const error::severity & [getSeverity](#) () const  
*/brief Give an access on the error's severity.*
- const char \* [what](#) () const throw ()  
*/brief Describe the error.*

### 4.9.1 Detailed Description

This class allow to throw some exception and handle it finely.

### 4.9.2 Constructor & Destructor Documentation

#### 4.9.2.1 Exception()

```
error::Exception::Exception (  
    const std::string & msg,  
    const error::severity & severity,  
    const char * file,  
    const char * func,  
    int line )
```

*/brief Constructor : Build the class.*

*/param* This function take 5 parameters : 1) 'file' is the file where the error occurred. 2) 'func' is the func where the error occurred. 3) 'line' is the line where the error occurred. 4) 'msg' is the message who describe the error. 5) 'severity' is the error's severity.

*/return* None.

#### 4.9.2.2 ~Exception()

```
error::Exception::~~Exception ( ) throw ( )    [inline]
```

*/brief Destructor : Destroy the class.*

*/param* None.

*/return* None.

### 4.9.3 Member Function Documentation

#### 4.9.3.1 getSeverity()

```
const error::severity & error::Exception::getSeverity ( ) const
```

/brief Give an access on the error's severity.

/param None.

/return None.

#### 4.9.3.2 what()

```
const char * error::Exception::what ( ) const throw ( )
```

/brief Describe the error.

/param None.

/return the error's message.

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

- inc/exception/[Exception.hh](#)
- src/exception/Exception.cpp

## 4.10 Game Class Reference

The [Game](#) class.

```
#include <Game.hh>
```

### Public Member Functions

- [Game](#) ()  
*Constructor.*
- [~Game](#) ()  
*Destructor.*
- const std::shared\_ptr< [Level](#) > & [getLevel](#) () const  
*Return the level's attribute.*
- void [setLevel](#) (const std::shared\_ptr< [Level](#) > &level)  
*Change the current level.*

#### 4.10.1 Detailed Description

The [Game](#) class.

#### 4.10.2 Constructor & Destructor Documentation

##### 4.10.2.1 Game()

```
Game::Game ( )
```

Constructor.

Build the class.

**Parameters**

<i>None</i>	
-------------	--

**4.10.2.2 ~Game()**

```
Game::~~Game ( )
```

Destructor.

Destroy the class.

**Parameters**

<i>None</i>	
-------------	--

**4.10.3 Member Function Documentation****4.10.3.1 getLevel()**

```
const std::shared_ptr< Level > & Game::getLevel ( ) const
```

Return the level's attribute.

**Parameters**

<i>None</i>	
-------------	--

**Returns**

the current level.

**4.10.3.2 setLevel()**

```
void Game::setLevel (
    const std::shared_ptr< Level > & level )
```

Change the current level.

**Parameters**

<i>The</i>	new level.
------------	------------

**Returns**

None.

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

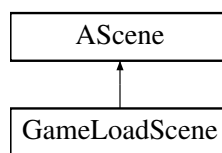
- [inc/game/Game.hh](#)
- [src/game/Game.cpp](#)

## 4.11 GameLoadScene Class Reference

[Button](#) interface of GameLoad scene.

```
#include <GameLoadScene.hh>
```

Inheritance diagram for GameLoadScene:

**Public Member Functions**

- [GameLoadScene](#) ()  
*Constructor.*
- [~GameLoadScene](#) ()  
*Destructor.*
- irr::IEventReceiver \* [getEventReceiver](#) () const  
*Allow the user to implemente the event callback class for each scene.*

### 4.11.1 Detailed Description

[Button](#) interface of GameLoad scene.

### 4.11.2 Constructor & Destructor Documentation



#### 4.11.2.1 GameLoadScene()

```
GameLoadScene::GameLoadScene ( )
```

Constructor.

Build the class.

#### 4.11.2.2 ~GameLoadScene()

```
GameLoadScene::~~GameLoadScene ( )
```

Destructor.

Destroy the class.

### 4.11.3 Member Function Documentation

#### 4.11.3.1 getEventReceiver()

```
irr::IEventReceiver * GameLoadScene::getEventReceiver ( ) const [virtual]
```

Allow the user to implemente the event callback class for each scene.

Return the event.

Implements [AScene](#).

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

- inc/graphics/scenes/[GameLoadScene.hh](#)
- src/graphics/scenes/GameLoadScene.cpp

## 4.12 GameMenuReceiver Class Reference

Manager of event of game scene.

```
#include <GameMenuReceiver.hh>
```

Inherits IEventReceiver.

## Public Member Functions

- [GameMenuReceiver](#) ()  
*Constructor.*
- [~GameMenuReceiver](#) ()  
*Destructor.*
- bool [OnEvent](#) (const irr::SEvent &event)  
*Event manager.*

### 4.12.1 Detailed Description

Manager of event of game scene.

### 4.12.2 Constructor & Destructor Documentation

#### 4.12.2.1 GameMenuReceiver()

```
GameMenuReceiver::GameMenuReceiver ( )
```

Constructor.

Build the class.

#### 4.12.2.2 ~GameMenuReceiver()

```
GameMenuReceiver::~~GameMenuReceiver ( )
```

Destructor.

Destroy the class.

### 4.12.3 Member Function Documentation

#### 4.12.3.1 OnEvent()

```
bool GameMenuReceiver::OnEvent (
    const irr::SEvent & event )
```

Event manager.

Return true if he know the event, false if not.

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

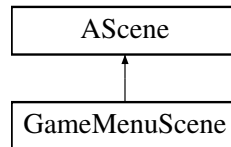
- inc/graphics/scenes/events/[GameMenuReceiver.hh](#)
- src/graphics/scenes/events/GameMenuReceiver.cpp

## 4.13 GameMenuScene Class Reference

[Button](#) interface of menu setting scene.

```
#include <GameMenuScene.hh>
```

Inheritance diagram for GameMenuScene:



### Public Member Functions

- [GameMenuScene](#) ()  
*Constructor.*
- [~GameMenuScene](#) ()  
*Destructor.*
- irr::IEventReceiver \* [getEventReceiver](#) () const  
*Allow the user to implemente the event callback class for each scene.*

### 4.13.1 Detailed Description

[Button](#) interface of menu setting scene.

### 4.13.2 Constructor & Destructor Documentation

#### 4.13.2.1 GameMenuScene()

```
GameMenuScene::GameMenuScene ( )
```

Constructor.

Build the class.

#### 4.13.2.2 ~GameMenuScene()

```
GameMenuScene::~~GameMenuScene ( )
```

Destructor.

Destroy the class.

### 4.13.3 Member Function Documentation

#### 4.13.3.1 `getEventReceiver()`

```
irr::IEventReceiver * GameMenuScene::getEventReceiver ( ) const [virtual]
```

Allow the user to implemente the event callback class for each scene.

Return the event.

Implements [AScene](#).

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

- inc/graphics/scenes/[GameMenuScene.hh](#)
- src/graphics/scenes/GameMenuScene.cpp

## 4.14 GameNewReceiver Class Reference

Manager of event of new game scene.

```
#include <GameNewReceiver.hh>
```

Inherits `IEventReceiver`.

### Public Member Functions

- [GameNewReceiver](#) (const [GameNewScene](#) \*parent)  
*Constructor.*
- [~GameNewReceiver](#) ()  
*Destructor.*
- bool [OnEvent](#) (const irr::SEvent &event)  
*Allow the user to handle envent into a scene.*

#### 4.14.1 Detailed Description

Manager of event of new game scene.

#### 4.14.2 Constructor & Destructor Documentation

#### 4.14.2.1 GameNewReceiver()

```
GameNewReceiver::GameNewReceiver (
    const GameNewScene * parent )
```

Constructor.

Build the class.

#### 4.14.2.2 ~GameNewReceiver()

```
GameNewReceiver::~GameNewReceiver ( )
```

Destructor.

Destroy the class.

### 4.14.3 Member Function Documentation

#### 4.14.3.1 OnEvent()

```
bool GameNewReceiver::OnEvent (
    const irr::SEvent & event )
```

Allow the user to handle event into a scene.

Return true if he know the event, false if not.

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

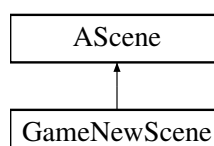
- inc/graphics/scenes/events/[GameNewReceiver.hh](#)
- src/graphics/scenes/events/GameNewReceiver.cpp

## 4.15 GameNewScene Class Reference

[Button](#) interface of new game setting scene.

```
#include <GameNewScene.hh>
```

Inheritance diagram for GameNewScene:



## Public Member Functions

- [GameNewScene](#) ()  
*Constructor.*
- [~GameNewScene](#) ()  
*Destructor.*
- irr::IEventReceiver \* [getEventReceiver](#) () const  
*Allow the user to implemente the event callback class for each scene.*
- const std::vector< std::shared\_ptr< AHero > > & [getPlayers](#) () const  
*Return the players table.*

### 4.15.1 Detailed Description

[Button](#) interface of new game setting scene.

### 4.15.2 Constructor & Destructor Documentation

#### 4.15.2.1 GameNewScene()

```
GameNewScene::GameNewScene ( )
```

Constructor.

Build the class.

#### 4.15.2.2 ~GameNewScene()

```
GameNewScene::~~GameNewScene ( )
```

Destructor.

Destroy the class.

### 4.15.3 Member Function Documentation

#### 4.15.3.1 getEventReceiver()

```
irr::IEventReceiver * GameNewScene::getEventReceiver ( ) const [virtual]
```

Allow the user to implemente the event callback class for each scene.

Return the event.

Implements [AScene](#).

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

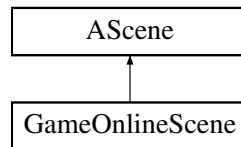
- inc/graphics/scenes/[GameNewScene.hh](#)
- src/graphics/scenes/GameNewScene.cpp

## 4.16 GameOnlineScene Class Reference

[Button](#) interface of GameOnlinescene.

```
#include <GameOnlineScene.hh>
```

Inheritance diagram for GameOnlineScene:



### Public Member Functions

- [GameOnlineScene](#) ()  
*Constructor.*
- [~GameOnlineScene](#) ()  
*Destructor.*
- irr::IEventReceiver \* [getEventReceiver](#) () const  
*Event getter.*

### 4.16.1 Detailed Description

[Button](#) interface of GameOnlinescene.

### 4.16.2 Constructor & Destructor Documentation

#### 4.16.2.1 GameOnlineScene()

```
GameOnlineScene::GameOnlineScene ( )
```

Constructor.

Build the class.

#### 4.16.2.2 ~GameOnlineScene()

```
GameOnlineScene::~~GameOnlineScene ( )
```

Destructor.

Destroy the class.

### 4.16.3 Member Function Documentation

#### 4.16.3.1 `getEventReceiver()`

```
irr::IEventReceiver * GameOnlineScene::getEventReceiver ( ) const [virtual]
```

Event getter.

Return the event.

Implements [AScene](#).

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

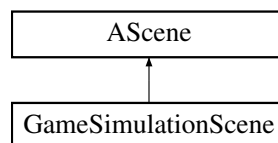
- inc/graphics/scenes/[GameOnlineScene.hh](#)
- src/graphics/scenes/GameOnlineScene.cpp

## 4.17 GameSimulationScene Class Reference

[Button](#) interface of GameSimulationScene.

```
#include <GameSimulationScene.hh>
```

Inheritance diagram for GameSimulationScene:



### Public Member Functions

- [GameSimulationScene](#) (const std::shared\_ptr< [Level](#) > &level)  
*Constructor.*
- [~GameSimulationScene](#) ()  
*Destructor.*
- void [updateCamera](#) ([Window](#) \*win)  
*Update camera's position.*
- irr::IEventReceiver \* [getEventReceiver](#) () const  
*Allow the user to implemente the event callback class for each scene.*

### 4.17.1 Detailed Description

[Button](#) interface of GameSimulationScene.



## 4.17.2 Constructor & Destructor Documentation

### 4.17.2.1 GameSimulationScene()

```
GameSimulationScene::GameSimulationScene (
    const std::shared_ptr< Level > & level )
```

Constructor.

Build the class.

### 4.17.2.2 ~GameSimulationScene()

```
GameSimulationScene::~~GameSimulationScene ( )
```

Destructor.

Destroy the class.

## 4.17.3 Member Function Documentation

### 4.17.3.1 getEventReceiver()

```
irr::IEventReceiver * GameSimulationScene::getEventReceiver ( ) const [virtual]
```

Allow the user to implemente the event callback class for each scene.

Return the event.

Implements [AScene](#).

### 4.17.3.2 updateCamera()

```
void GameSimulationScene::updateCamera (
    Window * win ) [virtual]
```

Update camera's position.

Parameters

<i>The</i>	window who contain the camera.
------------	--------------------------------

Reimplemented from [AScene](#).

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

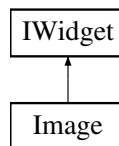
- inc/graphics/scenes/[GameSimulationScene.hh](#)
- src/graphics/scenes/GameSimulationScene.cpp

## 4.18 Image Class Reference

Represent a image in the user interface.

```
#include <Image.hh>
```

Inheritance diagram for Image:



### Public Member Functions

- [Image](#) (const String &name, const Position2d &pos)  
*Constructor.*
- [~Image](#) ()  
*Destructor.*
- void [print](#) ([Window](#) \*win)  
*Print the image into the window.*

### 4.18.1 Detailed Description

Represent a image in the user interface.

### 4.18.2 Constructor & Destructor Documentation

#### 4.18.2.1 Image()

```
Image::Image (
    const String & name,
    const Position2d & pos )
```

Constructor.

Build the class.

## Parameters

1)	'name' the name of the image that will be load and print. 2) 'pos' the position of the image where the image will be print.
----	---

## 4.18.2.2 ~Image()

```
Image::~Image ( )
```

Destructor.

Destroy the class.

## 4.18.3 Member Function Documentation

## 4.18.3.1 print()

```
void Image::print (
    Window * win ) [virtual]
```

Print the image into the window.

## Parameters

'window'	the window where the image will be print.
----------	---

Implements [IWidget](#).

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

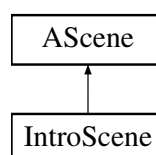
- inc/graphics/widgets/[Image.hh](#)
- src/graphics/widgets/Image.cpp

## 4.19 IntroScene Class Reference

[Button](#) interface of Introcene.

```
#include <IntroScene.hh>
```

Inheritance diagram for IntroScene:



## Public Member Functions

- [IntroScene](#) ()  
*Constructor.*
- [~IntroScene](#) ()  
*Destructor.*
- irr::IEventReceiver \* [getEventReceiver](#) () const  
*Event getter.*

### 4.19.1 Detailed Description

[Button](#) interface of Introcene.

### 4.19.2 Constructor & Destructor Documentation

#### 4.19.2.1 IntroScene()

```
IntroScene::IntroScene ( )
```

Constructor.

Build the class.

#### 4.19.2.2 ~IntroScene()

```
IntroScene::~~IntroScene ( )
```

Destructor.

Destroy the class.

### 4.19.3 Member Function Documentation

#### 4.19.3.1 getEventReceiver()

```
irr::IEventReceiver * IntroScene::getEventReceiver ( ) const [virtual]
```

Event getter.

Return the event.

Implements [AScene](#).

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

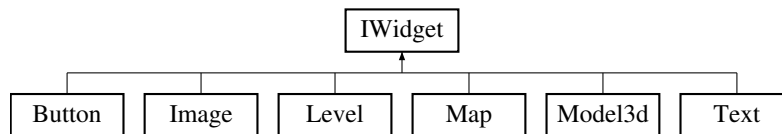
- inc/graphics/scenes/[IntroScene.hh](#)
- src/graphics/scenes/[IntroScene.cpp](#)

## 4.20 IWidget Class Reference

Represent a generic object in the user interface.

```
#include <IWidget.hh>
```

Inheritance diagram for IWidget:



### Public Member Functions

- virtual [~IWidget](#) ()  
*Destructor.*
- virtual void [print](#) ([Window](#) \*win)=0  
*Print the object into the window.*

#### 4.20.1 Detailed Description

Represent a generic object in the user interface.

#### 4.20.2 Constructor & Destructor Documentation

##### 4.20.2.1 ~IWidget()

```
virtual IWidget::~~IWidget ( ) [inline], [virtual]
```

Destructor.

Destroy the class.

#### 4.20.3 Member Function Documentation

##### 4.20.3.1 print()

```
virtual void IWidget::print (
    Window * win ) [pure virtual]
```

Print the object into the window.

## Parameters

<code>'window'</code>	the window where the object will be print.
-----------------------	--

Implemented in [Level](#), [Model3d](#), [Map](#), [Button](#), [Text](#), and [Image](#).

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

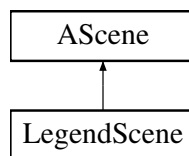
- [inc/graphics/widgets/IWidget.hh](#)

## 4.21 LegendScene Class Reference

[Button](#) interface of [LegendScene](#).

```
#include <LegendScene.hh>
```

Inheritance diagram for LegendScene:



### Public Member Functions

- [LegendScene](#) ()  
*Constructor.*
- [~LegendScene](#) ()  
*Destructor.*
- `irr::IEventReceiver * getEventReceiver () const`  
*Event getter.*

### 4.21.1 Detailed Description

[Button](#) interface of [LegendScene](#).

### 4.21.2 Constructor & Destructor Documentation

#### 4.21.2.1 LegendScene()

```
LegendScene::LegendScene ( )
```

Constructor.

Build the class.

## 4.21.2.2 ~LegendScene()

```
LegendScene::~~LegendScene ( )
```

Destructor.

Destroy the class.

## 4.21.3 Member Function Documentation

## 4.21.3.1 getEventReceiver()

```
irr::IEventReceiver * LegendScene::getEventReceiver ( ) const [virtual]
```

Event getter.

Return the event.

Implements [AScene](#).

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

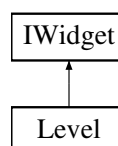
- inc/graphics/scenes/[LegendScene.hh](#)
- src/graphics/scenes/LegendScene.cpp

## 4.22 Level Class Reference

Represent a game's level.

```
#include <Level.hh>
```

Inheritance diagram for Level:



## Public Member Functions

- [Level](#) (const WString &mapPath, std::vector< std::shared\_ptr< AHero > > players=std::vector< std::shared\_ptr< AHero > >(), std::vector< std::shared\_ptr< AMonster > > monsters=std::vector< std::shared\_ptr< AMonster > >(), std::vector< std::shared\_ptr< [APickable](#) > > objects=std::vector< std::shared\_ptr< [APickable](#) > >())  
*Constructor.*
- [~Level](#) ()  
*Destructor.*
- void [movePlayer](#) (const irr::u32 &playerID, const Vector3d &position)  
*Change the player's position by adding the position passed as parameter.*
- void [loadLevel](#) ()
- void [print](#) ([Window](#) \*win)  
*Print the object into the window.*

### 4.22.1 Detailed Description

Represent a game's level.

### 4.22.2 Constructor & Destructor Documentation

#### 4.22.2.1 Level()

```
Level::Level (
    const WString & mapPath,
    std::vector< std::shared_ptr< AHero > > players = std::vector< std::shared_ptr< AHero > > (),
    std::vector< std::shared_ptr< AMonster > > monsters = std::vector< std::shared_ptr< AMonster > > (),
    std::vector< std::shared_ptr< APickable > > objects = std::vector< std::shared_ptr< APickable > > () )
```

Constructor.

Build the class

#### 4.22.2.2 ~Level()

```
Level::~~Level ( )
```

Destructor.

Destroy the class

### 4.22.3 Member Function Documentation

#### 4.22.3.1 loadLevel()

```
void Level::loadLevel ( )
```

load level according to XML file xml is loaded thankfully to levelNumber Loading :

- map
- player
- monsters
- pickable objects

#### 4.22.3.2 movePlayer()

```
void Level::movePlayer (
    const irr::u32 & playerId,
    const Vector3d & position )
```

Change the player's position by adding the position passed as parameter.



## Parameters

1)	The player's id who will move.
2)	The player's id who will move.

## Returns

None.

## 4.22.3.3 print()

```
void Level::print (
    Window * win ) [virtual]
```

Print the object into the window.

## Parameters

'window'	the window where the object will be print.
----------	--

Implements [IWidget](#).

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

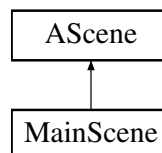
- inc/game/level/[Level.hh](#)
- src/game/level/Level.cpp

## 4.23 MainScene Class Reference

[Button](#) interface of main scene.

```
#include <MainScene.hh>
```

Inheritance diagram for MainScene:



## Public Member Functions

- [MainScene](#) ()  
*Constructor.*
- [~MainScene](#) ()  
*Destructor.*
- irr::IEventReceiver \* [getEventReceiver](#) () const  
*Event getter.*

### 4.23.1 Detailed Description

[Button](#) interface of main scene.

### 4.23.2 Constructor & Destructor Documentation

#### 4.23.2.1 MainScene()

```
MainScene::MainScene ( )
```

Constructor.

Build the class.

#### 4.23.2.2 ~MainScene()

```
MainScene::~~MainScene ( )
```

Destructor.

Destroy the class.

### 4.23.3 Member Function Documentation

#### 4.23.3.1 getEventReceiver()

```
irr::IEventReceiver * MainScene::getEventReceiver ( ) const [virtual]
```

Event getter.

Return the event.

Implements [AScene](#).

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

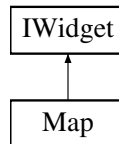
- [inc/graphics/scenes/MainScene.hh](#)
- [src/graphics/scenes/MainScene.cpp](#)

## 4.24 Map Class Reference

Represent a level's map.

```
#include <Map.hh>
```

Inheritance diagram for Map:



### Public Member Functions

- [Map](#) (WString const mapName)  
*Constructs the map with the related file.*
- [~Map](#) ()  
*Destructor...*
- void [print](#) (Window \*win)  
*Print the map into the window, by adding some nodes into the scene manager.*
- WString const [getMapPath](#) (void) const  
*Accessor on map name.*
- bool [loadMap](#) (Window \*win) const  
*Load map in global scene manager.*
- bool [loadSceneNodes](#) (Window \*win)  
*Load all the nodes of the map.*
- bool [initCollisionResponseAnimator](#) (Window \*win)  
*Initialize the collisions.*

### 4.24.1 Detailed Description

Represent a level's map.

### 4.24.2 Constructor & Destructor Documentation

#### 4.24.2.1 Map()

```
Map::Map (
    WString const mapName )
```

Constructs the map with the related file.

## Parameters

<i>mapName</i>	: the related .irr map file
----------------	-----------------------------

## 4.24.3 Member Function Documentation

### 4.24.3.1 loadMap()

```
bool Map::loadMap (
    Window * win ) const
```

Load map in global scene manager.

.irr files can store the whole scene graph including animators, materials and particle systems. And there is also the possibility to store arbitrary user data for every scene node in that file. this example simple, we are simply loading the scene here. See the documentation at `ISceneManager::loadScene` and `ISceneManager::saveScene` for more information.

### 4.24.3.2 loadSceneNodes()

```
bool Map::loadSceneNodes (
    Window * win )
```

Load all the nodes of the map.

We find all the nodes in the scene It would be possible to make a more informed decision about which nodes to perform collision checks on. We would such capture that information in the node name/id.

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

- [inc/game/level/Map.hh](#)
- [src/game/level/Map.cpp](#)

## 4.25 MapFactory Class Reference

Represent a maps' factory.

```
#include <MapFactory.hh>
```

### 4.25.1 Detailed Description

Represent a maps' factory.

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

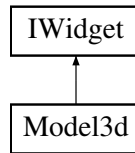
- [inc/game/level/MapFactory.hh](#)
- [src/game/level/MapFactory.cpp](#)

## 4.26 Model3d Class Reference

Represent a model 3d in the user interface.

```
#include <Model3d.hh>
```

Inheritance diagram for Model3d:



### Classes

- struct [Animation](#)  
*Represent a model's animation.*

### Public Member Functions

- [Model3d](#) (const String &fileXML, const Vector3d &position, const Vector3d &rotation, Float scale=1, size\_t animationSpeed=4)  
*Constructor.*
- [~Model3d](#) ()  
*Destructor.*
- void [print](#) ([Window](#) \*win)  
*Print the model 3d into the window.*
- void [setCurrentAnim](#) (const String &name)  
*Allow to change the current animated mesh of the model.*

### 4.26.1 Detailed Description

Represent a model 3d in the user interface.

### 4.26.2 Constructor & Destructor Documentation

#### 4.26.2.1 Model3d()

```

Model3d::Model3d (
    const String & fileXML,
    const Vector3d & position,
    const Vector3d & rotation,
    Float scale = 1,
    size_t animationSpeed = 4 )
  
```

Constructor.

Build the class.

## Parameters

1)	'fileXML' path to the xml file who describe the model. 2) 'position' the position in the space where the model will be print. 3) 'rotation' the rotation of the model in the space. 4) 'scale' an additionnal scale coefficient for the model.
----	--

## 4.26.2.2 ~Model3d()

```
Model3d::~Model3d ( )
```

Destructor.

Destroy the class.

## 4.26.3 Member Function Documentation

## 4.26.3.1 print()

```
void Model3d::print (
    Window * win ) [virtual]
```

Print the model 3d into the window.

## Parameters

'window'	the window where the model 3d will be print.
----------	--

Implements [IWidget](#).

## 4.26.3.2 setCurrentAnim()

```
void Model3d::setCurrentAnim (
    const String & name )
```

Allow to change the current animated mesh of the model.

## Parameters

'name'	of the animation that will be load.
--------	-------------------------------------

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

- [inc/graphics/widgets/Model3d.hh](#)
- [src/graphics/widgets/Model3d.cpp](#)

## 4.27 ModelsManager Class Reference

The class to load all the information about a model.

```
#include <ModelsManager.hh>
```

### Public Member Functions

- [ModelsManager](#) ()  
*Constructor.*
- bool [loadModel](#) (String, String)  
*Allow you to load a model.*
- const config::Model \* [getModel](#) (String) const  
*Give you a pointeur to an object Model.*

### 4.27.1 Detailed Description

The class to load all the information about a model.

### 4.27.2 Constructor & Destructor Documentation

#### 4.27.2.1 ModelsManager()

```
ModelsManager::ModelsManager ( )
```

Constructor.

Parameters

<i>nothing</i>	
----------------	--

### 4.27.3 Member Function Documentation

#### 4.27.3.1 getModel()

```
const config::Model * ModelsManager::getModel (
    String toFind ) const
```

Give you a pointeur to an object Model.

#### Parameters

<i>The</i>	name previously given to the model
------------	------------------------------------

#### Returns

A pointer to the model

#### 4.27.3.2 loadModel()

```
bool ModelsManager::loadModel (
    String path,
    String modelName )
```

Allow you to load a model.

#### Parameters

<i>the</i>	path to the xml configuration and a name
------------	--

#### Returns

true is the model is load and false otherwise

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

- inc/managers/[ModelsManager.hh](#)
- src/managers/ModelsManager.cpp

## 4.28 OptionControl Class Reference

[Button](#) interface of OptionControlscene.

```
#include <OptionControlScene.hh>
```

### 4.28.1 Detailed Description

[Button](#) interface of OptionControlscene.

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

- inc/graphics/scenes/[OptionControlScene.hh](#)

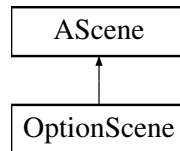


## 4.29 OptionScene Class Reference

[Button](#) interface of option scene.

```
#include <OptionScene.hh>
```

Inheritance diagram for OptionScene:



### Public Member Functions

- [OptionScene](#) ()  
*Constructor.*
- [~OptionScene](#) ()  
*Destructor.*
- irr::IEventReceiver \* [getEventReceiver](#) () const  
*Event getter.*

### 4.29.1 Detailed Description

[Button](#) interface of option scene.

### 4.29.2 Constructor & Destructor Documentation

#### 4.29.2.1 OptionScene()

```
OptionScene::OptionScene ( )
```

Constructor.

Build the class.

#### 4.29.2.2 ~OptionScene()

```
OptionScene::~~OptionScene ( )
```

Destructor.

Destroy the class.

### 4.29.3 Member Function Documentation

#### 4.29.3.1 `getEventReceiver()`

```
irr::IEventReceiver * OptionScene::getEventReceiver ( ) const [virtual]
```

Event getter.

Return the event.

Implements [AScene](#).

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

- `inc/graphics/scenes/OptionScene.hh`
- `src/graphics/scenes/OptionScene.cpp`

## 4.30 OptionSoundReceiver Class Reference

Manager of event of Sound scene setting.

```
#include <OptionSoundReceiver.hh>
```

Inherits `IEventReceiver`.

### Public Member Functions

- [OptionSoundReceiver](#) ([OptionSoundScene](#) \*)  
*Constructor.*
- [~OptionSoundReceiver](#) ()  
*Destructor.*
- `bool` [OnEvent](#) (`const irr::SEvent &event`)  
*Allow the user to handle event into a scene.*

#### 4.30.1 Detailed Description

Manager of event of Sound scene setting.

#### 4.30.2 Constructor & Destructor Documentation

#### 4.30.2.1 OptionSoundReceiver()

```
OptionSoundReceiver::OptionSoundReceiver (
    OptionSoundScene * scene )
```

Constructor.

Build the class.

#### 4.30.2.2 ~OptionSoundReceiver()

```
OptionSoundReceiver::~~OptionSoundReceiver ( )
```

Destructor.

Destroy the class.

### 4.30.3 Member Function Documentation

#### 4.30.3.1 OnEvent()

```
bool OptionSoundReceiver::OnEvent (
    const irr::SEvent & event )
```

Allow the user to handle event into a scene.

Return true if he know the event, false if not.

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

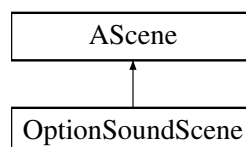
- inc/graphics/scenes/events/[OptionSoundReceiver.hh](#)
- src/graphics/scenes/events/OptionSoundReceiver.cpp

## 4.31 OptionSoundScene Class Reference

[Button](#) interface of sound setting scene.

```
#include <OptionSoundScene.hh>
```

Inheritance diagram for OptionSoundScene:



## Public Member Functions

- [OptionSoundScene](#) ()  
*Constructor.*
- [~OptionSoundScene](#) ()  
*Destructor.*
- irr::IEventReceiver \* [getEventReceiver](#) () const  
*Allow the user to implemente the event callback class for each scene.*

### 4.31.1 Detailed Description

[Button](#) interface of sound setting scene.

### 4.31.2 Constructor & Destructor Documentation

#### 4.31.2.1 OptionSoundScene()

```
OptionSoundScene::OptionSoundScene ( )
```

Constructor.

Build the class.

#### 4.31.2.2 ~OptionSoundScene()

```
OptionSoundScene::~~OptionSoundScene ( )
```

Destructor.

Destroy the class.

### 4.31.3 Member Function Documentation

#### 4.31.3.1 getEventReceiver()

```
irr::IEventReceiver * OptionSoundScene::getEventReceiver ( ) const [virtual]
```

Allow the user to implemente the event callback class for each scene.

Return the event's receiver.

Implements [AScene](#).

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

- inc/graphics/scenes/[OptionSoundScene.hh](#)
- src/graphics/scenes/[OptionSoundScene.cpp](#)

## 4.32 OptionVideoReceiver Class Reference

Manager of event of video setting scene.

```
#include <OptionVideoReceiver.hh>
```

Inherits IEventReceiver.

### Public Member Functions

- [OptionVideoReceiver](#) ()  
*Constructor.*
- [~OptionVideoReceiver](#) ()  
*Destructor.*
- bool [OnEvent](#) (const irr::SEvent &event)  
*Event manager.*

### 4.32.1 Detailed Description

Manager of event of video setting scene.

### 4.32.2 Constructor & Destructor Documentation

#### 4.32.2.1 OptionVideoReceiver()

```
OptionVideoReceiver::OptionVideoReceiver ( )
```

Constructor.

Build the class.

#### 4.32.2.2 ~OptionVideoReceiver()

```
OptionVideoReceiver::~~OptionVideoReceiver ( )
```

Destructor.

Destroy the class.

### 4.32.3 Member Function Documentation

#### 4.32.3.1 OnEvent()

```
bool OptionVideoReceiver::OnEvent (
    const irr::SEvent & event )
```

Event manager.

Return true if he know the event, false if not.

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

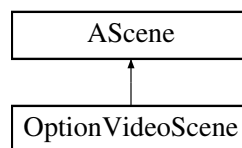
- [inc/graphics/scenes/events/OptionVideoReceiver.hh](#)
- [src/graphics/scenes/events/OptionVideoReceiver.cpp](#)

### 4.33 OptionVideoScene Class Reference

[Button](#) interface of video setting scene.

```
#include <OptionVideoScene.hh>
```

Inheritance diagram for OptionVideoScene:



#### Public Member Functions

- [OptionVideoScene \(\)](#)  
*Constructor.*
- [~OptionVideoScene \(\)](#)  
*Destructor.*
- `irr::IEventReceiver * getEventReceiver \(\) const`  
*Event getter.*

#### 4.33.1 Detailed Description

[Button](#) interface of video setting scene.

#### 4.33.2 Constructor & Destructor Documentation

#### 4.33.2.1 OptionVideoScene()

```
OptionVideoScene::OptionVideoScene ( )
```

Constructor.

Build the class.

#### 4.33.2.2 ~OptionVideoScene()

```
OptionVideoScene::~~OptionVideoScene ( )
```

Destructor.

Destroy the class.

### 4.33.3 Member Function Documentation

#### 4.33.3.1 getEventReceiver()

```
irr::IEventReceiver * OptionVideoScene::getEventReceiver ( ) const [virtual]
```

Event getter.

Return the event.

Implements [AScene](#).

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

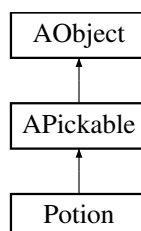
- inc/graphics/scenes/[OptionVideoScene.hh](#)
- src/graphics/scenes/OptionVideoScene.cpp

## 4.34 Potion Class Reference

create a [APickable AObject](#) key

```
#include <Potion.hh>
```

Inheritance diagram for Potion:



## Public Member Functions

- [Potion](#) (irr::s32 regenPoint, indie::potionType type=indie::potionType::LIFE, irr::u32 id=0)  
*Constructor.*
- [~Potion](#) ()  
*Destructor.*

### 4.34.1 Detailed Description

create a [APickable AObject](#) key

### 4.34.2 Constructor & Destructor Documentation

#### 4.34.2.1 Potion()

```
Potion::Potion (
    irr::s32 regenPoint,
    indie::potionType type = indie::potionType::LIFE,
    irr::u32 id = 0 )
```

Constructor.

Build the class, first param is number of regenPoint, second the type of potion (LIFE, MANA, POWER)

#### 4.34.2.2 ~Potion()

```
Potion::~~Potion ( )
```

Destructor.

Destroy a class

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

- inc/game/object/Potion.hh
- src/game/object/Potion.cpp

## 4.35 SaveManager Class Reference

The [SaveManager](#) class.

```
#include <SaveManager.hh>
```



## Public Member Functions

- [SaveManager](#) ()  
*The constructor.*
- `std::shared_ptr< Level > loadLevel` (const String &levelName)  
*Allow you to load a game previously save.*
- `void saveLevel` (const std::shared\_ptr< [Level](#) > &level, const String &fileName)  
*Allow you to save the object [Game](#) passed as parameter during the construction.*
- `const std::shared_ptr< Game > & getGame` () const  
*Getter on the game's variable.*

### 4.35.1 Detailed Description

The [SaveManager](#) class.

### 4.35.2 Member Function Documentation

#### 4.35.2.1 loadLevel()

```
std::shared_ptr< Level > SaveManager::loadLevel (
    const String & levelName )
```

Allow you to load a game previously save.

#### Parameters

<i>The</i>	path to the game's file
------------	-------------------------

#### Returns

A pointeur to a [Game](#) object

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

- inc/managers/[SaveManager.hh](#)
- src/managers/SaveManager.cpp

## 4.36 ScoreManager Class Reference

The [ScoreManager](#) class.

```
#include <ScoreManager.hh>
```

## Public Member Functions

- [ScoreManager](#) ()  
*The constructor.*
- void [loadScores](#) (std::string)  
*Allow you to load a score sheet (from a previous session)*
- void [saveScores](#) () const  
*Allow you to save the score contained in the object.*
- void [createFile](#) (std::string)  
*Create an empty score file.*
- void [addScore](#) (std::string, std::size\_t)  
*Add a score.*
- const std::vector< config::Score > & [getTopScores](#) ()  
*Get the 10 best scores.*
- std::size\_t [getPlayerScore](#) (std::string) const  
*Get score for a specifique player.*
- void [changeScore](#) (std::string, std::size\_t)  
*Change score for a specifique player.*

### 4.36.1 Detailed Description

The [ScoreManager](#) class.

### 4.36.2 Constructor & Destructor Documentation

#### 4.36.2.1 ScoreManager()

```
ScoreManager::ScoreManager ( )
```

The constructor.

Parameters

<i>Nothing</i>	
----------------	--

### 4.36.3 Member Function Documentation

#### 4.36.3.1 addScore()

```
void ScoreManager::addScore (
    std::string playerName,
    std::size_t playerScore )
```

Add a score.

**Parameters**

1.	Player name 2. Is score
----	-------------------------

**4.36.3.2 changeScore()**

```
void ScoreManager::changeScore (
    std::string ,
    std::size_t )
```

Change score for a specifiqu player.

**Parameters**

1.	Player name 2. New score
----	--------------------------

**4.36.3.3 createFile()**

```
void ScoreManager::createFile (
    std::string fileName )
```

Create an empty score file.

**Parameters**

<i>Path</i>	to the score file
-------------	-------------------

**4.36.3.4 getPlayerScore()**

```
std::size_t ScoreManager::getPlayerScore (
    std::string name ) const
```

Get score for a specifiqu player.

**Parameters**

<i>Player</i>	name
---------------	------

## 4.36.3.5 loadScores()

```
void ScoreManager::loadScores (
    std::string fileName )
```

Allow you to load a score sheet (from a previous session)

## Parameters

<i>The</i>	path to the score file
------------	------------------------

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

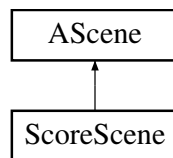
- inc/managers/ScoreManager.hh
- src/managers/ScoreManager.cpp

## 4.37 ScoreScene Class Reference

[Button](#) interface of [ScoreScene](#).

```
#include <ScoreScene.hh>
```

Inheritance diagram for ScoreScene:



## Public Member Functions

- [ScoreScene](#) ()  
*Constructor.*
- [~ScoreScene](#) ()  
*Destructor.*
- irr::IEventReceiver \* [getEventReceiver](#) () const  
*Event getter.*
- WString [to\\_WString](#) (std::string) const  
*to\_Wstring*
- WString [getScores](#) () const  
*Event getter.*

## 4.37.1 Detailed Description

[Button](#) interface of [ScoreScene](#).

## 4.37.2 Constructor & Destructor Documentation

### 4.37.2.1 ScoreScene()

```
ScoreScene::ScoreScene ( )
```

Constructor.

Build the class.

### 4.37.2.2 ~ScoreScene()

```
ScoreScene::~~ScoreScene ( )
```

Destructor.

Destroy the class.

## 4.37.3 Member Function Documentation

### 4.37.3.1 getEventReceiver()

```
irr::IEventReceiver * ScoreScene::getEventReceiver ( ) const [virtual]
```

Event getter.

Return the event.

Implements [AScene](#).

### 4.37.3.2 getScores()

```
WString ScoreScene::getScores ( ) const
```

Event getter.

Return the Score.

## 4.37.3.3 to\_WString()

```
WString ScoreScene::to_WString (
    std::string txt ) const
```

to\_Wstring

exchange a string to a Wstring for irrlicht

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

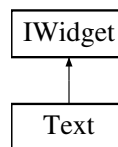
- inc/graphics/scenes/ScoreScene.hh
- src/graphics/scenes/ScoreScene.cpp

## 4.38 Text Class Reference

Inherited from a [IWidget](#).

```
#include <Text.hh>
```

Inheritance diagram for Text:



## Public Member Functions

- [Text](#) (WString text, Rect pos, std::string fontName="chow\_fun", std::size\_t fontSize=96)  
*Constructor.*
- void [print](#) ([Window](#) \*win)  
*Display the text box on the screen.*

## 4.38.1 Detailed Description

Inherited from a [IWidget](#).

## 4.38.2 Constructor &amp; Destructor Documentation

## 4.38.2.1 Text()

```
Text::Text (
    WString text,
    Rect pos,
    std::string fontName = "chow_fun",
    std::size_t fontSize = 96 )
```

Constructor.

**Parameters**

<i>text</i>	The data
<i>pos</i>	The position and the size of the text box
<i>fontName</i>	The font name to be used
<i>fontSize</i>	The size of the font (8, 9, 10, 11, 12, 14, 18, 24, 30, 36, 48, 60, 72, 96)

**4.38.3 Member Function Documentation****4.38.3.1 print()**

```
void Text::print (
    Window * win ) [virtual]
```

Display the text box on the screen.

**Parameters**

<i>win</i>	The window
------------	------------

Implements [IWidget](#).

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

- inc/graphics/widgets/[Text.hh](#)
- src/graphics/widgets/Text.cpp

**4.39 Window Class Reference**

Represent the window that will print some scene (view of the application).

```
#include <Window.hh>
```

**Public Member Functions**

- [Window](#) (const wchar\_t \*windowName)  
*Constructor.*
- [~Window](#) ()  
*Destructor.*
- void [addScene](#) (const String &name, const std::shared\_ptr< [AScene](#) > &scene)  
*Allow the user to add a scene in the window.*
- void [printScene](#) (const String &name)  
*Allow the user to print a scene in the window.*



- void `deleteScene` (const String &name)  
*Allow the user ti delete a scene in the window.*
- void `setCurrentScene` (const String &sceneName)  
*Set the current scene that will be print.*
- void `changeScene` ()  
*Apply the setCurrentScene's changes.*
- void `close` ()  
*Close the window.*
- bool `isOpen` () const  
*Getter on the open's statement of the window.*
- void `loop` (const String &sceneName)  
*Keep the window open util the end of the program's execution.*

### 4.39.1 Detailed Description

Represent the window that will print some scene (view of the application).

### 4.39.2 Constructor & Destructor Documentation

#### 4.39.2.1 Window()

```
Window::Window (
    const wchar_t * windowName )
```

Constructor.

Build the class.

#### Parameters

'windowName'	the window name that will be print on the top of the window.
--------------	--

#### 4.39.2.2 ~Window()

```
Window::~~Window ( )
```

Destructor.

Destroy the class.

### 4.39.3 Member Function Documentation

#### 4.39.3.1 addScene()

```
void Window::addScene (
    const String & name,
    const std::shared_ptr< AScene > & scene )
```

Allow the user to add a scene in the window.

##### Parameters

<i>name</i>	The name of the scene that will be add.
<i>scene</i>	Reference to a smart pointer on the scene that will be add.

#### 4.39.3.2 deleteScene()

```
void Window::deleteScene (
    const String & name )
```

Allow the user ti delete a scene in the window.

##### Parameters

<i>'name'</i>	the name of the scene that will be delete.
---------------	--

#### 4.39.3.3 isOpen()

```
bool Window::isOpen ( ) const
```

Getter on the open's statement of the window.

##### Returns

True if the window is open, else false.

#### 4.39.3.4 loop()

```
void Window::loop (
    const String & sceneName )
```

Keep the window open until the end of the program's execution.

## Parameters

<code>'sceneName'</code>	the scene's name that will be launch at the begin of the program's execution.
--------------------------	---

## 4.39.3.5 printScene()

```
void Window::printScene (  
    const String & name )
```

Allow the user to print a scene in the window.

## Parameters

<code>'name'</code>	the name of the scene that will be print.
---------------------	---

## 4.39.3.6 setCurrentScene()

```
void Window::setCurrentScene (  
    const String & sceneName )
```

Set the current scene that will be print.

## Parameters

<code>'sceneName'</code>	the name of the scene that will be set as the current scene.
--------------------------	--

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

- [inc/graphics/Window.hh](#)
- [src/graphics/Window.cpp](#)



## Chapter 5

# File Documentation

### 5.1 inc/exception/Exception.hh File Reference

Exception class.

```
#include <exception>
#include <sstream>
#include <string>
```

#### Classes

- class [error::Exception](#)  
*This class allow to throw some execption and handle it finely.*

#### 5.1.1 Detailed Description

Exception class.

#### Author

Théophile Champion

#### Version

1.0

### 5.2 inc/game/Game.hh File Reference

Contain all the information about the game.

```
#include <memory>
#include "Level.hh"
```

## Classes

- class [Game](#)  
*The [Game](#) class.*

### 5.2.1 Detailed Description

Contain all the information about the game.

#### Author

Théophile Champion

#### Version

1.0

## 5.3 inc/game/level/Level.hh File Reference

Use to create a level of the game.

```
#include <irrlicht/irrlicht.h>
#include <vector>
#include <memory>
#include <string>
#include "MapFactory.hh"
#include "Map.hh"
```

## Classes

- class [Level](#)  
*Represent a game's level.*

### 5.3.1 Detailed Description

Use to create a level of the game.

#### Author

Théophile Champion

#### Version

1.0

## 5.4 inc/game/level/Map.hh File Reference

Defines the map that is related to a level. The process to create a map is relatively simple. First of all, you've got to load a map with [Map::loadMap\(\)](#) In a second time, you've got to retrieve all the nodes with a triangle selector container (meta) with `loadSceneNodes` Then, you can create collision response thanks to these nodes by calling `initCollisionResponseAnimator()` Finally, you can draw the map (that should be achieved with `drawAll()`) bugs: If meshes/models aren't loading properly. Double check the paths in the .irr file.

```
#include <iostream>
#include <string>
#include "graphics_header.hh"
#include "IWidget.hh"
```

### Classes

- class [Map](#)  
*Represent a level's map.*

#### 5.4.1 Detailed Description

Defines the map that is related to a level. The process to create a map is relatively simple. First of all, you've got to load a map with [Map::loadMap\(\)](#) In a second time, you've got to retrieve all the nodes with a triangle selector container (meta) with `loadSceneNodes` Then, you can create collision response thanks to these nodes by calling `initCollisionResponseAnimator()` Finally, you can draw the map (that should be achieved with `drawAll()`) bugs: If meshes/models aren't loading properly. Double check the paths in the .irr file.

#### Author

Charles Paulet

#### Version

0.1

## 5.5 inc/game/level/MapFactory.hh File Reference

Maps' factory use to build the maps objects.

```
#include <memory>
#include "Map.hh"
```

### Classes

- class [MapFactory](#)  
*Represent a maps' factory.*

### 5.5.1 Detailed Description

Maps' factory use to build the maps objects.

#### Author

Charles Paulet

#### Version

0.1

## 5.6 inc/game/object/AObject.hh File Reference

Use to create object.

```
#include <irrlicht/irrlicht.h>
#include "Model3d.hh"
```

### Classes

- class [AObject](#)

*An [AObject](#) instance has his own [Model3d](#) attribute, and can modify it..*

### 5.6.1 Detailed Description

Use to create object.

#### Author

Hugo Ailleres

#### Version

1.0

## 5.7 inc/game/object/APickable.hh File Reference

Use to create pickable object.

```
#include "AObject.hh"
```



## Classes

- class [APickable](#)  
*Create a pickable [AObject](#).*

### 5.7.1 Detailed Description

Use to create pickable object.

Use to create a key [APickable AObject](#).

#### Author

Hugo Ailleres

#### Version

1.0

## 5.8 inc/graphics/AScene.hh File Reference

Define a window's scene.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include <vector>
#include "type_compatibility.hh"
```

## Classes

- class [AScene](#)  
*Represent the window's scene that will print some widget/object of the scene : buttons, images, etc...*

### 5.8.1 Detailed Description

Define a window's scene.

#### Author

Théophile Champion

#### Version

1.0

## 5.9 inc/graphics/Camera.hh File Reference

Defines the camera, its type and behaviour bugs: Actually you have to give the constructor the camera. That may change soon with integration on indie\_studio This way, cam will be created inside cosntructor with the g\_Window->sceneManager that would be such accessible.

### Classes

- class [Camera](#)  
*where cameras behaviour is defined*

#### 5.9.1 Detailed Description

Defines the camera, its type and behaviour bugs: Actually you have to give the constructor the camera. That may change soon with integration on indie\_studio This way, cam will be created inside cosntructor with the g\_Window->sceneManager that would be such accessible.

[Camera](#) positionning may be set using Maths::getCentroid as position

#### Author

Charles Paulet

#### Version

0.1

## 5.10 inc/graphics/scenes/events/GameMenuReceiver.hh File Reference

Use to manage event of the game (new game / load) scene.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "AScene.hh"
```

### Classes

- class [GameMenuReceiver](#)  
*Manager of event of game scene.*

#### 5.10.1 Detailed Description

Use to manage event of the game (new game / load) scene.

#### Author

Leo PAOL

#### Version

1.0

## 5.11 inc/graphics/scenes/events/GameNewReceiver.hh File Reference

Use to manage event of the new game (character selection) scene.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "AScene.hh"
```

### Classes

- class [GameNewReceiver](#)  
*Manager of event of new game scene.*

#### 5.11.1 Detailed Description

Use to manage event of the new game (character selection) scene.

##### Author

Leo PAOL

##### Version

1.0

## 5.12 inc/graphics/scenes/events/OptionSoundReceiver.hh File Reference

Allow to manage event of Sound scene setting.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "AScene.hh"
```

### Classes

- class [OptionSoundReceiver](#)  
*Manager of event of Sound scene setting.*

#### 5.12.1 Detailed Description

Allow to manage event of Sound scene setting.

##### Author

Hugo Ailleres

##### Version

1.0

## 5.13 inc/graphics/scenes/events/OptionVideoReceiver.hh File Reference

Use to manage event of the video setting scene.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "AScene.hh"
```

### Classes

- class [OptionVideoReceiver](#)  
*Manager of event of video setting scene.*

#### 5.13.1 Detailed Description

Use to manage event of the video setting scene.

##### Author

Leo PAOL

##### Version

1.0

## 5.14 inc/graphics/scenes/GameLoadScene.hh File Reference

Use to display the gameload scene.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "graphics_header.hh"
```

### Classes

- class [GameLoadScene](#)  
*[Button](#) interface of GameLoad scene.*

#### 5.14.1 Detailed Description

Use to display the gameload scene.

##### Author

Clovis Peridy

##### Version

1.0

## 5.15 inc/graphics/scenes/GameMenuScene.hh File Reference

Use to choose between new game and load game.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "graphics_header.hh"
```

### Classes

- class [GameMenuScene](#)  
*Button interface of menu setting scene.*

#### 5.15.1 Detailed Description

Use to choose between new game and load game.

##### Author

Leo PAOL

##### Version

1.0

## 5.16 inc/graphics/scenes/GameNewScene.hh File Reference

Use to choose characters for each player.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "graphics_header.hh"
```

### Classes

- class [GameNewScene](#)  
*Button interface of new game setting scene.*

#### 5.16.1 Detailed Description

Use to choose characters for each player.

##### Author

Léo PAOL

##### Version

1.0

## 5.17 inc/graphics/scenes/GameOnlineScene.hh File Reference

Use to display the GameOnline scene.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "graphics_header.hh"
```

### Classes

- class [GameOnlineScene](#)  
*[Button](#) interface of GameOnlinescene.*

### 5.17.1 Detailed Description

Use to display the GameOnline scene.

#### Author

Clovis Peridy

#### Version

1.0

## 5.18 inc/graphics/scenes/GameScene.hh File Reference

Use to display the [Game](#) scene.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "graphics_header.hh"
```

### 5.18.1 Detailed Description

Use to display the [Game](#) scene.

#### Author

Clovis Peridy

#### Version

1.0

## 5.19 inc/graphics/scenes/GameSimulationScene.hh File Reference

Use to display the GameOnline scene.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "graphics_header.hh"
```

### Classes

- class [GameSimulationScene](#)  
*[Button](#) interface of GameSimulationscene.*

#### 5.19.1 Detailed Description

Use to display the GameOnline scene.

##### Author

Clovis Peridy

##### Version

1.0

## 5.20 inc/graphics/scenes/IntroScene.hh File Reference

Use to display the [IntroScene](#) scene.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "graphics_header.hh"
```

### Classes

- class [IntroScene](#)  
*[Button](#) interface of Introcene.*

#### 5.20.1 Detailed Description

Use to display the [IntroScene](#) scene.

##### Author

Clovis Peridy

##### Version

1.0

## 5.21 inc/graphics/scenes/LegendScene.hh File Reference

Use to display the Legend scene.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "graphics_header.hh"
```

### Classes

- class [LegendScene](#)  
*Button interface of [LegendScene](#).*

#### 5.21.1 Detailed Description

Use to display the Legend scene.

##### Author

Clovis Peridy

##### Version

1.0

## 5.22 inc/graphics/scenes/MainScene.hh File Reference

Main scene who lead to each menus.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "graphics_header.hh"
```

### Classes

- class [MainScene](#)  
*Button interface of main scene.*

#### 5.22.1 Detailed Description

Main scene who lead to each menus.

##### Author

Clovis Peridy

##### Version

1.0



## 5.23 inc/graphics/scenes/OptionControlScene.hh File Reference

Use to display the [OptionControl](#) scene.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "graphics_header.hh"
```

### 5.23.1 Detailed Description

Use to display the [OptionControl](#) scene.

#### Author

Clovis Peridy

#### Version

1.0

## 5.24 inc/graphics/scenes/OptionScene.hh File Reference

Use to display the option menu.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "graphics_header.hh"
```

### Classes

- class [OptionScene](#)  
*[Button](#) interface of option scene.*

### 5.24.1 Detailed Description

Use to display the option menu.

#### Author

Clovis Peridy

#### Version

1.0

## 5.25 inc/graphics/scenes/OptionSoundScene.hh File Reference

Use to create the button interface of the sound setting scene.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "graphics_header.hh"
```

### Classes

- class [OptionSoundScene](#)  
*Button interface of sound setting scene.*

#### 5.25.1 Detailed Description

Use to create the button interface of the sound setting scene.

##### Author

Hugo Ailleres

##### Version

1.0

## 5.26 inc/graphics/scenes/OptionVideoScene.hh File Reference

Use to create the button interface of the video setting scene.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "graphics_header.hh"
```

### Classes

- class [OptionVideoScene](#)  
*Button interface of video setting scene.*

#### 5.26.1 Detailed Description

Use to create the button interface of the video setting scene.

##### Author

Leo PAOL

##### Version

1.0

## 5.27 inc/graphics/scenes/ScoreScene.hh File Reference

Use to display the Score scene.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include "ScoreManager.hh"
#include "graphics_header.hh"
#include "Text.hh"
```

### Classes

- class [ScoreScene](#)  
*Button interface of [ScoreScene](#).*

#### 5.27.1 Detailed Description

Use to display the Score scene.

#### Author

Clovis Peridy

#### Version

1.0

## 5.28 inc/graphics/widgets/Button.hh File Reference

Allow to add some button in the window.

```
#include "type_compatibility.hh"
#include "graphics_header.hh"
#include "GUIButtonId.hh"
```

### Classes

- class [Button](#)  
*Represent a button in the user interface.*

### 5.28.1 Detailed Description

Allow to add some button in the window.

#### Author

Théophile Champion

#### Version

1.0

## 5.29 inc/graphics/widgets/Image.hh File Reference

Allow to add some image in the window.

```
#include <irrlicht/irrlicht.h>
#include "type_compatibility.hh"
#include "graphics_header.hh"
```

### Classes

- class [Image](#)  
*Represent a image in the user interface.*

### 5.29.1 Detailed Description

Allow to add some image in the window.

#### Author

Théophile Champion

#### Version

1.0

## 5.30 inc/graphics/widgets/IWidget.hh File Reference

Give a interface for all the class that could be print on the window.

```
#include "graphics_header.hh"
```

## Classes

- class [IWidget](#)

*Represent a generic object in the user interface.*

### 5.30.1 Detailed Description

Give a interface for all the class that could be print on the window.

#### Author

Théophile Champion

#### Version

1.0

## 5.31 inc/graphics/widgets/Model3d.hh File Reference

Allow to add some model 3d in the window.

```
#include <vector>
#include <string>
#include "type_compatibility.hh"
#include "graphics_header.hh"
```

## Classes

- class [Model3d](#)

*Represent a model 3d in the user interface.*

- struct [Model3d::Animation](#)

*Represent a model's animation.*

### 5.31.1 Detailed Description

Allow to add some model 3d in the window.

#### Author

Théophile Champion

#### Version

1.0

## 5.32 inc/graphics/widgets/Text.hh File Reference

Allow you to create a text object.

```
#include "type_compatibility.hh"
#include "graphics_header.hh"
#include "GUIButtonId.hh"
#include "CoreManager.hh"
#include "IWidget.hh"
```

### Classes

- class [Text](#)

*Inherited from a [IWidget](#).*

### 5.32.1 Detailed Description

Allow you to create a text object.

#### Author

Nicolas-Emmanuel Robert

#### Version

1.0

## 5.33 inc/graphics/Window.hh File Reference

Define a window.

```
#include <irrlicht/irrlicht.h>
#include <memory>
#include <string>
#include <vector>
#include "type_compatibility.hh"
#include "CoreManager.hh"
```

### Classes

- class [Window](#)

*Represent the window that will print some scene (view of the application).*

### 5.33.1 Detailed Description

Define a window.

Author

Théophile Champion

Version

1.0

## 5.34 inc/managers/config/Model.hh File Reference

The Class Model allow you to manipulate model's informations.

```
#include <vector>
#include "type_compatibility.hh"
```

### 5.34.1 Detailed Description

The Class Model allow you to manipulate model's informations.

The Class Video allow you to manipulate video's configuration.

The Class Sound allow you to manipulate sound's configuration.

The Class Score allow you to manipulate score's informations.

Author

Nicolas-Emmanuel Robert

Version

1.0

## 5.35 inc/managers/ConfigManager.hh File Reference

Allow you to load, modify and save the configuration.

```
#include <irrlicht/irrlicht.h>
#include <irrlicht/irrXML.h>
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
#include <map>
#include "Control.hh"
#include "Video.hh"
#include "Sound.hh"
#include "Exception.hh"
#include "IControl.hh"
```

## Classes

- class [ConfigManager](#)

*This class contain all the methode to manipulate the configuration.*

### 5.35.1 Detailed Description

Allow you to load, modify and save the configuration.

#### Author

Nicolas-Emmanuel Robert

#### Version

1.0

## 5.36 inc/managers/CoreManager.hh File Reference

[CoreManager](#) regroups all the application's manager.

```
#include "ModelsManager.hh"
#include "ConfigManager.hh"
#include "ScoreManager.hh"
#include "LevelManager.hh"
#include "SaveManager.hh"
#include "EventManager.hh"
```

## Classes

- class [CoreManager](#)

*[CoreManager](#) is the main manager who regroups all the other manager.*

### 5.36.1 Detailed Description

[CoreManager](#) regroups all the application's manager.

#### Author

Théophile Champion

#### Version

1.0



## 5.37 inc/managers/LevelManager.hh File Reference

Load level configuration out of XML format.

```
#include <vector>
#include <string>
#include <iostream>
#include <fstream>
#include <irrlicht/irrlicht.h>
#include <irrlicht/irrXML.h>
#include "LevelDefinition.hh"
#include "Exception.hh"
```

### 5.37.1 Detailed Description

Load level configuration out of XML format.

#### Author

Charles && Clovis

#### Version

1.0

## 5.38 inc/managers/ModelsManager.hh File Reference

The Models manager will allow you to get all the information to load a model.

```
#include <irrlicht/irrlicht.h>
#include <irrlicht/irrXML.h>
#include <iostream>
#include <fstream>
#include <map>
#include "type_compatibility.hh"
#include "Exception.hh"
#include "Model.hh"
```

### Classes

- class [ModelsManager](#)

*The class to load all the information about a model.*

### 5.38.1 Detailed Description

The Models manager will allow you to get all the information to load a model.

#### Author

Nicolas-Emmanuel Robert

#### Version

1.1

## 5.39 inc/managers/SaveManager.hh File Reference

The save manager will allow you to save and restore a game.

```
#include <memory>
#include "type_compatibility.hh"
```

### Classes

- class [SaveManager](#)  
The *SaveManager* class.

### 5.39.1 Detailed Description

The save manager will allow you to save and restore a game.

#### Author

Nicolas-Emmanuel Robert

#### Version

1.0

## 5.40 inc/managers/ScoreManager.hh File Reference

The score manager will allow you to save, load and restore game scores.

```
#include <vector>
#include <string>
#include <iostream>
#include <fstream>
#include <irrlicht/irrlicht.h>
#include <irrlicht/irrXML.h>
#include "Score.hh"
#include "Exception.hh"
```

## Classes

- class [ScoreManager](#)  
*The [ScoreManager](#) class.*

### 5.40.1 Detailed Description

The score manager will allow you to save, load and restore game scores.

#### Author

Nicolas-Emmanuel Robert

#### Version

1.0



# Index

- ~AObject
  - AObject, [10](#)
- ~APickable
  - APickable, [12](#)
- ~AScene
  - AScene, [14](#)
- ~Button
  - Button, [17](#)
- ~CoreManager
  - CoreManager, [22](#)
- ~Exception
  - error::Exception, [23](#)
- ~Game
  - Game, [25](#)
- ~GameLoadScene
  - GameLoadScene, [27](#)
- ~GameMenuReceiver
  - GameMenuReceiver, [28](#)
- ~GameMenuScene
  - GameMenuScene, [29](#)
- ~GameNewReceiver
  - GameNewReceiver, [31](#)
- ~GameNewScene
  - GameNewScene, [32](#)
- ~GameOnlineScene
  - GameOnlineScene, [33](#)
- ~GameSimulationScene
  - GameSimulationScene, [35](#)
- ~IWidget
  - IWidget, [39](#)
- ~Image
  - Image, [37](#)
- ~IntroScene
  - IntroScene, [38](#)
- ~LegendScene
  - LegendScene, [40](#)
- ~Level
  - Level, [42](#)
- ~MainScene
  - MainScene, [44](#)
- ~Model3d
  - Model3d, [48](#)
- ~OptionScene
  - OptionScene, [51](#)
- ~OptionSoundReceiver
  - OptionSoundReceiver, [53](#)
- ~OptionSoundScene
  - OptionSoundScene, [54](#)
- ~OptionVideoReceiver
  - OptionVideoReceiver, [55](#)
- ~OptionVideoScene
  - OptionVideoScene, [57](#)
- ~Potion
  - Potion, [58](#)
- ~ScoreScene
  - ScoreScene, [64](#)
- ~Window
  - Window, [67](#)
- AObject, [9](#)
  - ~AObject, [10](#)
  - getId, [10](#)
  - getModel, [10](#)
  - getType, [11](#)
  - print, [11](#)
  - setPosition, [11](#)
  - setRotation, [11](#)
- APickable, [12](#)
  - ~APickable, [12](#)
  - APickable, [12](#)
- AScene, [13](#)
  - ~AScene, [14](#)
  - AScene, [14](#)
  - addWidget, [14](#)
  - delWidget, [15](#)
  - getEventReceiver, [15](#)
  - print, [15](#)
  - updateCamera, [16](#)
- addScene
  - Window, [67](#)
- addScore
  - ScoreManager, [60](#)
- addWidget
  - AScene, [14](#)
- Button, [16](#)
  - ~Button, [17](#)
  - Button, [17](#)
  - print, [17](#)
- Camera, [18](#)
  - Camera, [18](#)
  - changePosition, [19](#)
  - changeSight, [19](#)
- changePosition
  - Camera, [19](#)
- changeScore
  - ScoreManager, [62](#)
- changeSight

- Camera, [19](#)
- ConfigManager, [19](#)
  - ConfigManager, [20](#)
  - createFile, [20](#)
  - loadConfig, [20](#)
  - sound, [21](#)
  - videoConfig, [21](#)
- CoreManager, [21](#)
  - ~CoreManager, [22](#)
  - CoreManager, [22](#)
- createFile
  - ConfigManager, [20](#)
  - ScoreManager, [62](#)
- delWidget
  - AScene, [15](#)
- deleteScene
  - Window, [68](#)
- error::Exception, [22](#)
  - ~Exception, [23](#)
  - Exception, [23](#)
  - getSeverity, [23](#)
  - what, [24](#)
- Exception
  - error::Exception, [23](#)
- Game, [24](#)
  - ~Game, [25](#)
  - Game, [24](#)
  - getLevel, [25](#)
  - setLevel, [25](#)
- GameLoadScene, [26](#)
  - ~GameLoadScene, [27](#)
  - GameLoadScene, [26](#)
  - getEventReceiver, [27](#)
- GameMenuReceiver, [27](#)
  - ~GameMenuReceiver, [28](#)
  - GameMenuReceiver, [28](#)
  - OnEvent, [28](#)
- GameMenuScene, [29](#)
  - ~GameMenuScene, [29](#)
  - GameMenuScene, [29](#)
  - getEventReceiver, [30](#)
- GameNewReceiver, [30](#)
  - ~GameNewReceiver, [31](#)
  - GameNewReceiver, [30](#)
  - OnEvent, [31](#)
- GameNewScene, [31](#)
  - ~GameNewScene, [32](#)
  - GameNewScene, [32](#)
  - getEventReceiver, [32](#)
- GameOnlineScene, [33](#)
  - ~GameOnlineScene, [33](#)
  - GameOnlineScene, [33](#)
  - getEventReceiver, [34](#)
- GameSimulationScene, [34](#)
  - ~GameSimulationScene, [35](#)
  - GameSimulationScene, [35](#)
- getEventReceiver, [35](#)
  - updateCamera, [35](#)
- getEventReceiver
  - AScene, [15](#)
  - GameLoadScene, [27](#)
  - GameMenuScene, [30](#)
  - GameNewScene, [32](#)
  - GameOnlineScene, [34](#)
  - GameSimulationScene, [35](#)
  - IntroScene, [38](#)
  - LegendScene, [41](#)
  - MainScene, [44](#)
  - OptionScene, [52](#)
  - OptionSoundScene, [54](#)
  - OptionVideoScene, [57](#)
  - ScoreScene, [64](#)
- getId
  - AObject, [10](#)
- getLevel
  - Game, [25](#)
- getModel
  - AObject, [10](#)
  - ModelsManager, [49](#)
- getPlayerScore
  - ScoreManager, [62](#)
- getScores
  - ScoreScene, [64](#)
- getSeverity
  - error::Exception, [23](#)
- getType
  - AObject, [11](#)
- IWidget, [39](#)
  - ~IWidget, [39](#)
  - print, [39](#)
- Image, [36](#)
  - ~Image, [37](#)
  - Image, [36](#)
  - print, [37](#)
- inc/exception/Exception.hh, [71](#)
- inc/game/Game.hh, [71](#)
- inc/game/level/Level.hh, [72](#)
- inc/game/level/Map.hh, [73](#)
- inc/game/level/MapFactory.hh, [73](#)
- inc/game/object/AObject.hh, [74](#)
- inc/game/object/APickable.hh, [74](#)
- inc/graphics/AScene.hh, [75](#)
- inc/graphics/Camera.hh, [76](#)
- inc/graphics/Window.hh, [88](#)
- inc/graphics/scenes/GameLoadScene.hh, [78](#)
- inc/graphics/scenes/GameMenuScene.hh, [79](#)
- inc/graphics/scenes/GameNewScene.hh, [79](#)
- inc/graphics/scenes/GameOnlineScene.hh, [80](#)
- inc/graphics/scenes/GameScene.hh, [80](#)
- inc/graphics/scenes/GameSimulationScene.hh, [81](#)
- inc/graphics/scenes/IntroScene.hh, [81](#)
- inc/graphics/scenes/LegendScene.hh, [82](#)
- inc/graphics/scenes/MainScene.hh, [82](#)
- inc/graphics/scenes/OptionControlScene.hh, [83](#)

- inc/graphics/scenes/OptionScene.hh, 83
- inc/graphics/scenes/OptionSoundScene.hh, 84
- inc/graphics/scenes/OptionVideoScene.hh, 84
- inc/graphics/scenes/ScoreScene.hh, 85
- inc/graphics/scenes/events/GameMenuReceiver.hh, 76
- inc/graphics/scenes/events/GameNewReceiver.hh, 77
- inc/graphics/scenes/events/OptionSoundReceiver.hh, 77
- inc/graphics/scenes/events/OptionVideoReceiver.hh, 78
- inc/graphics/widgets/Button.hh, 85
- inc/graphics/widgets/IWidget.hh, 86
- inc/graphics/widgets/Image.hh, 86
- inc/graphics/widgets/Model3d.hh, 87
- inc/graphics/widgets/Text.hh, 88
- inc/managers/ConfigManager.hh, 89
- inc/managers/CoreManager.hh, 90
- inc/managers/LevelManager.hh, 91
- inc/managers/ModelsManager.hh, 91
- inc/managers/SaveManager.hh, 92
- inc/managers/ScoreManager.hh, 92
- inc/managers/config/Model.hh, 89
- IntroScene, 37
  - ~IntroScene, 38
  - getEventReceiver, 38
  - IntroScene, 38
- isOpen
  - Window, 68
- LegendScene, 40
  - ~LegendScene, 40
  - getEventReceiver, 41
  - LegendScene, 40
- Level, 41
  - ~Level, 42
  - Level, 42
  - loadLevel, 42
  - movePlayer, 42
  - print, 43
- loadConfig
  - ConfigManager, 20
- loadLevel
  - Level, 42
  - SaveManager, 59
- loadMap
  - Map, 46
- loadModel
  - ModelsManager, 50
- loadSceneNodes
  - Map, 46
- loadScores
  - ScoreManager, 62
- loop
  - Window, 68
- MainScene, 43
  - ~MainScene, 44
  - getEventReceiver, 44
  - MainScene, 44
- Map, 45
  - loadMap, 46
  - loadSceneNodes, 46
  - Map, 45
- MapFactory, 46
- Model3d, 47
  - ~Model3d, 48
  - Model3d, 47
  - print, 48
  - setCurrentAnim, 48
- Model3d::Animation, 9
- ModelsManager, 49
  - getModel, 49
  - loadModel, 50
  - ModelsManager, 49
- movePlayer
  - Level, 42
- OnEvent
  - GameMenuReceiver, 28
  - GameNewReceiver, 31
  - OptionSoundReceiver, 53
  - OptionVideoReceiver, 55
- OptionControl, 50
- OptionScene, 51
  - ~OptionScene, 51
  - getEventReceiver, 52
  - OptionScene, 51
- OptionSoundReceiver, 52
  - ~OptionSoundReceiver, 53
  - OnEvent, 53
  - OptionSoundReceiver, 52
- OptionSoundScene, 53
  - ~OptionSoundScene, 54
  - getEventReceiver, 54
  - OptionSoundScene, 54
- OptionVideoReceiver, 55
  - ~OptionVideoReceiver, 55
  - OnEvent, 55
  - OptionVideoReceiver, 55
- OptionVideoScene, 56
  - ~OptionVideoScene, 57
  - getEventReceiver, 57
  - OptionVideoScene, 56
- Potion, 57
  - ~Potion, 58
  - Potion, 58
- print
  - AObject, 11
  - AScene, 15
  - Button, 17
  - IWidget, 39
  - Image, 37
  - Level, 43
  - Model3d, 48
  - Text, 66
- printScene
  - Window, 69

- SaveManager, 58
  - loadLevel, 59
- ScoreManager, 59
  - addScore, 60
  - changeScore, 62
  - createFile, 62
  - getPlayerScore, 62
  - loadScores, 62
  - ScoreManager, 60
- ScoreScene, 63
  - ~ScoreScene, 64
  - getEventReceiver, 64
  - getScores, 64
  - ScoreScene, 64
  - to\_WString, 64
- setCurrentAnim
  - Model3d, 48
- setCurrentScene
  - Window, 69
- setLevel
  - Game, 25
- setPosition
  - AObject, 11
- setRotation
  - AObject, 11
- sound
  - ConfigManager, 21
- Text, 65
  - print, 66
  - Text, 65
- to\_WString
  - ScoreScene, 64
- updateCamera
  - AScene, 16
  - GameSimulationScene, 35
- videoConfig
  - ConfigManager, 21
- what
  - error::Exception, 24
- Window, 66
  - ~Window, 67
  - addScene, 67
  - deleteScene, 68
  - isOpen, 68
  - loop, 68
  - printScene, 69
  - setCurrentScene, 69
  - Window, 67