

Zappy [GUI]

Generated by Doxygen 1.9.1

1 Namespace Index	1
1.1 Namespace List	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Data Structure Index	5
3.1 Data Structures	5
4 File Index	9
4.1 File List	9
5 Namespace Documentation	11
5.1 gui Namespace Reference	11
6 Data Structure Documentation	13
6.1 AFactory Class Reference	13
6.1.1 Detailed Description	14
6.1.2 Constructor & Destructor Documentation	14
6.1.2.1 AFactory() [1/3]	15
6.1.2.2 ~AFactory()	15
6.1.2.3 AFactory() [2/3]	15
6.1.2.4 AFactory() [3/3]	15
6.1.3 Member Function Documentation	15
6.1.3.1 createObject()	16
6.1.3.2 operator=() [1/2]	16
6.1.3.3 operator=() [2/2]	16
6.2 Arguments Class Reference	17
6.2.1 Detailed Description	18
6.2.2 Constructor & Destructor Documentation	18
6.2.2.1 Arguments() [1/3]	18
6.2.2.2 ~Arguments()	19
6.2.2.3 Arguments() [2/3]	19
6.2.2.4 Arguments() [3/3]	19
6.2.3 Member Function Documentation	19
6.2.3.1 checkArgs()	19
6.2.3.2 checkRessources()	20
6.2.3.3 displayHelp()	20
6.2.3.4 getHostname()	20
6.2.3.5 getPort()	20
6.2.3.6 operator=() [1/2]	20
6.2.3.7 operator=() [2/2]	21
6.2.3.8 processArguments()	21
6.2.4 Field Documentation	21

6.2.4.1 hostname	21
6.2.4.2 port	21
6.2.4.3 resources	21
6.3 Background Struct Reference	22
6.3.1 Detailed Description	22
6.3.2 Field Documentation	22
6.3.2.1 position	22
6.3.2.2 speed	22
6.3.2.3 sprite	23
6.3.2.4 texture	23
6.4 gui::Button Class Reference	23
6.4.1 Detailed Description	25
6.4.2 Constructor & Destructor Documentation	25
6.4.2.1 Button() [1/3]	25
6.4.2.2 Button() [2/3]	26
6.4.2.3 Button() [3/3]	26
6.4.2.4 ~Button()	26
6.4.3 Member Function Documentation	26
6.4.3.1 applyStateButton()	27
6.4.3.2 displayButton()	27
6.4.3.3 getButtonState()	27
6.4.3.4 getName()	27
6.4.3.5 getPos()	27
6.4.3.6 getRectButton()	27
6.4.3.7 getSizeButton()	27
6.4.3.8 hoverHandler()	28
6.4.3.9 isButtonPressed()	28
6.4.3.10 lockStateButton()	28
6.4.3.11 operator=() [1/2]	28
6.4.3.12 operator=() [2/2]	28
6.4.3.13 playSoundButton()	29
6.4.3.14 setButtonState()	29
6.4.3.15 setPos()	29
6.4.3.16 setShapeButton()	29
6.4.3.17 stopSoundButton()	29
6.4.4 Field Documentation	30
6.4.4.1 bufferButton	30
6.4.4.2 buttonState	30
6.4.4.3 nameButton	30
6.4.4.4 posButton	30
6.4.4.5 rectButton	30
6.4.4.6 shape	30

6.4.4.7 shapeButton	30
6.4.4.8 shapePos	31
6.4.4.9 soundButton	31
6.5 gui::Camera Class Reference	31
6.5.1 Detailed Description	32
6.5.2 Constructor & Destructor Documentation	32
6.5.2.1 Camera() [1/3]	32
6.5.2.2 ~Camera()	33
6.5.2.3 Camera() [2/3]	33
6.5.2.4 Camera() [3/3]	33
6.5.3 Member Function Documentation	33
6.5.3.1 getCenter()	33
6.5.3.2 getView()	34
6.5.3.3 getZoom()	34
6.5.3.4 operator=() [1/2]	34
6.5.3.5 operator=() [2/2]	35
6.5.3.6 setCenter()	35
6.5.3.7 setSize()	35
6.5.3.8 setView()	35
6.5.3.9 setZoom()	36
6.5.4 Field Documentation	36
6.5.4.1 view	36
6.5.4.2 zoomFactor	36
6.6 Command Struct Reference	37
6.6.1 Detailed Description	37
6.6.2 Field Documentation	37
6.6.2.1 handler	37
6.6.2.2 keyword	37
6.7 gui::Data Class Reference	38
6.7.1 Detailed Description	40
6.7.2 Constructor & Destructor Documentation	40
6.7.2.1 Data() [1/3]	40
6.7.2.2 ~Data()	40
6.7.2.3 Data() [2/3]	40
6.7.2.4 Data() [3/3]	41
6.7.3 Member Function Documentation	41
6.7.3.1 getCharacterById()	41
6.7.3.2 getCharacters()	41
6.7.3.3 getFactoryCharacters()	42
6.7.3.4 getFrequency()	42
6.7.3.5 getMap()	42
6.7.3.6 getMapFrame()	42

6.7.3.7 getTeamByName()	42
6.7.3.8 getTeamNameByPlayerId()	43
6.7.3.9 getTeams()	43
6.7.3.10 getTextBox()	44
6.7.3.11 getTilesRect()	44
6.7.3.12 getWinningTeam()	44
6.7.3.13 isPlayer()	44
6.7.3.14 isTeam()	45
6.7.3.15 operator=() [1/2]	45
6.7.3.16 operator=() [2/2]	45
6.7.3.17 setCharacters()	45
6.7.3.18 setFrequency()	46
6.7.3.19 setMapFrame()	46
6.7.3.20 setTilesRect()	46
6.7.3.21 setWinningTeam()	47
6.7.4 Field Documentation	47
6.7.4.1 characters	47
6.7.4.2 factoryCharacters	47
6.7.4.3 factoryObjects	47
6.7.4.4 map	47
6.7.4.5 mapFrame	47
6.7.4.6 teams	48
6.7.4.7 textBox	48
6.7.4.8 tilesRect	48
6.7.4.9 timeUnit	48
6.7.4.10 winningTeam	48
6.8 Eggs Class Reference	49
6.8.1 Detailed Description	52
6.8.2 Constructor & Destructor Documentation	52
6.8.2.1 Eggs() [1/3]	52
6.8.2.2 ~Eggs()	53
6.8.2.3 Eggs() [2/3]	53
6.8.2.4 Eggs() [3/3]	53
6.8.3 Member Function Documentation	53
6.8.3.1 animate()	53
6.8.3.2 display()	54
6.8.3.3 doAction()	54
6.8.3.4 getColor()	54
6.8.3.5 getId()	55
6.8.3.6 getInventory()	55
6.8.3.7 getLevel()	55
6.8.3.8 getOrientation()	56

6.8.3.9 getPos()	56
6.8.3.10 getRectLeft()	56
6.8.3.11 getTeamName()	57
6.8.3.12 getType()	57
6.8.3.13 isAlive()	57
6.8.3.14 isBroadcasting()	57
6.8.3.15 isElevating()	58
6.8.3.16 isLaying()	58
6.8.3.17 operator=() [1/2]	58
6.8.3.18 operator=() [2/2]	59
6.8.3.19 randomColor()	59
6.8.3.20 setAlive()	59
6.8.3.21 setBroadcasting()	60
6.8.3.22 setElevating()	60
6.8.3.23 setInventory()	60
6.8.3.24 setLaying()	61
6.8.3.25 setLevel()	61
6.8.3.26 setPos()	61
6.8.3.27 setResources()	61
6.8.4 Field Documentation	62
6.8.4.1 alive	62
6.8.4.2 broadcasting	62
6.8.4.3 clock	62
6.8.4.4 id	62
6.8.4.5 initial	62
6.8.4.6 laying	63
6.8.4.7 level	63
6.8.4.8 orientation	63
6.8.4.9 pos	63
6.8.4.10 rect	63
6.8.4.11 soundRadius	63
6.8.4.12 teamName	63
6.8.4.13 type	64
6.9 Errors::ErrorBroadcast Class Reference	64
6.9.1 Detailed Description	65
6.9.2 Constructor & Destructor Documentation	66
6.9.2.1 ErrorBroadcast() [1/3]	66
6.9.2.2 ~ErrorBroadcast()	66
6.9.2.3 ErrorBroadcast() [2/3]	66
6.9.2.4 ErrorBroadcast() [3/3]	66
6.9.3 Member Function Documentation	67
6.9.3.1 operator=() [1/2]	67

6.9.3.2 operator=() [2/2]	67
6.9.3.3 what()	67
6.9.4 Field Documentation	68
6.9.4.1 name	68
6.10 Errors::ErrorButton Class Reference	68
6.10.1 Detailed Description	69
6.10.2 Constructor & Destructor Documentation	70
6.10.2.1 ErrorButton() [1/3]	70
6.10.2.2 ~ErrorButton()	70
6.10.2.3 ErrorButton() [2/3]	70
6.10.2.4 ErrorButton() [3/3]	70
6.10.3 Member Function Documentation	71
6.10.3.1 operator=() [1/2]	71
6.10.3.2 operator=() [2/2]	71
6.10.3.3 what()	71
6.10.4 Field Documentation	72
6.10.4.1 name	72
6.11 Errors::ErrorCharacter Class Reference	72
6.11.1 Detailed Description	73
6.11.2 Constructor & Destructor Documentation	74
6.11.2.1 ErrorCharacter() [1/3]	74
6.11.2.2 ~ErrorCharacter()	74
6.11.2.3 ErrorCharacter() [2/3]	74
6.11.2.4 ErrorCharacter() [3/3]	74
6.11.3 Member Function Documentation	75
6.11.3.1 operator=() [1/2]	75
6.11.3.2 operator=() [2/2]	75
6.11.3.3 what()	75
6.11.4 Field Documentation	76
6.11.4.1 name	76
6.12 Errors::ErrorColor Class Reference	76
6.12.1 Constructor & Destructor Documentation	77
6.12.1.1 ErrorColor() [1/3]	78
6.12.1.2 ~ErrorColor()	78
6.12.1.3 ErrorColor() [2/3]	78
6.12.1.4 ErrorColor() [3/3]	78
6.12.2 Member Function Documentation	79
6.12.2.1 operator=() [1/2]	79
6.12.2.2 operator=() [2/2]	79
6.12.2.3 what()	79
6.12.3 Field Documentation	80
6.12.3.1 name	80

6.13 Errors::ErrorElevate Class Reference	80
6.13.1 Detailed Description	81
6.13.2 Constructor & Destructor Documentation	82
6.13.2.1 ErrorElevate() [1/3]	82
6.13.2.2 ~ErrorElevate()	82
6.13.2.3 ErrorElevate() [2/3]	82
6.13.2.4 ErrorElevate() [3/3]	82
6.13.3 Member Function Documentation	83
6.13.3.1 operator=() [1/2]	83
6.13.3.2 operator=() [2/2]	83
6.13.3.3 what()	83
6.13.4 Field Documentation	84
6.13.4.1 name	84
6.14 Errors::ErrorEnv Class Reference	84
6.14.1 Detailed Description	85
6.14.2 Constructor & Destructor Documentation	86
6.14.2.1 ErrorEnv() [1/3]	86
6.14.2.2 ~ErrorEnv()	86
6.14.2.3 ErrorEnv() [2/3]	86
6.14.2.4 ErrorEnv() [3/3]	86
6.14.3 Member Function Documentation	87
6.14.3.1 operator=() [1/2]	87
6.14.3.2 operator=() [2/2]	87
6.14.3.3 what()	87
6.14.4 Field Documentation	88
6.14.4.1 name	88
6.15 Errors::ErrorFile Class Reference	88
6.15.1 Detailed Description	89
6.15.2 Constructor & Destructor Documentation	90
6.15.2.1 ErrorFile() [1/3]	90
6.15.2.2 ~ErrorFile()	90
6.15.2.3 ErrorFile() [2/3]	90
6.15.2.4 ErrorFile() [3/3]	90
6.15.3 Member Function Documentation	91
6.15.3.1 operator=() [1/2]	91
6.15.3.2 operator=() [2/2]	91
6.15.3.3 what()	91
6.15.4 Field Documentation	92
6.15.4.1 name	92
6.16 Errors::ErrorFont Class Reference	92
6.16.1 Detailed Description	93
6.16.2 Constructor & Destructor Documentation	94

6.16.2.1 ErrorFont() [1/3]	94
6.16.2.2 ~ErrorFont()	94
6.16.2.3 ErrorFont() [2/3]	94
6.16.2.4 ErrorFont() [3/3]	94
6.16.3 Member Function Documentation	95
6.16.3.1 operator=() [1/2]	95
6.16.3.2 operator=() [2/2]	95
6.16.3.3 what()	95
6.16.4 Field Documentation	96
6.16.4.1 name	96
6.17 Errors::ErrorInventory Class Reference	96
6.17.1 Detailed Description	97
6.17.2 Constructor & Destructor Documentation	98
6.17.2.1 ErrorInventory() [1/3]	98
6.17.2.2 ~ErrorInventory()	98
6.17.2.3 ErrorInventory() [2/3]	98
6.17.2.4 ErrorInventory() [3/3]	98
6.17.3 Member Function Documentation	99
6.17.3.1 operator=() [1/2]	99
6.17.3.2 operator=() [2/2]	99
6.17.3.3 what()	99
6.17.4 Field Documentation	100
6.17.4.1 name	100
6.18 Errors::ErrorOpen Class Reference	100
6.18.1 Detailed Description	101
6.18.2 Constructor & Destructor Documentation	102
6.18.2.1 ErrorOpen() [1/3]	102
6.18.2.2 ~ErrorOpen()	102
6.18.2.3 ErrorOpen() [2/3]	102
6.18.2.4 ErrorOpen() [3/3]	102
6.18.3 Member Function Documentation	103
6.18.3.1 operator=() [1/2]	103
6.18.3.2 operator=() [2/2]	103
6.18.3.3 what()	103
6.18.4 Field Documentation	104
6.18.4.1 name	104
6.19 Errors Class Reference	104
6.19.1 Constructor & Destructor Documentation	106
6.19.1.1 Errors() [1/3]	106
6.19.1.2 ~Errors()	106
6.19.1.3 Errors() [2/3]	106
6.19.1.4 Errors() [3/3]	107

6.19.2 Member Function Documentation	107
6.19.2.1 operator=() [1/2]	107
6.19.2.2 operator=() [2/2]	107
6.20 Errors::ErrorSocket Class Reference	108
6.20.1 Detailed Description	109
6.20.2 Constructor & Destructor Documentation	110
6.20.2.1 ErrorSocket() [1/3]	110
6.20.2.2 ~ErrorSocket()	110
6.20.2.3 ErrorSocket() [2/3]	110
6.20.2.4 ErrorSocket() [3/3]	110
6.20.3 Member Function Documentation	111
6.20.3.1 operator=() [1/2]	111
6.20.3.2 operator=() [2/2]	111
6.20.3.3 what()	111
6.20.4 Field Documentation	112
6.20.4.1 name	112
6.21 ErrorT Class Reference	112
6.21.1 Detailed Description	112
6.22 Errors::ErrorTeam Class Reference	113
6.22.1 Detailed Description	114
6.22.2 Constructor & Destructor Documentation	115
6.22.2.1 ErrorTeam() [1/3]	115
6.22.2.2 ~ErrorTeam()	115
6.22.2.3 ErrorTeam() [2/3]	115
6.22.2.4 ErrorTeam() [3/3]	115
6.22.3 Member Function Documentation	116
6.22.3.1 operator=() [1/2]	116
6.22.3.2 operator=() [2/2]	116
6.22.3.3 what()	116
6.22.4 Field Documentation	117
6.22.4.1 name	117
6.23 Errors::ErrorTexture Class Reference	117
6.23.1 Detailed Description	118
6.23.2 Constructor & Destructor Documentation	119
6.23.2.1 ErrorTexture() [1/3]	119
6.23.2.2 ~ErrorTexture()	119
6.23.2.3 ErrorTexture() [2/3]	119
6.23.2.4 ErrorTexture() [3/3]	119
6.23.3 Member Function Documentation	120
6.23.3.1 operator=() [1/2]	120
6.23.3.2 operator=() [2/2]	120
6.23.3.3 what()	120

6.23.4 Field Documentation	121
6.23.4.1 name	121
6.24 Errors::ErrorUsage Class Reference	121
6.24.1 Detailed Description	122
6.24.2 Constructor & Destructor Documentation	123
6.24.2.1 ErrorUsage() [1/3]	123
6.24.2.2 ~ErrorUsage()	123
6.24.2.3 ErrorUsage() [2/3]	123
6.24.2.4 ErrorUsage() [3/3]	123
6.24.3 Member Function Documentation	124
6.24.3.1 operator=() [1/2]	124
6.24.3.2 operator=() [2/2]	124
6.24.3.3 what()	124
6.24.4 Field Documentation	125
6.24.4.1 name	125
6.25 FactoryObjects Class Reference	125
6.25.1 Constructor & Destructor Documentation	126
6.25.1.1 FactoryObjects() [1/3]	126
6.25.1.2 ~FactoryObjects()	127
6.25.1.3 FactoryObjects() [2/3]	127
6.25.1.4 FactoryObjects() [3/3]	127
6.25.2 Member Function Documentation	127
6.25.2.1 createObjects()	127
6.25.2.2 operator=() [1/2]	127
6.25.2.3 operator=() [2/2]	127
6.26 FactoryPlayers Class Reference	128
6.26.1 Detailed Description	129
6.26.2 Constructor & Destructor Documentation	130
6.26.2.1 FactoryPlayers() [1/3]	130
6.26.2.2 ~FactoryPlayers()	130
6.26.2.3 FactoryPlayers() [2/3]	130
6.26.2.4 FactoryPlayers() [3/3]	130
6.26.3 Member Function Documentation	131
6.26.3.1 createPlayers()	131
6.26.3.2 operator=() [1/2]	131
6.26.3.3 operator=() [2/2]	131
6.27 Foods Class Reference	133
6.27.1 Detailed Description	135
6.27.2 Constructor & Destructor Documentation	135
6.27.2.1 Foods() [1/3]	135
6.27.2.2 ~Foods()	135
6.27.2.3 Foods() [2/3]	135

6.27.2.4 Foods() [3/3]	136
6.27.3 Member Function Documentation	136
6.27.3.1 display()	136
6.27.3.2 getType()	136
6.27.3.3 operator=() [1/2]	137
6.27.3.4 operator=() [2/2]	137
6.27.4 Field Documentation	137
6.27.4.1 pos	137
6.27.4.2 rect	138
6.27.4.3 type	138
6.28 gui::Handler Class Reference	138
6.28.1 Detailed Description	141
6.28.2 Constructor & Destructor Documentation	141
6.28.2.1 Handler() [1/3]	141
6.28.2.2 ~Handler()	141
6.28.2.3 Handler() [2/3]	141
6.28.2.4 Handler() [3/3]	142
6.28.3 Member Function Documentation	142
6.28.3.1 bctCommand()	142
6.28.3.2 eboCommand()	142
6.28.3.3 ediCommand()	142
6.28.3.4 enwCommand()	143
6.28.3.5 handle()	143
6.28.3.6 mszCommand()	143
6.28.3.7 operator=() [1/2]	144
6.28.3.8 operator=() [2/2]	144
6.28.3.9 pbcCommand()	144
6.28.3.10 pdiCommand()	145
6.28.3.11 pdrCommand()	145
6.28.3.12 pexCommand()	145
6.28.3.13 pfkCommand()	145
6.28.3.14 pgtCommand()	146
6.28.3.15 picCommand()	146
6.28.3.16 pieCommand()	146
6.28.3.17 pinCommand()	147
6.28.3.18 plvCommand()	147
6.28.3.19 pnwCommand()	147
6.28.3.20 ppoCommand()	148
6.28.3.21 sbpCommand()	148
6.28.3.22 segCommand()	148
6.28.3.23 sgtCommand()	149
6.28.3.24 smgCommand()	149

6.28.3.25 sstCommand()	149
6.28.3.26 sucCommand()	149
6.28.3.27 tnaCommand()	150
6.28.4 Field Documentation	150
6.28.4.1 commands	150
6.29 ICharacters Class Reference	151
6.29.1 Detailed Description	154
6.29.2 Constructor & Destructor Documentation	154
6.29.2.1 ~ICharacters()	154
6.29.3 Member Function Documentation	155
6.29.3.1 animate()	155
6.29.3.2 display()	155
6.29.3.3 doAction()	155
6.29.3.4 getColor()	156
6.29.3.5 getId()	156
6.29.3.6 getInventory()	156
6.29.3.7 getLevel()	156
6.29.3.8 getOrientation()	157
6.29.3.9 getPos()	157
6.29.3.10 getRectLeft()	157
6.29.3.11 getTeamName()	157
6.29.3.12 getType()	158
6.29.3.13 isAlive()	158
6.29.3.14 isBroadcasting()	158
6.29.3.15 isElevating()	159
6.29.3.16 isLaying()	159
6.29.3.17 setAlive()	159
6.29.3.18 setBroadcasting()	159
6.29.3.19 setElevating()	160
6.29.3.20 setInventory()	160
6.29.3.21 setLaying()	160
6.29.3.22 setLevel()	161
6.29.3.23 setPos()	161
6.29.3.24 setResources()	161
6.30 Infos Struct Reference	162
6.30.1 Detailed Description	162
6.31 infos Struct Reference	162
6.31.1 Field Documentation	163
6.31.1.1 obj	163
6.31.1.2 pos	163
6.31.1.3 spriteld	163
6.32 IObjects Class Reference	163

6.32.1 Detailed Description	165
6.32.2 Constructor & Destructor Documentation	165
6.32.2.1 ~IObjects()	165
6.32.3 Member Function Documentation	165
6.32.3.1 display()	165
6.32.3.2 getType()	165
6.33 IScene Class Reference	166
6.33.1 Constructor & Destructor Documentation	167
6.33.1.1 ~IScene()	167
6.33.2 Member Function Documentation	167
6.33.2.1 checkEvents()	167
6.33.2.2 createCharacter()	168
6.33.2.3 createObject()	168
6.33.2.4 display()	168
6.33.2.5 loadScene()	168
6.34 gui::ItemDrawer Class Reference	169
6.34.1 Detailed Description	170
6.34.2 Constructor & Destructor Documentation	170
6.34.2.1 ItemDrawer() [1/3]	170
6.34.2.2 ~ItemDrawer()	170
6.34.2.3 ItemDrawer() [2/3]	170
6.34.2.4 ItemDrawer() [3/3]	171
6.34.3 Member Function Documentation	171
6.34.3.1 drawItem()	171
6.34.3.2 drawLevel()	171
6.34.3.3 drawMeteor()	172
6.34.3.4 operator=() [1/2]	172
6.34.3.5 operator=() [2/2]	173
6.34.3.6 setSprite()	173
6.34.4 Field Documentation	173
6.34.4.1 itemRect	173
6.34.4.2 itemSprite	174
6.34.4.3 levelRect	174
6.34.4.4 meteorRect	174
6.35 gui::Map Class Reference	175
6.35.1 Detailed Description	176
6.35.2 Constructor & Destructor Documentation	176
6.35.2.1 Map() [1/3]	176
6.35.2.2 ~Map()	176
6.35.2.3 Map() [2/3]	176
6.35.2.4 Map() [3/3]	177
6.35.3 Member Function Documentation	177

6.35.3.1 display()	177
6.35.3.2 getTiles()	177
6.35.3.3 initMap()	177
6.35.3.4 operator=() [1/2]	178
6.35.3.5 operator=() [2/2]	178
6.35.3.6 setTileSkin()	179
6.35.4 Field Documentation	179
6.35.4.1 backTile	179
6.35.4.2 height	179
6.35.4.3 texture	179
6.35.4.4 tiles	179
6.35.4.5 width	179
6.36 gui::Network Class Reference	180
6.36.1 Detailed Description	181
6.36.2 Constructor & Destructor Documentation	181
6.36.2.1 Network() [1/3]	181
6.36.2.2 ~Network()	182
6.36.2.3 Network() [2/3]	182
6.36.2.4 Network() [3/3]	182
6.36.3 Member Function Documentation	182
6.36.3.1 connectToServer()	182
6.36.3.2 isConnected()	183
6.36.3.3 operator=() [1/2]	183
6.36.3.4 operator=() [2/2]	183
6.36.3.5 parseCommand()	183
6.36.3.6 processData()	184
6.36.3.7 run()	184
6.36.3.8 sendCommand()	184
6.36.4 Field Documentation	184
6.36.4.1 buffer	185
6.36.4.2 connected	185
6.36.4.3 handler	185
6.36.4.4 receivedSize	185
6.36.4.5 serverIP	185
6.36.4.6 serverPort	185
6.36.4.7 socket	185
6.37 gui::PanelBar Class Reference	186
6.37.1 Detailed Description	187
6.37.2 Constructor & Destructor Documentation	187
6.37.2.1 PanelBar() [1/3]	188
6.37.2.2 ~PanelBar()	189
6.37.2.3 PanelBar() [2/3]	189

6.37.2.4 PanelBar() [3/3]	189
6.37.3 Member Function Documentation	189
6.37.3.1 createText()	190
6.37.3.2 display()	190
6.37.3.3 displayInventoryText()	190
6.37.3.4 displayRessourcesText()	190
6.37.3.5 getCamera()	192
6.37.3.6 getRessourcesText()	192
6.37.3.7 operator=() [1/2]	192
6.37.3.8 operator=() [2/2]	192
6.37.3.9 setInventoryText()	193
6.37.3.10 setRessourcesText()	193
6.37.4 Field Documentation	193
6.37.4.1 camera	193
6.37.4.2 facingRectColor	194
6.37.4.3 font	194
6.37.4.4 playerLevel	194
6.37.4.5 pos	194
6.37.4.6 resourcesInvText	194
6.37.4.7 resourcesText	194
6.38 gui::Parallax Class Reference	195
6.38.1 Detailed Description	196
6.38.2 Constructor & Destructor Documentation	196
6.38.2.1 Parallax() [1/3]	196
6.38.2.2 Parallax() [2/3]	196
6.38.2.3 Parallax() [3/3]	196
6.38.2.4 ~Parallax()	197
6.38.3 Member Function Documentation	197
6.38.3.1 display()	197
6.38.3.2 load()	197
6.38.3.3 move()	197
6.38.3.4 operator=() [1/2]	197
6.38.3.5 operator=() [2/2]	198
6.38.3.6 reset()	198
6.38.4 Field Documentation	198
6.38.4.1 clock	198
6.38.4.2 parallax	199
6.39 Players Class Reference	199
6.39.1 Detailed Description	203
6.39.2 Constructor & Destructor Documentation	203
6.39.2.1 Players() [1/4]	203
6.39.2.2 Players() [2/4]	204

6.39.2.3 ~Players()	204
6.39.2.4 Players() [3/4]	204
6.39.2.5 Players() [4/4]	205
6.39.3 Member Function Documentation	205
6.39.3.1 animate()	205
6.39.3.2 display()	205
6.39.3.3 doAction()	206
6.39.3.4 getColor()	206
6.39.3.5 getId()	206
6.39.3.6 getInventory()	207
6.39.3.7 getLevel()	207
6.39.3.8 getOrientation()	207
6.39.3.9 getPos()	207
6.39.3.10 getRectLeft()	208
6.39.3.11 getTeamName()	208
6.39.3.12 getType()	208
6.39.3.13 isAlive()	208
6.39.3.14 isBroadcasting()	209
6.39.3.15 isElevating()	209
6.39.3.16 isLaying()	209
6.39.3.17 operator=() [1/2]	209
6.39.3.18 operator=() [2/2]	210
6.39.3.19 randomColor()	210
6.39.3.20 setAlive()	210
6.39.3.21 setBroadcasting()	212
6.39.3.22 setElevating()	212
6.39.3.23 setInventory()	212
6.39.3.24 setLaying()	213
6.39.3.25 setLevel()	213
6.39.3.26 setPos()	213
6.39.3.27 setResources()	214
6.39.4 Field Documentation	214
6.39.4.1 alive	214
6.39.4.2 broadcastClock	214
6.39.4.3 broadcasting	214
6.39.4.4 circle	214
6.39.4.5 clock	215
6.39.4.6 color	215
6.39.4.7 elevateClock	215
6.39.4.8 elevating	215
6.39.4.9 foodClock	215
6.39.4.10 id	215

6.39.4.11 initial	215
6.39.4.12 inventory	215
6.39.4.13 laying	216
6.39.4.14 level	216
6.39.4.15 orientation	216
6.39.4.16 pos	216
6.39.4.17 rect	216
6.39.4.18 soundRadius	216
6.39.4.19 teamName	216
6.40 SceneGame Class Reference	217
6.40.1 Detailed Description	220
6.40.2 Constructor & Destructor Documentation	220
6.40.2.1 SceneGame() [1/3]	220
6.40.2.2 ~SceneGame()	220
6.40.2.3 SceneGame() [2/3]	220
6.40.2.4 SceneGame() [3/3]	221
6.40.3 Member Function Documentation	221
6.40.3.1 buttonsHover()	221
6.40.3.2 buttonsPressed()	221
6.40.3.3 checkEvents()	222
6.40.3.4 checkName()	222
6.40.3.5 closeEvent()	222
6.40.3.6 createCharacter()	223
6.40.3.7 createObject()	223
6.40.3.8 display()	223
6.40.3.9 displayCharacter()	224
6.40.3.10 displayInventory()	224
6.40.3.11 displayPanel()	224
6.40.3.12 displaySelected()	225
6.40.3.13 displayTileContent()	225
6.40.3.14 getCamera()	225
6.40.3.15 loadScene()	226
6.40.3.16 mapMove()	226
6.40.3.17 operator=() [1/2]	226
6.40.3.18 operator=() [2/2]	227
6.40.3.19 previousSelected()	227
6.40.3.20 removeCharacter()	227
6.40.3.21 selected()	227
6.40.4 Field Documentation	228
6.40.4.1 camera	228
6.40.4.2 currentPlayerIndex	228
6.40.4.3 itemDrawer	228

6.40.4.4 mapFrame	228
6.40.4.5 panelBars	228
6.40.4.6 panelBarVisible	229
6.40.4.7 showInventory	229
6.40.4.8 sprite	229
6.40.4.9 spriteCharacters	229
6.40.4.10 spritePanel	229
6.40.4.11 texture	229
6.40.4.12 textureCharacters	229
6.40.4.13 texturePanel	230
6.41 gui::SceneManager Class Reference	230
6.41.1 Detailed Description	233
6.41.2 Constructor & Destructor Documentation	233
6.41.2.1 SceneManager() [1/3]	233
6.41.2.2 ~SceneManager()	233
6.41.2.3 SceneManager() [2/3]	233
6.41.2.4 SceneManager() [3/3]	233
6.41.3 Member Function Documentation	234
6.41.3.1 createButtons()	234
6.41.3.2 createSounds()	234
6.41.3.3 display()	234
6.41.3.4 getButton()	234
6.41.3.5 getButtons()	235
6.41.3.6 getMapFrameSprite()	235
6.41.3.7 getSoundBox()	235
6.41.3.8 getSprite()	235
6.41.3.9 getTileSprite()	236
6.41.3.10 getVolSprite()	236
6.41.3.11 getVolume()	236
6.41.3.12 loadScene()	236
6.41.3.13 operator=() [1/2]	236
6.41.3.14 operator=() [2/2]	237
6.41.3.15 setMapFrameSprite()	237
6.41.3.16 setState()	237
6.41.3.17 setTileSprite()	238
6.41.3.18 setVolume()	238
6.41.3.19 update()	238
6.41.4 Field Documentation	239
6.41.4.1 buttons	239
6.41.4.2 currentState	239
6.41.4.3 mapFrameSprite	239
6.41.4.4 mapFrameTexture	239

6.41.4.5 soundbox	239
6.41.4.6 sprite	239
6.41.4.7 texture	239
6.41.4.8 tileSprite	240
6.41.4.9 tileTexture	240
6.41.4.10 volSprite	240
6.42 SceneMenu Class Reference	240
6.42.1 Detailed Description	243
6.42.2 Constructor & Destructor Documentation	243
6.42.2.1 SceneMenu() [1/3]	243
6.42.2.2 ~SceneMenu()	243
6.42.2.3 SceneMenu() [2/3]	243
6.42.2.4 SceneMenu() [3/3]	244
6.42.3 Member Function Documentation	244
6.42.3.1 checkEvents()	244
6.42.3.2 createCharacter()	244
6.42.3.3 createObject()	245
6.42.3.4 display()	245
6.42.3.5 loadScene()	245
6.42.3.6 operator=() [1/2]	246
6.42.3.7 operator=() [2/2]	246
6.42.4 Field Documentation	246
6.42.4.1 camera	246
6.42.4.2 connexionSprite	247
6.42.4.3 logo	247
6.42.4.4 parallax	247
6.42.4.5 texture	247
6.42.4.6 textureLogo	247
6.43 SceneSettings Class Reference	248
6.43.1 Detailed Description	250
6.43.2 Constructor & Destructor Documentation	250
6.43.2.1 SceneSettings() [1/3]	250
6.43.2.2 ~SceneSettings()	250
6.43.2.3 SceneSettings() [2/3]	250
6.43.2.4 SceneSettings() [3/3]	251
6.43.3 Member Function Documentation	251
6.43.3.1 checkEvents()	251
6.43.3.2 createCharacter()	251
6.43.3.3 createObject()	252
6.43.3.4 display()	252
6.43.3.5 loadScene()	252
6.43.3.6 operator=() [1/2]	253

6.43.3.7 operator=() [2/2]	253
6.43.4 Field Documentation	253
6.43.4.1 camera	253
6.43.4.2 volumeChanged	253
6.44 SceneSkins Class Reference	254
6.44.1 Detailed Description	256
6.44.2 Constructor & Destructor Documentation	256
6.44.2.1 SceneSkins() [1/3]	256
6.44.2.2 ~SceneSkins()	257
6.44.2.3 SceneSkins() [2/3]	257
6.44.2.4 SceneSkins() [3/3]	257
6.44.3 Member Function Documentation	257
6.44.3.1 checkEvents()	257
6.44.3.2 createCharacter()	258
6.44.3.3 createObject()	258
6.44.3.4 directionnalButtons()	258
6.44.3.5 display()	258
6.44.3.6 loadScene()	259
6.44.3.7 operator=() [1/2]	259
6.44.3.8 operator=() [2/2]	260
6.44.4 Field Documentation	260
6.44.4.1 buffer	260
6.44.4.2 camera	260
6.44.4.3 sound	260
6.45 SceneWin Class Reference	261
6.45.1 Constructor & Destructor Documentation	263
6.45.1.1 SceneWin() [1/3]	263
6.45.1.2 ~SceneWin()	263
6.45.1.3 SceneWin() [2/3]	263
6.45.1.4 SceneWin() [3/3]	264
6.45.2 Member Function Documentation	264
6.45.2.1 checkEvents()	264
6.45.2.2 createCharacter()	264
6.45.2.3 createObject()	265
6.45.2.4 display()	265
6.45.2.5 loadScene()	265
6.45.2.6 operator=() [1/2]	266
6.45.2.7 operator=() [2/2]	266
6.45.3 Field Documentation	266
6.45.3.1 buffer	266
6.45.3.2 camera	266
6.45.3.3 font	267

6.45.3.4 haswonText	267
6.45.3.5 sound	267
6.45.3.6 winText	267
6.46 Song Struct Reference	267
6.46.1 Detailed Description	268
6.46.2 Field Documentation	268
6.46.2.1 name	269
6.46.2.2 soundPlayer	269
6.47 SoundBox Class Reference	269
6.47.1 Detailed Description	270
6.47.2 Constructor & Destructor Documentation	270
6.47.2.1 SoundBox() [1/3]	270
6.47.2.2 ~SoundBox()	271
6.47.2.3 SoundBox() [2/3]	271
6.47.2.4 SoundBox() [3/3]	271
6.47.3 Member Function Documentation	271
6.47.3.1 getSoundStatus()	271
6.47.3.2 getVolume()	272
6.47.3.3 initSong()	272
6.47.3.4 operator=() [1/2]	272
6.47.3.5 operator=() [2/2]	272
6.47.3.6 play()	274
6.47.3.7 setVolume()	274
6.47.3.8 stop()	274
6.47.4 Field Documentation	275
6.47.4.1 sounds	275
6.47.4.2 volume	275
6.48 SoundPlayer Class Reference	275
6.48.1 Detailed Description	276
6.48.2 Constructor & Destructor Documentation	276
6.48.2.1 SoundPlayer() [1/3]	276
6.48.2.2 ~SoundPlayer()	277
6.48.2.3 SoundPlayer() [2/3]	277
6.48.2.4 SoundPlayer() [3/3]	277
6.48.3 Member Function Documentation	277
6.48.3.1 getBuffer()	277
6.48.3.2 getIsPlaying()	277
6.48.3.3 getSound()	278
6.48.3.4 init()	278
6.48.3.5 operator=() [1/2]	278
6.48.3.6 operator=() [2/2]	278
6.48.3.7 setBuffer()	278

6.48.3.8 setIsPlaying()	279
6.48.3.9 setSound()	279
6.48.4 Field Documentation	279
6.48.4.1 buffer	279
6.48.4.2 isPlaying	279
6.48.4.3 sound	279
6.49 Stones Class Reference	280
6.49.1 Detailed Description	282
6.49.2 Constructor & Destructor Documentation	282
6.49.2.1 Stones() [1/3]	282
6.49.2.2 ~Stones()	282
6.49.2.3 Stones() [2/3]	282
6.49.2.4 Stones() [3/3]	283
6.49.3 Member Function Documentation	283
6.49.3.1 display()	283
6.49.3.2 getType()	283
6.49.3.3 operator=() [1/2]	284
6.49.3.4 operator=() [2/2]	284
6.49.4 Field Documentation	284
6.49.4.1 pos	284
6.49.4.2 rect	284
6.49.4.3 type	285
6.49.4.4 typeObject	285
6.50 gui::Team Class Reference	285
6.50.1 Detailed Description	286
6.50.2 Constructor & Destructor Documentation	286
6.50.2.1 Team() [1/4]	286
6.50.2.2 Team() [2/4]	286
6.50.2.3 Team() [3/4]	287
6.50.2.4 Team() [4/4]	287
6.50.2.5 ~Team()	287
6.50.3 Member Function Documentation	287
6.50.3.1 addCharacter()	287
6.50.3.2 getCharacters()	288
6.50.3.3 getName()	288
6.50.3.4 operator=() [1/2]	288
6.50.3.5 operator=() [2/2]	289
6.50.3.6 setName()	289
6.50.4 Field Documentation	289
6.50.4.1 characters	289
6.50.4.2 name	289
6.51 TextBox Class Reference	290

6.51.1 Detailed Description	291
6.51.2 Constructor & Destructor Documentation	291
6.51.2.1 TextBox() [1/3]	291
6.51.2.2 ~TextBox()	292
6.51.2.3 TextBox() [2/3]	292
6.51.2.4 TextBox() [3/3]	292
6.51.3 Member Function Documentation	292
6.51.3.1 activate()	292
6.51.3.2 clear()	293
6.51.3.3 draw()	293
6.51.3.4 getPosition()	293
6.51.3.5 getString()	293
6.51.3.6 initTextBox()	294
6.51.3.7 isFinished()	294
6.51.3.8 isInputMode()	294
6.51.3.9 isPressed()	294
6.51.3.10 operator=() [1/2]	295
6.51.3.11 operator=() [2/2]	295
6.51.3.12 setString()	295
6.51.3.13 textEvent()	296
6.51.4 Field Documentation	296
6.51.4.1 box	296
6.51.4.2 descriptionText	296
6.51.4.3 finished	296
6.51.4.4 inputMode	297
6.51.4.5 inputText	297
6.51.4.6 rect	297
6.51.4.7 text	297
6.52 gui::Tile Class Reference	297
6.52.1 Constructor & Destructor Documentation	299
6.52.1.1 Tile() [1/3]	299
6.52.1.2 ~Tile()	300
6.52.1.3 Tile() [2/3]	300
6.52.1.4 Tile() [3/3]	300
6.52.2 Member Function Documentation	300
6.52.2.1 drawContent()	300
6.52.2.2 getFRect()	301
6.52.2.3 getIsSelected()	301
6.52.2.4 getPosition()	301
6.52.2.5 getRect()	301
6.52.2.6 getResources()	302
6.52.2.7 getShape()	302

6.52.2.8 operator=() [1/2]	302
6.52.2.9 operator=() [2/2]	302
6.52.2.10 setFRect()	303
6.52.2.11 setIsSelected()	303
6.52.2.12 setPosition()	303
6.52.2.13 setRect()	303
6.52.2.14 setResources() [1/2]	304
6.52.2.15 setResources() [2/2]	304
6.52.2.16 setShape()	304
6.52.3 Field Documentation	305
6.52.3.1 fRect	305
6.52.3.2 isSelected	305
6.52.3.3 pos	305
6.52.3.4 rect	305
6.52.3.5 resources	305
6.52.3.6 shape	305
6.53 Tile Class Reference	306
6.53.1 Detailed Description	306
6.54 gui::Window Class Reference	306
6.54.1 Detailed Description	309
6.54.2 Constructor & Destructor Documentation	309
6.54.2.1 Window() [1/3]	309
6.54.2.2 ~Window()	309
6.54.2.3 Window() [2/3]	309
6.54.2.4 Window() [3/3]	309
6.54.3 Member Function Documentation	310
6.54.3.1 clearWindow()	310
6.54.3.2 createWindow()	310
6.54.3.3 destroyWindow()	310
6.54.3.4 displayHostPortText()	311
6.54.3.5 displayWindow()	311
6.54.3.6 drawObject()	311
6.54.3.7 getBackground()	311
6.54.3.8 getCamera()	311
6.54.3.9 getEvent()	312
6.54.3.10 getMousePos()	312
6.54.3.11 getWindow()	312
6.54.3.12 isOpenWindow()	312
6.54.3.13 operator=() [1/2]	312
6.54.3.14 operator=() [2/2]	313
6.54.3.15 setBackground()	313
6.54.3.16 setClearColor()	313

6.54.3.17 setCursor()	314
6.54.3.18 setCursorPos()	314
6.54.3.19 setMousePos()	314
6.54.4 Field Documentation	314
6.54.4.1 clearColor	315
6.54.4.2 cursorSprite	315
6.54.4.3 cursorTexture	315
6.54.4.4 event	315
6.54.4.5 font	315
6.54.4.6 hostname	315
6.54.4.7 hostText	315
6.54.4.8 mousePos	315
6.54.4.9 port	316
6.54.4.10 portText	316
6.54.4.11 sprite	316
6.54.4.12 texture	316
6.54.4.13 window	316
7 File Documentation	317
7.1 Arguments/Arguments.cpp File Reference	317
7.2 Arguments/Arguments.hpp File Reference	317
7.3 Camera/Camera.cpp File Reference	318
7.4 Camera/Camera.hpp File Reference	318
7.5 Characters/Eggs/Eggs.cpp File Reference	319
7.5.1 Function Documentation	320
7.5.1.1 findRectEggs()	320
7.5.2 Variable Documentation	320
7.5.2.1 rectEggs	320
7.6 Characters/Eggs/Eggs.hpp File Reference	320
7.6.1 Enumeration Type Documentation	321
7.6.1.1 TypeEggs	321
7.7 Characters/ICharacters.hpp File Reference	321
7.7.1 Enumeration Type Documentation	322
7.7.1.1 CharacterType	322
7.8 Characters/Players/Orientation.cpp File Reference	323
7.8.1 Function Documentation	323
7.8.1.1 blueRectFunc()	323
7.8.1.2 darkBlueRectFunc()	323
7.8.1.3 getFacingRectByColor()	324
7.8.1.4 getRightRect()	324
7.8.1.5 greenRectFunc()	324
7.8.1.6 purpleRectFunc()	324

7.8.1.7 redRectFunc()	324
7.8.1.8 yellowRectFunc()	324
7.9 Characters/Players/Players.cpp File Reference	325
7.9.1 Function Documentation	325
7.9.1.1 getRightRect()	325
7.9.2 Variable Documentation	325
7.9.2.1 PATH_GOLEM	325
7.10 Characters/Players/Players.hpp File Reference	326
7.10.1 Variable Documentation	326
7.10.1.1 broadcastColor	326
7.11 Components/Buttons/Button.cpp File Reference	327
7.12 Components/Buttons/Button.hpp File Reference	327
7.12.1 Enumeration Type Documentation	328
7.12.1.1 ButtonState	328
7.13 Components/PanelBar/PanelBar.cpp File Reference	328
7.13.1 Function Documentation	329
7.13.1.1 getFacingRectByColor()	329
7.14 Components/PanelBar/PanelBar.hpp File Reference	329
7.15 Components/Parallax/Parallax.cpp File Reference	330
7.16 Components/Parallax/Parallax.hpp File Reference	330
7.17 Components/SoundBox/SoundBox.cpp File Reference	331
7.18 Components/SoundBox/SoundBox.hpp File Reference	332
7.19 Components/SoundBox/SoundPlayers.cpp File Reference	333
7.20 Components/SoundBox/SoundPlayers.hpp File Reference	333
7.21 Components/TextBox/TextBox.cpp File Reference	334
7.22 Components/TextBox/TextBox.hpp File Reference	335
7.23 Components/Window/Window.cpp File Reference	335
7.24 Components/Window/Window.hpp File Reference	336
7.25 Database/Data.cpp File Reference	336
7.26 Database/Data.hpp File Reference	337
7.27 Errors/Errors.cpp File Reference	338
7.28 Errors/Errors.hpp File Reference	338
7.29 Factory/AFactory.hpp File Reference	339
7.30 Factory/FactoryObjects/FactoryObjects.hpp File Reference	340
7.31 Factory/FactoryPlayers/FactoryPlayers.hpp File Reference	340
7.32 ItemDrawer/ItemDrawer.cpp File Reference	341
7.33 ItemDrawer/ItemDrawer.hpp File Reference	341
7.34 Main.cpp File Reference	342
7.34.1 Function Documentation	342
7.34.1.1 launch()	343
7.34.1.2 main()	343
7.35 Map/Map.cpp File Reference	343

7.36 Map/Map.hpp File Reference	344
7.36.1 Variable Documentation	345
7.36.1.1 PATH_BACK	345
7.36.1.2 RECT_GRASS	345
7.37 Map/Tile.cpp File Reference	345
7.38 Map/Tile.hpp File Reference	346
7.38.1 Variable Documentation	346
7.38.1.1 TILE_SIZE	347
7.39 Network/Handlers/Commands/bct.cpp File Reference	347
7.39.1 Function Documentation	347
7.39.1.1 get_char_occurrence()	347
7.40 Network/Handlers/Commands/ebo.cpp File Reference	347
7.41 Network/Handlers/Commands/edi.cpp File Reference	348
7.42 Network/Handlers/Commands/enw.cpp File Reference	348
7.43 Network/Handlers/Commands/msz.cpp File Reference	348
7.44 Network/Handlers/Commands/pbc.cpp File Reference	349
7.45 Network/Handlers/Commands/pdi.cpp File Reference	349
7.46 Network/Handlers/Commands/pdr.cpp File Reference	349
7.47 Network/Handlers/Commands/pex.cpp File Reference	350
7.48 Network/Handlers/Commands/pfk.cpp File Reference	350
7.49 Network/Handlers/Commands/pgt.cpp File Reference	350
7.50 Network/Handlers/Commands/pic.cpp File Reference	351
7.51 Network/Handlers/Commands/pie.cpp File Reference	351
7.52 Network/Handlers/Commands/pin.cpp File Reference	351
7.53 Network/Handlers/Commands/plv.cpp File Reference	352
7.54 Network/Handlers/Commands/pnw.cpp File Reference	352
7.55 Network/Handlers/Commands/ppo.cpp File Reference	352
7.56 Network/Handlers/Commands/sbp.cpp File Reference	353
7.57 Network/Handlers/Commands/seg.cpp File Reference	353
7.58 Network/Handlers/Commands/sgt.cpp File Reference	353
7.59 Network/Handlers/Commands/msg.cpp File Reference	354
7.60 Network/Handlers/Commands/sst.cpp File Reference	354
7.61 Network/Handlers/Commands/suc.cpp File Reference	354
7.62 Network/Handlers/Commands/tna.cpp File Reference	355
7.63 Network/Handlers/Handler.cpp File Reference	355
7.63.1 Function Documentation	355
7.63.1.1 operator>>()	355
7.64 Network/Handlers/Handler.hpp File Reference	356
7.64.1 Function Documentation	357
7.64.1.1 operator>>()	357
7.65 Network/Network.cpp File Reference	357
7.66 Network/Network.hpp File Reference	357

7.66.1 Variable Documentation	359
7.66.1.1 maxBufferSize	359
7.67 Objects/Foods/Foods.cpp File Reference	359
7.67.1 Function Documentation	359
7.67.1.1 findRect()	360
7.67.2 Variable Documentation	360
7.67.2.1 rects	360
7.68 Objects/Foods/Foods.hpp File Reference	360
7.69 Objects/IObjects.hpp File Reference	361
7.69.1 Enumeration Type Documentation	362
7.69.1.1 TypeObject	362
7.69.2 Variable Documentation	362
7.69.2.1 PATH_SPRITE	362
7.70 Objects/Stones/Stones.cpp File Reference	362
7.70.1 Function Documentation	363
7.70.1.1 findRect()	363
7.70.2 Variable Documentation	363
7.70.2.1 rects	364
7.71 Objects/Stones/Stones.hpp File Reference	364
7.71.1 Enumeration Type Documentation	365
7.71.1.1 TypeStone	365
7.72 Scene/Game/Display/Display.cpp File Reference	365
7.73 Scene/Game/Display/DisplayTile.cpp File Reference	365
7.74 Scene/Game/Event/Event.cpp File Reference	366
7.75 Scene/Game/Event/MapMove.cpp File Reference	366
7.76 Scene/Game/Event/Selected.cpp File Reference	367
7.77 Scene/Game/SceneGame.cpp File Reference	367
7.77.1 Variable Documentation	367
7.77.1.1 PATH_GOLEM	368
7.77.1.2 PATH_PANEL	368
7.78 Scene/Game/SceneGame.hpp File Reference	368
7.79 Scene/IScene.hpp File Reference	369
7.80 Scene/Manager/SceneManager.cpp File Reference	369
7.81 Scene/Manager/SceneManager.hpp File Reference	370
7.82 Scene/Menu/SceneMenu.cpp File Reference	371
7.83 Scene/Menu/SceneMenu.hpp File Reference	371
7.84 Scene/Settings/SceneSettings.cpp File Reference	372
7.85 Scene/Settings/SceneSettings.hpp File Reference	372
7.86 Scene/Skins/SceneSkins.cpp File Reference	373
7.87 Scene/Skins/SceneSkins.hpp File Reference	373
7.88 Scene/Win/SceneWin.cpp File Reference	374
7.89 Scene/Win/SceneWin.hpp File Reference	374

7.90 Team/Team.cpp File Reference	375
7.91 Team/Team.hpp File Reference	375
Index	377

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

gui	11
-------------------------------	--------------------

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

AFactory	13
FactoryObjects	125
FactoryPlayers	128
Arguments	17
Background	22
gui::Button	23
gui::Camera	31
Command	37
gui::Data	38
ErrorT	112
std::exception	
Errors	104
Errors::ErrorBroadcast	64
Errors::ErrorButton	68
Errors::ErrorCharacter	72
Errors::ErrorColor	76
Errors::ErrorElevate	80
Errors::ErrorEnv	84
Errors::ErrorFile	88
Errors::ErrorFont	92
Errors::ErrorInventory	96
Errors::ErrorOpen	100
Errors::ErrorSocket	108
Errors::ErrorTeam	113
Errors::ErrorTexture	117
Errors::ErrorUsage	121
gui::Handler	138
ICharacters	151
Eggs	49
Players	199
Infos	162
infos	162
IObjects	163
Foods	133
Stones	280

IScene	166
SceneGame	217
SceneMenu	240
SceneSettings	248
SceneSkins	254
SceneWin	261
gui::ItemDrawer	169
gui::Map	175
gui::Network	180
gui::PanelBar	186
gui::Parallax	195
gui::SceneManager	230
Song	267
SoundBox	269
SoundPlayer	275
gui::Team	285
TextBox	290
gui::Tile	297
Tile	306
gui::Window	306

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

AFactory	
AFactory class	13
Arguments	
Class for the arguments	17
Background	
Background struct	22
gui::Button	
Button class	23
gui::Camera	
Camera class	31
Command	
Command struct	37
gui::Data	
Data class	38
Eggs	
Eggs class	49
Errors::ErrorBroadcast	
A class for broadcast Error	64
Errors::ErrorButton	
A class for Button Error	68
Errors::ErrorCharacter	
A class for Character Error	72
Errors::ErrorColor	76
Errors::ErrorElevate	
A class for Elevate Error	80
Errors::ErrorEnv	
A class for env Error	84
Errors::ErrorFile	
A class for file Error	88
Errors::ErrorFont	
A class for font Error	92
Errors::ErrorInventory	
A class for Inventory Error	96
Errors::ErrorOpen	
A class for open Error	100

Errors	104
Errors::ErrorSocket	
A class for socket Error	108
ErrorT	
A class to handle errors. This class provides a way to throw errors	112
Errors::ErrorTeam	
A class for Team Error	113
Errors::ErrorTexture	
A class for Texture Error	117
Errors::ErrorUsage	
A class for Usage Error	121
FactoryObjects	125
FactoryPlayers	
FactoryPlayers class	128
Foods	
Foods class	133
gui::Handler	
Handler class	138
ICharacters	
Interface for the characters	151
Infos	
Infos struct	162
infos	162
IObjects	
IObjects class	163
IScene	166
gui::ItemDrawer	
ItemDrawer class	169
gui::Map	
Map Class	175
gui::Network	
Network class	180
gui::PanelBar	
PanelBar class	186
gui::Parallax	
Parallax class	195
Players	
Players class	199
SceneGame	
Game scene class	217
gui::SceneManager	
SceneManager class	230
SceneMenu	
SceneMenu class	240
SceneSettings	
Settings scene class	248
SceneSkins	
Skins scene class	254
SceneWin	261
Song	
Song struct	267
SoundBox	
SoundBox class	269
SoundPlayer	
SoundPlayer class	275
Stones	
Stones class	280

gui::Team		
Team	class	285
TextBox		
TextBox	class	290
gui::Tile		297
Tile		
Tile	class	306
gui::Window		
Window	class	306

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Main.cpp	342
Arguments/Arguments.cpp	317
Arguments/Arguments.hpp	317
Camera/Camera.cpp	318
Camera/Camera.hpp	318
Characters/ICharacters.hpp	321
Characters/Eggs/Eggs.cpp	319
Characters/Eggs/Eggs.hpp	320
Characters/Players/Orientation.cpp	323
Characters/Players/Players.cpp	325
Characters/Players/Players.hpp	326
Components/Buttons/Button.cpp	327
Components/Buttons/Button.hpp	327
Components/PanelBar/PanelBar.cpp	328
Components/PanelBar/PanelBar.hpp	329
Components/Parallax/Parallax.cpp	330
Components/Parallax/Parallax.hpp	330
Components/SoundBox/SoundBox.cpp	331
Components/SoundBox/SoundBox.hpp	332
Components/SoundBox/SoundPlayers.cpp	333
Components/SoundBox/SoundPlayers.hpp	333
Components/TextBox/TextBox.cpp	334
Components/TextBox/TextBox.hpp	335
Components/Window/Window.cpp	335
Components/Window/Window.hpp	336
Database/Data.cpp	336
Database/Data.hpp	337
Errors/Errors.cpp	338
Errors/Errors.hpp	338
Factory/AFactory.hpp	339
Factory/FactoryObjects/FactoryObjects.hpp	340
Factory/FactoryPlayers/FactoryPlayers.hpp	340
ItemDrawer/ItemDrawer.cpp	341
ItemDrawer/ItemDrawer.hpp	341
Map/Map.cpp	343

Map/Map.hpp	344
Map/Tile.cpp	345
Map/Tile.hpp	346
Network/Network.cpp	357
Network/Network.hpp	357
Network/Handlers/Handler.cpp	355
Network/Handlers/Handler.hpp	356
Network/Handlers/Commands/bct.cpp	347
Network/Handlers/Commands/ebo.cpp	347
Network/Handlers/Commands/edi.cpp	348
Network/Handlers/Commands/enw.cpp	348
Network/Handlers/Commands/msz.cpp	348
Network/Handlers/Commands/pbc.cpp	349
Network/Handlers/Commands/pdi.cpp	349
Network/Handlers/Commands/pdr.cpp	349
Network/Handlers/Commands/pex.cpp	350
Network/Handlers/Commands/pfk.cpp	350
Network/Handlers/Commands/pgt.cpp	350
Network/Handlers/Commands/pic.cpp	351
Network/Handlers/Commands/pie.cpp	351
Network/Handlers/Commands/pin.cpp	351
Network/Handlers/Commands/plv.cpp	352
Network/Handlers/Commands/pnw.cpp	352
Network/Handlers/Commands/ppo.cpp	352
Network/Handlers/Commands/sbp.cpp	353
Network/Handlers/Commands/seg.cpp	353
Network/Handlers/Commands/sgt.cpp	353
Network/Handlers/Commands/smg.cpp	354
Network/Handlers/Commands/sst.cpp	354
Network/Handlers/Commands/suc.cpp	354
Network/Handlers/Commands/tna.cpp	355
Objects/IObjects.hpp	361
Objects/Foods/Foods.cpp	359
Objects/Foods/Foods.hpp	360
Objects/Stones/Stones.cpp	362
Objects/Stones/Stones.hpp	364
Scene/IScene.hpp	369
Scene/Game/SceneGame.cpp	367
Scene/Game/SceneGame.hpp	368
Scene/Game/Display/Display.cpp	365
Scene/Game/Display/DisplayTile.cpp	365
Scene/Game/Event/Event.cpp	366
Scene/Game/Event/MapMove.cpp	366
Scene/Game/Event/Selected.cpp	367
Scene/Manager/SceneManager.cpp	369
Scene/Manager/SceneManager.hpp	370
Scene/Menu/SceneMenu.cpp	371
Scene/Menu/SceneMenu.hpp	371
Scene/Settings/SceneSettings.cpp	372
Scene/Settings/SceneSettings.hpp	372
Scene/Skins/SceneSkins.cpp	373
Scene/Skins/SceneSkins.hpp	373
Scene/Win/SceneWin.cpp	374
Scene/Win/SceneWin.hpp	374
Team/Team.cpp	375
Team/Team.hpp	375

Chapter 5

Namespace Documentation

5.1 gui Namespace Reference

Data Structures

- class [Camera](#)
Camera class.
- class [Button](#)
Button class.
- class [PanelBar](#)
PanelBar class.
- class [Parallax](#)
Parallax class.
- class [Window](#)
Window class.
- class [Data](#)
Data class.
- class [ItemDrawer](#)
ItemDrawer class.
- class [Map](#)
Map Class.
- class [Tile](#)
- class [Handler](#)
Handler class.
- class [Network](#)
Network class.
- class [SceneManager](#)
SceneManager class.
- class [Team](#)
Team class.

Chapter 6

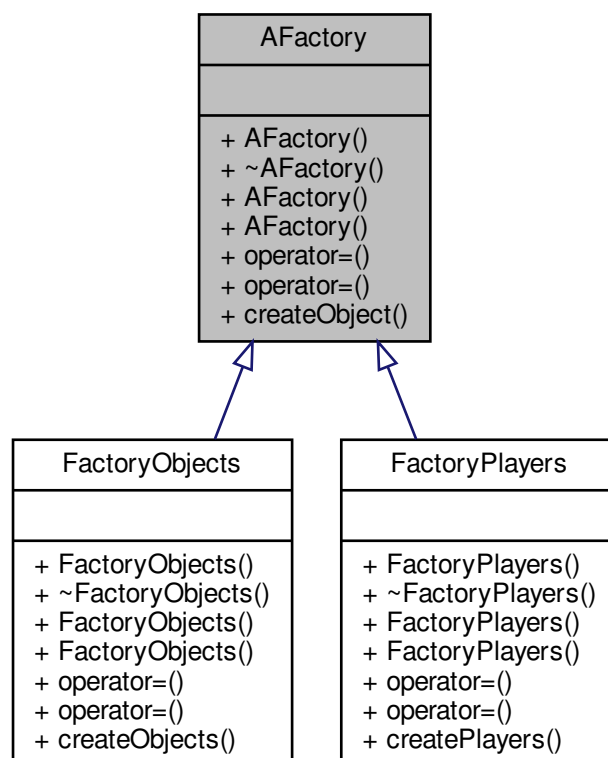
Data Structure Documentation

6.1 AFactory Class Reference

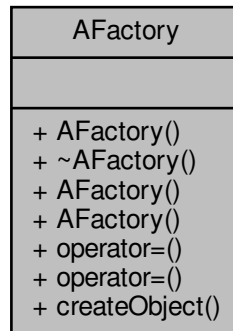
[AFactory](#) class.

```
#include <AFactory.hpp>
```

Inheritance diagram for AFactory:



Collaboration diagram for AFactory:



Public Member Functions

- [AFactory](#) ()=default
Construct a new [AFactory](#) object.
- [~AFactory](#) () noexcept=default
Destroy the [AFactory](#) object.
- [AFactory](#) (const [AFactory](#) &other)=delete
Construct a new [AFactory](#) object by copy = delete.
- [AFactory](#) ([AFactory](#) &&other)=default
Construct a new [AFactory](#) object by move = default.
- [AFactory](#) & operator= (const [AFactory](#) &other)=delete
Assign a [AFactory](#) object by copy = delete.
- [AFactory](#) & operator= ([AFactory](#) &&other)=default
Assign a [AFactory](#) object by move = default.
- template<typename T , typename ... Args>
std::unique_ptr< T > [createObject](#) (Args &&...args)
Create a Object.

6.1.1 Detailed Description

[AFactory](#) class.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 AFactory() [1/3]

```
AFactory::AFactory ( ) [default]
```

Construct a new [AFactory](#) object.

6.1.2.2 ~AFactory()

```
AFactory::~~AFactory ( ) [default], [noexcept]
```

Destroy the [AFactory](#) object.

6.1.2.3 AFactory() [2/3]

```
AFactory::AFactory (
    const AFactory & other ) [delete]
```

Construct a new [AFactory](#) object by copy = delete.

Parameters

<i>other</i>	
--------------	--

6.1.2.4 AFactory() [3/3]

```
AFactory::AFactory (
    AFactory && other ) [default]
```

Construct a new [AFactory](#) object by move = default.

Parameters

<i>other</i>	
--------------	--

6.1.3 Member Function Documentation

6.1.3.1 createObject()

```
template<typename T , typename ... Args>
std::unique_ptr<T> AFactory::createObject (
    Args &&... args ) [inline]
```

Create a Object.

Template Parameters

<i>T</i>	
<i>Args</i>	

Parameters

<i>args</i>	
-------------	--

Returns

std::unique_ptr<T>

6.1.3.2 operator=() [1/2]

```
AFactory& AFactory::operator= (
    AFactory && other ) [default]
```

Assign a [AFactory](#) object by move = default.

Parameters

<i>other</i>	
--------------	--

Returns

[AFactory](#)&

6.1.3.3 operator=() [2/2]

```
AFactory& AFactory::operator= (
    const AFactory & other ) [delete]
```

Assign a [AFactory](#) object by copy = delete.

Parameters

<i>other</i>	
--------------	--

Returns

[AFactory](#)&

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

- [Factory/AFactory.hpp](#)

6.2 Arguments Class Reference

Class for the arguments.

```
#include <Arguments.hpp>
```

Collaboration diagram for Arguments:

Arguments
<ul style="list-style-type: none">- port- hostname- resources
<ul style="list-style-type: none">+ Arguments()+ ~Arguments()+ Arguments()+ operator=()+ Arguments()+ operator=()+ displayHelp()+ processArguments()+ checkRessources()+ checkArgs()+ getPort()+ getHostname()

Public Member Functions

- [Arguments](#) ()
Construct a new [Arguments](#) object.
- [~Arguments](#) ()=default
Destroy the [Arguments](#) object.
- [Arguments](#) (const [Arguments](#) &)=delete
Construct a new [Arguments](#) object by copy deleted.
- [Arguments](#) & operator= (const [Arguments](#) &)=delete
Assign a [Arguments](#) object by copy deleted.
- [Arguments](#) ([Arguments](#) &&)=default
Construct a new [Arguments](#) object by move.
- [Arguments](#) & operator= ([Arguments](#) &&)=default
Assign a [Arguments](#) object by move.
- void [displayHelp](#) ()
Display the help.
- void [processArguments](#) (const int argc, char *const argv[])
Process the arguments.
- void [checkRessources](#) ()
Check if the resources exist.
- int [checkArgs](#) (const int argc, char *const argv[])
Check the arguments.
- unsigned int [getPort](#) () const
Get the port.
- const char * [getHostname](#) () const
Get the hostname.

Private Attributes

- unsigned int [port](#)
- const char * [hostname](#)
- std::vector< std::string > [resources](#)

6.2.1 Detailed Description

Class for the arguments.

6.2.2 Constructor & Destructor Documentation

6.2.2.1 Arguments() [1/3]

```
Arguments::Arguments ( )
```

Construct a new [Arguments](#) object.

6.2.2.2 ~Arguments()

```
Arguments::~~Arguments ( ) [default]
```

Destroy the [Arguments](#) object.

6.2.2.3 Arguments() [2/3]

```
Arguments::Arguments (
    const Arguments & ) [delete]
```

Construct a new [Arguments](#) object by copy deleted.

6.2.2.4 Arguments() [3/3]

```
Arguments::Arguments (
    Arguments && ) [default]
```

Construct a new [Arguments](#) object by move.

6.2.3 Member Function Documentation

6.2.3.1 checkArgs()

```
int Arguments::checkArgs (
    const int argc,
    char *const argv[] )
```

Check the arguments.

Parameters

<i>argc</i>	The number of arguments
<i>argv</i>	The arguments

Returns

0 if no error, 84 otherwise

6.2.3.2 checkRessources()

```
void Arguments::checkRessources ( )
```

Check if the resources exist.

6.2.3.3 displayHelp()

```
void Arguments::displayHelp ( )
```

Display the help.

6.2.3.4 getHostname()

```
const char * Arguments::getHostname ( ) const
```

Get the hostname.

Returns

The hostname

6.2.3.5 getPort()

```
unsigned int Arguments::getPort ( ) const
```

Get the port.

Returns

The port

6.2.3.6 operator=() [1/2]

```
Arguments& Arguments::operator= (
    Arguments && ) [default]
```

Assign a [Arguments](#) object by move.

Returns

[Arguments&](#)

6.2.3.7 operator=() [2/2]

```
Arguments& Arguments::operator= (
    const Arguments & ) [delete]
```

Assign a [Arguments](#) object by copy deleted.

Returns

[Arguments&](#)

6.2.3.8 processArguments()

```
void Arguments::processArguments (
    const int argc,
    char *const argv[] )
```

Process the arguments.

Parameters

<i>argc</i>	The number of arguments
<i>argv</i>	The arguments

6.2.4 Field Documentation

6.2.4.1 hostname

```
const char * Arguments::hostname [private]
```

6.2.4.2 port

```
unsigned int Arguments::port [private]
```

6.2.4.3 resources

```
std::vector< std::string > Arguments::resources [private]
```

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

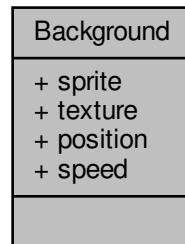
- [Arguments/Arguments.hpp](#)
- [Arguments/Arguments.cpp](#)

6.3 Background Struct Reference

[Background](#) struct.

```
#include <Parallax.hpp>
```

Collaboration diagram for Background:



Data Fields

- sf::Sprite [sprite](#)
- sf::Texture [texture](#)
- sf::Vector2f [position](#)
- float [speed](#)

6.3.1 Detailed Description

[Background](#) struct.

6.3.2 Field Documentation

6.3.2.1 position

```
sf::Vector2f Background::position
```

6.3.2.2 speed

```
float Background::speed
```

6.3.2.3 sprite

```
sf::Sprite Background::sprite
```

6.3.2.4 texture

```
sf::Texture Background::texture
```

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

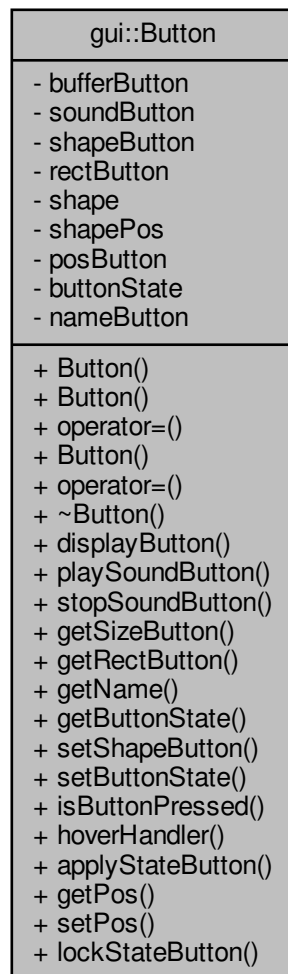
- Components/Parallax/[Parallax.hpp](#)

6.4 gui::Button Class Reference

[Button](#) class.

```
#include <Button.hpp>
```

Collaboration diagram for gui::Button:



Public Member Functions

- [Button](#) (const std::string &name, sf::IntRect [rectButton](#), sf::Vector2f position)
Construct a new [Button](#) object.
- [Button](#) (const [Button](#) &other)=delete
Construct a new [Button](#) object by copy.
- [Button](#) & [operator=](#) (const [Button](#) &other)=delete
Assign a button object by copy.
- [Button](#) ([Button](#) &&other)=default
Construct a new [Button](#) object by move.
- [Button](#) & [operator=](#) ([Button](#) &&other)=default
Assign a button object by move.
- [~Button](#) ()=default
Destroy the [Button](#) object.

- void [displayButton](#) ([Window](#) &window, [sf::Sprite](#) &sprite)
- void [playSoundButton](#) ()
- void [stopSoundButton](#) ()
- const [sf::Vector2f](#) & [getSizeButton](#) ()
- const [sf::Vector2f](#) [getRectButton](#) ()
- const [std::string](#) & [getName](#) ()
- const [ButtonState](#) & [getButtonState](#) ()
- void [setShapeButton](#) ([sf::Vector2f](#) shape)
- void [setButtonState](#) ([ButtonState](#) state)
- bool [isButtonPressed](#) ([sf::Vector2f](#) mousePos)
- void [hoverHandler](#) ([sf::Vector2f](#) mousePos)
- void [applyStateButton](#) ([sf::Sprite](#) &sprite)
- [sf::Vector2f](#) & [getPos](#) ()
- void [setPos](#) ([sf::Vector2f](#) pos)
Set the Pos of the button object.
- void [lockStateButton](#) ()
Lock the state of the button.

Private Attributes

- [sf::SoundBuffer](#) [bufferButton](#)
- [sf::Sound](#) [soundButton](#)
- [sf::RectangleShape](#) [shapeButton](#)
- [sf::IntRect](#) [rectButton](#)
- [sf::Vector2f](#) [shape](#)
- [sf::Vector2f](#) [shapePos](#)
- [sf::Vector2f](#) [posButton](#)
- [ButtonState](#) [buttonState](#)
- [std::string](#) [nameButton](#)

6.4.1 Detailed Description

[Button](#) class.

6.4.2 Constructor & Destructor Documentation

6.4.2.1 [Button\(\)](#) [1/3]

```
gui::Button::Button (
    const std::string & name,
    sf::IntRect rectButton,
    sf::Vector2f position )
```

Construct a new [Button](#) object.

Parameters

<i>name</i>	
<i>rectButton</i>	
<i>position</i>	

Exceptions

<i>Errors::ErrorTexture</i>	
---	--

6.4.2.2 Button() [2/3]

```
gui::Button::Button (
    const Button & other ) [delete]
```

Construct a new [Button](#) object by copy.

Parameters

<i>other</i>	
--------------	--

6.4.2.3 Button() [3/3]

```
gui::Button::Button (
    Button && other ) [default]
```

Construct a new [Button](#) object by move.

Parameters

<i>other</i>	
--------------	--

6.4.2.4 ~Button()

```
gui::Button::~~Button ( ) [default]
```

Destroy the [Button](#) object.

6.4.3 Member Function Documentation

6.4.3.1 applyStateButton()

```
void gui::Button::applyStateButton (
    sf::Sprite & sprite )
```

6.4.3.2 displayButton()

```
void gui::Button::displayButton (
    Window & window,
    sf::Sprite & sprite )
```

6.4.3.3 getButtonState()

```
const ButtonState & gui::Button::getButtonState ( )
```

6.4.3.4 getName()

```
const std::string & gui::Button::getName ( )
```

6.4.3.5 getPos()

```
sf::Vector2f & gui::Button::getPos ( )
```

6.4.3.6 getRectButton()

```
const sf::Vector2f gui::Button::getRectButton ( )
```

6.4.3.7 getSizeButton()

```
const sf::Vector2f & gui::Button::getSizeButton ( )
```

6.4.3.8 hoverHandler()

```
void gui::Button::hoverHandler (
    sf::Vector2f mousePos )
```

6.4.3.9 isButtonPressed()

```
bool gui::Button::isButtonPressed (
    sf::Vector2f mousePos )
```

6.4.3.10 lockStateButton()

```
void gui::Button::lockStateButton ( )
```

Lock the state of the button.

6.4.3.11 operator=() [1/2]

```
Button& gui::Button::operator= (
    Button && other ) [default]
```

Assign a button object by move.

Parameters

<i>other</i>	
--------------	--

Returns

Button&

6.4.3.12 operator=() [2/2]

```
Button& gui::Button::operator= (
    const Button & other ) [delete]
```

Assign a button object by copy.

Parameters

<i>other</i>	
--------------	--

Returns

[Button](#)&

6.4.3.13 playSoundButton()

```
void gui::Button::playSoundButton ( )
```

6.4.3.14 setButtonState()

```
void gui::Button::setButtonState (
    ButtonState state )
```

6.4.3.15 setPos()

```
void gui::Button::setPos (
    sf::Vector2f pos )
```

Set the Pos of the button object.

Parameters

<i>pos</i>	
------------	--

6.4.3.16 setShapeButton()

```
void gui::Button::setShapeButton (
    sf::Vector2f shape )
```

6.4.3.17 stopSoundButton()

```
void gui::Button::stopSoundButton ( )
```

6.4.4 Field Documentation

6.4.4.1 bufferButton

```
sf::SoundBuffer gui::Button::bufferButton [private]
```

6.4.4.2 buttonState

```
ButtonState gui::Button::buttonState [private]
```

6.4.4.3 nameButton

```
std::string gui::Button::nameButton [private]
```

6.4.4.4 posButton

```
sf::Vector2f gui::Button::posButton [private]
```

6.4.4.5 rectButton

```
sf::IntRect gui::Button::rectButton [private]
```

6.4.4.6 shape

```
sf::Vector2f gui::Button::shape [private]
```

6.4.4.7 shapeButton

```
sf::RectangleShape gui::Button::shapeButton [private]
```

6.4.4.8 shapePos

```
sf::Vector2f gui::Button::shapePos [private]
```

6.4.4.9 soundButton

```
sf::Sound gui::Button::soundButton [private]
```

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

- Components/Buttons/[Button.hpp](#)
- Components/Buttons/[Button.cpp](#)

6.5 gui::Camera Class Reference

[Camera](#) class.

```
#include <Camera.hpp>
```

Collaboration diagram for gui::Camera:

gui::Camera
- view - zoomFactor
+ Camera() + ~Camera() + Camera() + operator=() + Camera() + operator=() + setCenter() + getCenter() + setZoom() + getZoom() + setView() + setSize() + getView()

Public Member Functions

- [Camera](#) (unsigned int width, unsigned int height)
Construct a new [Camera](#) object.
- [~Camera](#) () noexcept=default
Destroy the [Camera](#) object.
- [Camera](#) (const [Camera](#) &cpy)=delete
Construct a new [Camera](#) object by copy.
- [Camera](#) & operator= (const [Camera](#) &src)=delete
Assign a [Camera](#) object by copy.
- [Camera](#) ([Camera](#) &&cpy)=default
Construct a new [Camera](#) object by move.
- [Camera](#) & operator= ([Camera](#) &&src)=default
Assign a [Camera](#) object by move.
- void [setCenter](#) (sf::Vector2f pos)
Set the Center of the camera Object.
- sf::Vector2f [getCenter](#) ()
Get the Center of the camera Object.
- void [setZoom](#) (float zoom)
Set the Zoom of the camera Object.
- const float & [getZoom](#) ()
Get the Zoom object.
- void [setView](#) (sf::RenderWindow &>window)
Set the View of the camera Object.
- void [setSize](#) (sf::Vector2f size)
Set the Size of the camera Object.
- sf::View & [getView](#) ()
Get the View of the camera Object.

Private Attributes

- sf::View [view](#)
- float [zoomFactor](#)

6.5.1 Detailed Description

[Camera](#) class.

6.5.2 Constructor & Destructor Documentation

6.5.2.1 Camera() [1/3]

```
gui::Camera::Camera (
    unsigned int width,
    unsigned int height )
```

Construct a new [Camera](#) object.

Parameters

<i>width</i>	
<i>height</i>	

6.5.2.2 ~Camera()

```
gui::Camera::~~Camera ( ) [default], [noexcept]
```

Destroy the [Camera](#) object.

6.5.2.3 Camera() [2/3]

```
gui::Camera::Camera (
    const Camera & cpy ) [delete]
```

Construct a new [Camera](#) object by copy.

Parameters

<i>cpy</i>	
------------	--

6.5.2.4 Camera() [3/3]

```
gui::Camera::Camera (
    Camera && cpy ) [default]
```

Construct a new [Camera](#) object by move.

Parameters

<i>cpy</i>	
------------	--

6.5.3 Member Function Documentation**6.5.3.1** getCenter()

```
sf::Vector2f gui::Camera::getCenter ( )
```

Get the Center of the camera Object.

Returns

sf::Vector2f

6.5.3.2 getView()

```
sf::View & gui::Camera::getView ( )
```

Get the View of the camera Object.

Returns

sf::View&

6.5.3.3 getZoom()

```
const float & gui::Camera::getZoom ( )
```

Get the Zoom object.

Returns

float

6.5.3.4 operator=() [1/2]

```
Camera& gui::Camera::operator= (
    Camera && src ) [default]
```

Assign a [Camera](#) object by move.

Parameters

<i>src</i>	
------------	--

Returns

[Camera](#)&

6.5.3.5 operator=() [2/2]

```
Camera& gui::Camera::operator= (
    const Camera & src ) [delete]
```

Assign a [Camera](#) object by copy.

Parameters

<i>src</i>	
------------	--

Returns

[Camera&](#)

6.5.3.6 setCenter()

```
void gui::Camera::setCenter (
    sf::Vector2f pos )
```

Set the Center of the camera Object.

Parameters

<i>pos</i>	
------------	--

6.5.3.7 setSize()

```
void gui::Camera::setSize (
    sf::Vector2f size )
```

Set the Size of the camera Object.

Parameters

<i>size</i>	
-------------	--

6.5.3.8 setView()

```
void gui::Camera::setView (
    sf::RenderWindow & window )
```

Set the View of the camera Object.

Parameters

<i>window</i>	
---------------	--

6.5.3.9 setZoom()

```
void gui::Camera::setZoom (
    float zoom )
```

Set the Zoom of the camera Object.

Parameters

<i>zoom</i>	
-------------	--

6.5.4 Field Documentation**6.5.4.1 view**

```
sf::View gui::Camera::view [private]
```

6.5.4.2 zoomFactor

```
float gui::Camera::zoomFactor [private]
```

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

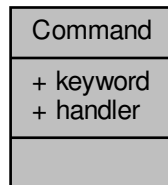
- Camera/[Camera.hpp](#)
- Camera/[Camera.cpp](#)

6.6 Command Struct Reference

[Command](#) struct.

```
#include <Handler.hpp>
```

Collaboration diagram for Command:



Data Fields

- `std::string` [keyword](#)
- `std::function< void(std::istream&, gui::Data &)>` [handler](#)

6.6.1 Detailed Description

[Command](#) struct.

6.6.2 Field Documentation

6.6.2.1 handler

```
std::function<void(std::istream&, gui::Data &)> Command::handler
```

6.6.2.2 keyword

```
std::string Command::keyword
```

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

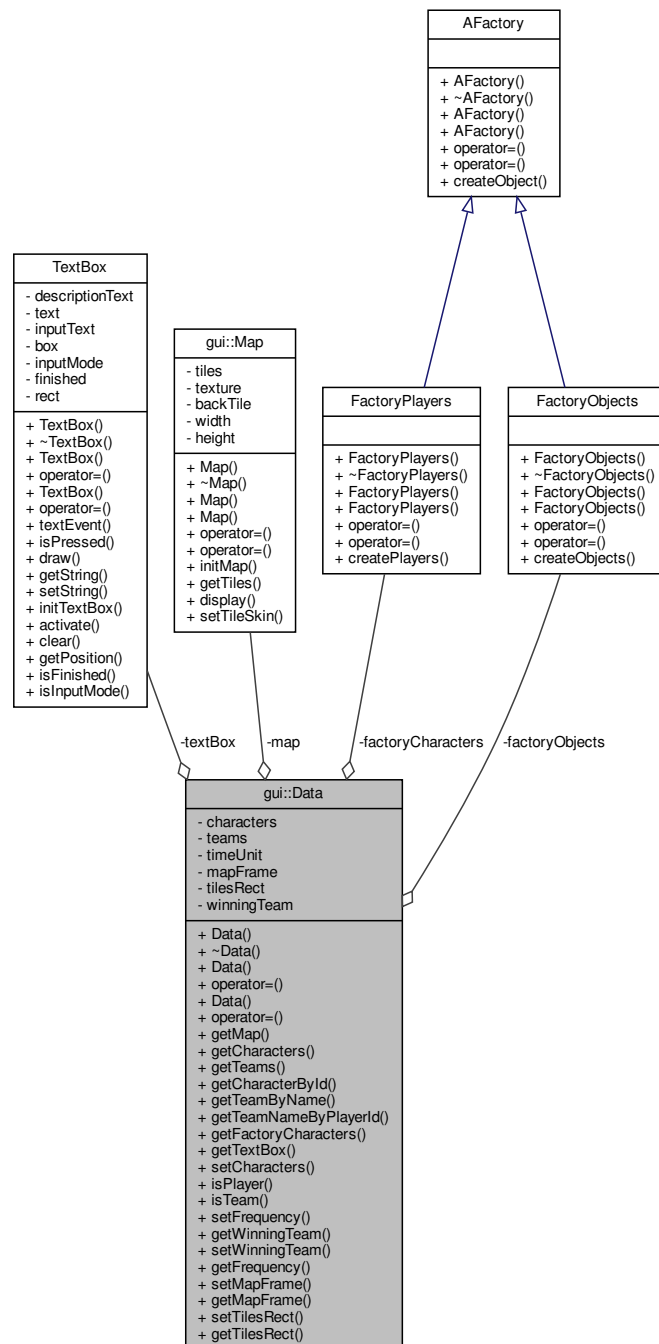
- Network/Handlers/[Handler.hpp](#)

6.7 gui::Data Class Reference

Data class.

```
#include <Data.hpp>
```

Collaboration diagram for gui::Data:



Public Member Functions

- [Data](#) ()=default
Construct a new [Data](#) object.
- [~Data](#) () noexcept=default
Destroy the [Data](#) object.
- [Data](#) ([Data](#) &&)=default
Construct a new [Data](#) object by move.
- [Data](#) & operator= ([Data](#) &&)=default
Assign a [Data](#) object by move.
- [Data](#) (const [Data](#) &)=delete
Construct a new [Data](#) object by copy.
- [Data](#) & operator= (const [Data](#) &)=delete
Assign a [Data](#) object by copy.
- [Map](#) & [getMap](#) ()
Get the [Map](#) object.
- std::vector< std::unique_ptr< [ICharacters](#) > > & [getCharacters](#) ()
Get the Characters.
- std::vector< [gui::Team](#) > & [getTeams](#) ()
Get the Teams.
- [ICharacters](#) & [getCharacterById](#) (int id)
Get the Character By Id.
- [gui::Team](#) & [getTeamByName](#) (const std::string &name)
Get the [Team](#) By Name.
- const std::string & [getTeamNameById](#) (int id)
Get the [Team](#) Name By Player Id.
- [FactoryPlayers](#) & [getFactoryCharacters](#) ()
Get the Factory Characters.
- [TextBox](#) & [getTextBox](#) ()
Get the Text Box.
- void [setCharacters](#) (std::vector< std::unique_ptr< [ICharacters](#) > > &characters)
Set the Characters.
- bool [isPlayer](#) (int id)
Check if the player is in the game.
- bool [isTeam](#) (const std::string &name)
Check if the team is in the game.
- void [setFrequency](#) (int frequency)
Set the Frequency.
- const std::string & [getWinningTeam](#) ()
Get the Winning [Team](#).
- void [setWinningTeam](#) (const std::string &team)
Set the Winning [Team](#).
- const int & [getFrequency](#) ()
Get the Frequency.
- void [setMapFrame](#) (int frame)
Set the [Map](#) Frame.
- const int & [getMapFrame](#) ()
Get the [Map](#) Frame.
- void [setTilesRect](#) (int rect)
Set the Tiles Rect.
- const int & [getTilesRect](#) ()
Get the Tiles Rect.

Private Attributes

- `gui::Map` `map`
- `std::vector< std::unique_ptr< ICharacters > >` `characters`
- `FactoryObjects` `factoryObjects`
- `FactoryPlayers` `factoryCharacters`
- `std::vector< gui::Team >` `teams`
- `int` `timeUnit`
- `int` `mapFrame` {0}
- `int` `tilesRect` {0}
- `std::string` `winningTeam` {""}
- `TextBox` `textBox`

6.7.1 Detailed Description

`Data` class.

6.7.2 Constructor & Destructor Documentation

6.7.2.1 `Data()` [1/3]

```
gui::Data::Data ( ) [default]
```

Construct a new `Data` object.

6.7.2.2 `~Data()`

```
gui::Data::~~Data ( ) [default], [noexcept]
```

Destroy the `Data` object.

6.7.2.3 `Data()` [2/3]

```
gui::Data::Data (
    Data && ) [default]
```

Construct a new `Data` object by move.

6.7.2.4 Data() [3/3]

```
gui::Data::Data (
    const Data & ) [delete]
```

Construct a new [Data](#) object by copy.

6.7.3 Member Function Documentation

6.7.3.1 getCharacterById()

```
ICharacters & gui::Data::getCharacterById (
    int id )
```

Get the Character By Id.

Parameters

<i>id</i>	
-----------	--

Returns

[ICharacters](#)&

Exceptions

Errors::ErrorCharacter	
--	--

6.7.3.2 getCharacters()

```
std::vector< std::unique_ptr< ICharacters > > & gui::Data::getCharacters ( )
```

Get the Characters.

Returns

std::vector<std::unique_ptr<ICharacters>>&

6.7.3.3 getFactoryCharacters()

```
FactoryPlayers & gui::Data::getFactoryCharacters ( )
```

Get the Factory Characters.

Returns

FactoryPlayers&

6.7.3.4 getFrequency()

```
const int & gui::Data::getFrequency ( )
```

Get the Frequency.

Returns

int

6.7.3.5 getMap()

```
gui::Map & gui::Data::getMap ( )
```

Get the Map object.

Returns

Map&

6.7.3.6 getMapFrame()

```
const int & gui::Data::getMapFrame ( )
```

Get the Map Frame.

Returns

int

6.7.3.7 getTeamByName()

```
gui::Team & gui::Data::getTeamByName (
    const std::string & name )
```

Get the Team By Name.

Parameters

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

Returns

[gui::Team](#)&

Exceptions

Errors::ErrorTeam	
-----------------------------------	--

6.7.3.8 getTeamNameById()

```
const std::string & gui::Data::getTeamNameById (
    int id )
```

Get the [Team](#) Name By Player Id.

Parameters

<i>id</i>	
-----------	--

Returns

const std::string&

Exceptions

Errors::ErrorTeam	
-----------------------------------	--

6.7.3.9 getTeams()

```
std::vector< gui::Team > & gui::Data::getTeams ( )
```

Get the Teams.

Returns

std::vector<[gui::Team](#)>&

6.7.3.10 `getTextBox()`

```
TextBox & gui::Data::getTextBox ( )
```

Get the Text Box.

Returns

TextBox&

6.7.3.11 `getTilesRect()`

```
const int & gui::Data::getTilesRect ( )
```

Get the Tiles Rect.

Returns

int

6.7.3.12 `getWinningTeam()`

```
const std::string & gui::Data::getWinningTeam ( )
```

Get the Winning Team.

Returns

std::string&

6.7.3.13 `isPlayer()`

```
bool gui::Data::isPlayer (
    int id )
```

Check if the player is in the game.

Parameters

<i>id</i>	
-----------	--

Returns

true
false

6.7.3.14 isTeam()

```
bool gui::Data::isTeam (
    const std::string & name )
```

Check if the team is in the game.

Parameters

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

Returns

true
false

6.7.3.15 operator=() [1/2]

```
Data& gui::Data::operator= (
    const Data & ) [delete]
```

Assign a [Data](#) object by copy.

Returns

[Data&](#)

6.7.3.16 operator=() [2/2]

```
Data& gui::Data::operator= (
    Data && ) [default]
```

Assign a [Data](#) object by move.

Returns

[Data&](#)

6.7.3.17 setCharacters()

```
void gui::Data::setCharacters (
    std::vector< std::unique_ptr< ICharacters >> & characters )
```

Set the Characters.

Parameters

<i>characters</i>	
-------------------	--

6.7.3.18 setFrequency()

```
void gui::Data::setFrequency (
    int frequency )
```

Set the Frequency.

Parameters

<i>frequency</i>	
------------------	--

6.7.3.19 setMapFrame()

```
void gui::Data::setMapFrame (
    int frame )
```

Set the [Map](#) Frame.

Parameters

<i>frame</i>	
--------------	--

6.7.3.20 setTilesRect()

```
void gui::Data::setTilesRect (
    int rect )
```

Set the Tiles Rect.

Parameters

<i>rect</i>	
-------------	--

6.7.3.21 setWinningTeam()

```
void gui::Data::setWinningTeam (
    const std::string & team )
```

Set the Winning [Team](#).

Parameters

<i>team</i>	
-------------	--

6.7.4 Field Documentation

6.7.4.1 characters

```
std::vector<std::unique_ptr<ICharacters> > gui::Data::characters [private]
```

6.7.4.2 factoryCharacters

```
FactoryPlayers gui::Data::factoryCharacters [private]
```

6.7.4.3 factoryObjects

```
FactoryObjects gui::Data::factoryObjects [private]
```

6.7.4.4 map

```
gui::Map gui::Data::map [private]
```

6.7.4.5 mapFrame

```
int gui::Data::mapFrame {0} [private]
```

6.7.4.6 teams

```
std::vector< gui::Team > gui::Data::teams [private]
```

6.7.4.7 textBox

```
TextBox gui::Data::textBox [private]
```

6.7.4.8 tilesRect

```
gui::Data::tilesRect {0} [private]
```

6.7.4.9 timeUnit

```
int gui::Data::timeUnit [private]
```

6.7.4.10 winningTeam

```
std::string gui::Data::winningTeam {""} [private]
```

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

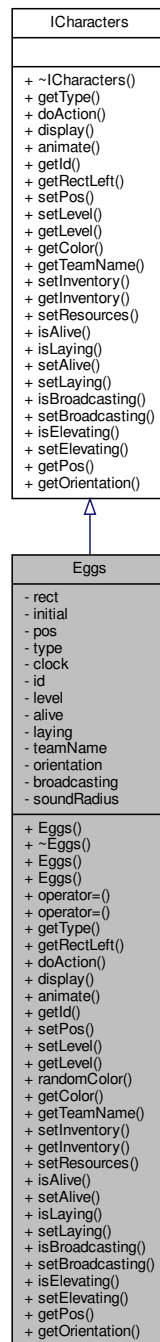
- Database/[Data.hpp](#)
- Database/[Data.cpp](#)

6.8 Eggs Class Reference

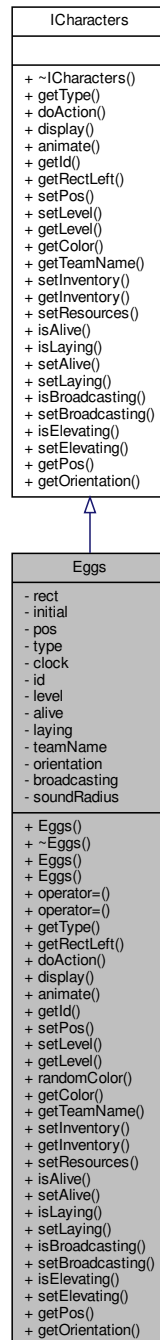
[Eggs](#) class.

```
#include <Eggs.hpp>
```

Inheritance diagram for Eggs:



Collaboration diagram for Eggs:



Public Member Functions

- **Eggs** (sf::Vector2f **pos**, int **id**, std::string **teamName**, gui::Data &data, Orientation **orientation**)
Construct a new **Eggs** object.
- **~Eggs** () noexcept override=default
Destroy the **Eggs** object.
- **Eggs** (const **Eggs** &other)=delete

- Construct a new *Eggs* object by copy deleted.

 - *Eggs* (*Eggs* &&other)=default

Construct a new *Eggs* object by move.

 - *Eggs* & operator= (const *Eggs* &other)=delete

Assign a *Eggs* object by copy deleted.

 - *Eggs* & operator= (*Eggs* &&other)=default

Assign a *Eggs* object by move.

 - *CharacterType* *getType* () const override

Get the *Type* of the *ICcharacters*.

 - const int & *getRectLeft* () override

Get the *Rect Left* of the *Eggs*.

 - void *doAction* () override

Do specific action for *Eggs*.

 - void *display* (sf::RenderWindow &>window, sf::Sprite &sprite, *gui::ItemDrawer* &drawer, int timeUnit, *SoundBox* &soundBox) override

Display the *Eggs*.

 - void *animate* (sf::Sprite &sprite) override

Animate the *Eggs*.

 - const int & *getId* () override

Get the *Id* of the *Eggs*.

 - void *setPos* (sf::Vector2f *pos*, Orientation *orientation*) override

Set the *Pos* of the *Eggs*.

 - void *setLevel* (int *level*) override

Set the *Level* of the *Eggs*.

 - const int & *getLevel* () override

Get the *Level* of the *Eggs*.

 - *TypeEggs* *randomColor* (*gui::Data* &data)

Choose a random color for the *Eggs*.

 - const std::string & *getColor* () override

Get the *Color* of the *Eggs*.

 - const std::string & *getTeamName* () override

Get the *Team Name* of the *Eggs*.

 - void *setInventory* (const std::array< int, 7 > &res) override

Set the *Inventory* of the *Eggs*.

 - const std::array< int, 7 > & *getInventory* () override

Get the *Inventory* of the *Eggs*.

 - void *setResources* (const int &res, const int &nb) override

Set the *Resources* of the *Eggs*.

 - const bool & *isAlive* () override

Check if the *Eggs* is alive.

 - void *setAlive* (bool *alive*) override

Set the *Alive* boolean of the *Eggs*.

 - const bool & *isLaying* () override

Check if the *Eggs* is laying.

 - void *setLaying* (bool *laying*) override

Set the *Laying* boolean of the *Eggs*.

 - const bool & *isBroadcasting* () override

Check if the *Eggs* is broadcasting.

 - void *setBroadcasting* (bool *broadcastingParams*) override

Set the *Broadcasting* boolean of the *Eggs*.

 - const int & *isElevating* () override

- Check if the [Eggs](#) is elevating.
- void [setElevating](#) (int elevating) override
Set the Elevating process of the [Eggs](#).
- const sf::Vector2f & [getPos](#) () override
Get the Pos of the [Eggs](#).
- const Orientation & [getOrientation](#) () override
Get the Orientation of the [Eggs](#).

Private Attributes

- sf::IntRect [rect](#)
- sf::IntRect [initial](#)
- sf::Vector2f [pos](#)
- [TypeEggs](#) type
- sf::Clock [clock](#)
- int [id](#)
- int [level](#)
- bool [alive](#)
- bool [laying](#)
- std::string [teamName](#)
- Orientation [orientation](#)
- bool [broadcasting](#)
- int [soundRadius](#)

6.8.1 Detailed Description

[Eggs](#) class.

6.8.2 Constructor & Destructor Documentation

6.8.2.1 Eggs() [1/3]

```
Eggs::Eggs (
    sf::Vector2f pos,
    int id,
    std::string teamName,
    gui::Data & data,
    Orientation orientation )
```

Construct a new [Eggs](#) object.

Parameters

<i>pos</i>	
<i>id</i>	
<i>teamName</i>	
<i>data</i>	
<i>orientation</i>	

6.8.2.2 ~Eggs()

```
Eggs::~Eggs ( ) [override], [default], [noexcept]
```

Destroy the [Eggs](#) object.

6.8.2.3 Eggs() [2/3]

```
Eggs::Eggs (
    const Eggs & other ) [delete]
```

Construct a new [Eggs](#) object by copy deleted.

Parameters

<i>other</i>	
--------------	--

6.8.2.4 Eggs() [3/3]

```
Eggs::Eggs (
    Eggs && other ) [default]
```

Construct a new [Eggs](#) object by move.

Parameters

<i>other</i>	
--------------	--

6.8.3 Member Function Documentation

6.8.3.1 animate()

```
void Eggs::animate (
    sf::Sprite & sprite ) [override], [virtual]
```

Animate the [Eggs](#).

Parameters

<i>sprite</i>	
---------------	--

Implements [ICharacters](#).

6.8.3.2 display()

```
void Eggs::display (
    sf::RenderWindow & window,
    sf::Sprite & sprite,
    gui::ItemDrawer & drawer,
    int timeUnit,
    SoundBox & soundBox ) [override], [virtual]
```

Display the [Eggs](#).

Parameters

<i>window</i>	
<i>sprite</i>	
<i>drawer</i>	
<i>timeUnit</i>	
<i>soundbox</i>	

Implements [ICharacters](#).

6.8.3.3 doAction()

```
void Eggs::doAction ( ) [override], [virtual]
```

Do specific action for [Eggs](#).

Implements [ICharacters](#).

6.8.3.4 getColor()

```
const std::string & Eggs::getColor ( ) [override], [virtual]
```

Get the Color of the [Eggs](#).

Returns

```
const std::string&
```

Exceptions

Errors::ErrorColor	
------------------------------------	--

Implements [ICharacters](#).

6.8.3.5 getId()

```
const int & Eggs::getId ( ) [override], [virtual]
```

Get the Id of the [Eggs](#).

Returns

const int&

Implements [ICharacters](#).

6.8.3.6 getInventory()

```
const std::array< int, 7 > & Eggs::getInventory ( ) [override], [virtual]
```

Get the Inventory of the [Eggs](#).

Returns

std::array<int, 7>&

Exceptions

Errors::ErrorInventory	
--	--

Implements [ICharacters](#).

6.8.3.7 getLevel()

```
const int & Eggs::getLevel ( ) [override], [virtual]
```

Get the Level of the [Eggs](#).

Returns

const int&

Implements [ICharacters](#).

6.8.3.8 getOrientation()

```
const Orientation & Eggs::getOrientation ( ) [override], [virtual]
```

Get the Orientation of the [Eggs](#).

Returns

const Orientation&

Implements [ICharacters](#).

6.8.3.9 getPos()

```
const sf::Vector2f & Eggs::getPos ( ) [override], [virtual]
```

Get the Pos of the [Eggs](#).

Returns

sf::Vector2f

Implements [ICharacters](#).

6.8.3.10 getRectLeft()

```
const int & Eggs::getRectLeft ( ) [override], [virtual]
```

Get the Rect Left of the [Eggs](#).

Returns

const int&

Implements [ICharacters](#).

6.8.3.11 getTeamName()

```
const std::string & Eggs::getTeamName ( ) [override], [virtual]
```

Get the Team Name of the [Eggs](#).

Returns

const std::string&

Implements [ICharacters](#).

6.8.3.12 getType()

```
CharacterType Eggs::getType ( ) const [override], [virtual]
```

Get the Type of the [ICharacters](#).

Returns

CharacterType

Implements [ICharacters](#).

6.8.3.13 isAlive()

```
const bool & Eggs::isAlive ( ) [override], [virtual]
```

Check if the [Eggs](#) is alive.

Returns

true

false

Implements [ICharacters](#).

6.8.3.14 isBroadcasting()

```
const bool & Eggs::isBroadcasting ( ) [override], [virtual]
```

Check if the [Eggs](#) is broadcasting.

Returns

true

false

Implements [ICharacters](#).

6.8.3.15 isElevating()

```
const int & Eggs::isElevating ( ) [override], [virtual]
```

Check if the [Eggs](#) is elevating.

Returns

int

Exceptions

Errors::ErrorElevate	
--------------------------------------	--

Implements [ICharacters](#).

6.8.3.16 isLaying()

```
const bool & Eggs::isLaying ( ) [override], [virtual]
```

Check if the [Eggs](#) is laying.

Returns

true

false

Implements [ICharacters](#).

6.8.3.17 operator=() [1/2]

```
Eggs& Eggs::operator= (
    const Eggs & other ) [delete]
```

Assign a [Eggs](#) object by copy deleted.

Parameters

<i>other</i>	
--------------	--

Returns

[Eggs](#)&

6.8.3.18 operator=() [2/2]

```
Eggs& Eggs::operator= (
    Eggs && other ) [default]
```

Assign a [Eggs](#) object by move.

Parameters

<i>other</i>	
--------------	--

Returns

[Eggs&](#)

6.8.3.19 randomColor()

```
TypeEggs Eggs::randomColor (
    gui::Data & data )
```

Choose a random color for the [Eggs](#).

Parameters

<i>data</i>	
-------------	--

Returns

TypeEggs

6.8.3.20 setAlive()

```
void Eggs::setAlive (
    bool alive ) [override], [virtual]
```

Set the Alive boolean of the [Eggs](#).

Parameters

<i>alive</i>	
--------------	--

Implements [ICharacters](#).

6.8.3.21 setBroadcasting()

```
void Eggs::setBroadcasting (
    bool broadcastingParams ) [override], [virtual]
```

Set the Broadcasting boolean of the [Eggs](#).

Parameters

<i>broadcasting</i>	
---------------------	--

Implements [ICharacters](#).

6.8.3.22 setElevating()

```
void Eggs::setElevating (
    int elevating ) [override], [virtual]
```

Set the Elevating process of the [Eggs](#).

Parameters

<i>elevating</i>	
------------------	--

Exceptions

Errors::ErrorElevate	
--------------------------------------	--

Implements [ICharacters](#).

6.8.3.23 setInventory()

```
void Eggs::setInventory (
    const std::array< int, 7 > & res ) [override], [virtual]
```

Set the Inventory of the [Eggs](#).

Parameters

<i>res</i>	
------------	--

Implements [ICharacters](#).

6.8.3.24 setLaying()

```
void Eggs::setLaying (
    bool laying ) [override], [virtual]
```

Set the Laying boolean of the [Eggs](#).

Parameters

<i>laying</i>	
---------------	--

Implements [ICharacters](#).

6.8.3.25 setLevel()

```
void Eggs::setLevel (
    int level ) [override], [virtual]
```

Set the Level of the [Eggs](#).

Parameters

<i>level</i>	
--------------	--

Implements [ICharacters](#).

6.8.3.26 setPos()

```
void Eggs::setPos (
    sf::Vector2f pos,
    Orientation orientation ) [override], [virtual]
```

Set the Pos of the [Eggs](#).

Parameters

<i>pos</i>	
<i>orientation</i>	

Implements [ICharacters](#).

6.8.3.27 setResources()

```
void Eggs::setResources (
```

```
const int & res,  
const int & nb ) [override], [virtual]
```

Set the Resources of the [Eggs](#).

Parameters

<i>res</i>	
<i>nb</i>	

Implements [ICharacters](#).

6.8.4 Field Documentation

6.8.4.1 alive

```
bool Eggs::alive [private]
```

6.8.4.2 broadcasting

```
bool Eggs::broadcasting [private]
```

6.8.4.3 clock

```
sf::Clock Eggs::clock [private]
```

6.8.4.4 id

```
int Eggs::id [private]
```

6.8.4.5 initial

```
sf::IntRect Eggs::initial [private]
```

6.8.4.6 laying

```
bool Eggs::laying [private]
```

6.8.4.7 level

```
int Eggs::level [private]
```

6.8.4.8 orientation

```
Orientation Eggs::orientation [private]
```

6.8.4.9 pos

```
sf::Vector2f Eggs::pos [private]
```

6.8.4.10 rect

```
sf::IntRect Eggs::rect [private]
```

6.8.4.11 soundRadius

```
int Eggs::soundRadius [private]
```

6.8.4.12 teamName

```
std::string Eggs::teamName [private]
```

6.8.4.13 type

```
TypeEggs Eggs::type [private]
```

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

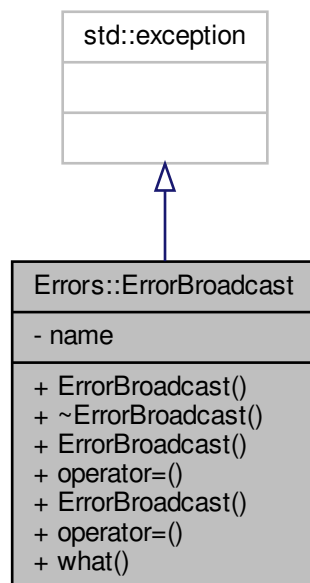
- Characters/Eggs/[Eggs.hpp](#)
- Characters/Eggs/[Eggs.cpp](#)

6.9 Errors::ErrorBroadcast Class Reference

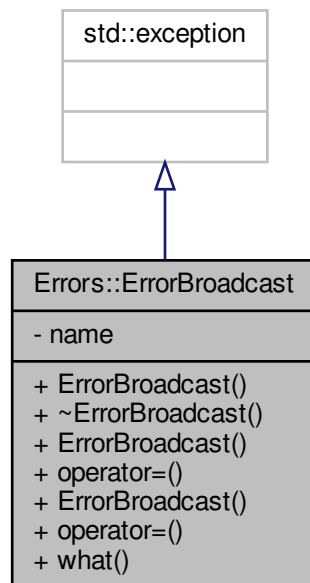
A class for broadcast Error.

```
#include <Errors.hpp>
```

Inheritance diagram for Errors::ErrorBroadcast:



Collaboration diagram for Errors::ErrorBroadcast:



Public Member Functions

- [ErrorBroadcast](#) (std::string m_what)
Constructor a new Error Broadcast object.
- [~ErrorBroadcast](#) () noexcept override=default
Destructor a new Error Broadcast object.
- [ErrorBroadcast](#) (const [ErrorBroadcast](#) &other)=delete
Construct a new Error Broadcast object by copy.
- [ErrorBroadcast](#) & [operator=](#) (const [ErrorBroadcast](#) &other)=delete
Assign a [ErrorBroadcast](#) object by copy.
- [ErrorBroadcast](#) ([ErrorBroadcast](#) &&other)=default
Construct a new [ErrorBroadcast](#) object by move.
- [ErrorBroadcast](#) & [operator=](#) ([ErrorBroadcast](#) &&other)=default
Assign a [ErrorBroadcast](#) object by move.
- const char * [what](#) () const noexcept override
Get the What object.

Private Attributes

- std::string [name](#)

6.9.1 Detailed Description

A class for broadcast Error.

6.9.2 Constructor & Destructor Documentation

6.9.2.1 ErrorBroadcast() [1/3]

```
Errors::ErrorBroadcast::ErrorBroadcast (
    std::string m_what ) [explicit]
```

Constructor a new Error Broadcast object.

Parameters

<i>m_what</i>	The error message
---------------	-------------------

6.9.2.2 ~ErrorBroadcast()

```
Errors::ErrorBroadcast::~~ErrorBroadcast ( ) [override], [default], [noexcept]
```

Destructor a new Error Broadcast object.

6.9.2.3 ErrorBroadcast() [2/3]

```
Errors::ErrorBroadcast::ErrorBroadcast (
    const ErrorBroadcast & other ) [delete]
```

Construct a new Error Broadcast object by copy.

Parameters

<i>other</i>	
--------------	--

6.9.2.4 ErrorBroadcast() [3/3]

```
Errors::ErrorBroadcast::ErrorBroadcast (
    ErrorBroadcast && other ) [default]
```

Construct a new [ErrorBroadcast](#) object by move.

Parameters

<i>other</i>	
--------------	--

6.9.3 Member Function Documentation

6.9.3.1 operator=() [1/2]

```
ErrorBroadcast& Errors::ErrorBroadcast::operator= (
    const ErrorBroadcast & other ) [delete]
```

Assign a [ErrorBroadcast](#) object by copy.

Parameters

<i>other</i>	
--------------	--

Returns

[ErrorBroadcast&](#)

6.9.3.2 operator=() [2/2]

```
ErrorBroadcast& Errors::ErrorBroadcast::operator= (
    ErrorBroadcast && other ) [default]
```

Assign a [ErrorBroadcast](#) object by move.

Parameters

<i>other</i>	
--------------	--

Returns

[ErrorBroadcast&](#)

6.9.3.3 what()

```
const char * Errors::ErrorBroadcast::what ( ) const [override], [noexcept]
```

Get the What object.

Returns

const char* The error message

6.9.4 Field Documentation

6.9.4.1 name

```
std::string Errors::ErrorBroadcast::name [private]
```

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

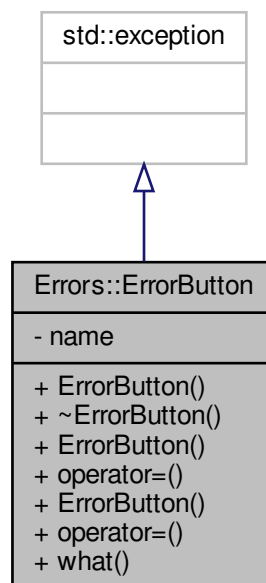
- Errors/[Errors.hpp](#)
- Errors/[Errors.cpp](#)

6.10 Errors::ErrorButton Class Reference

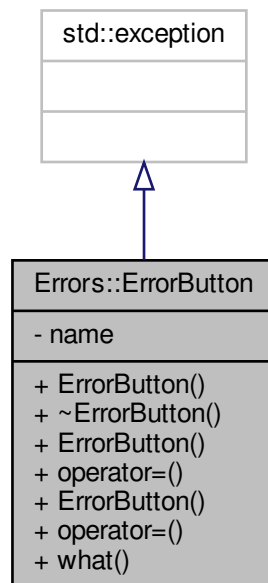
A class for Button Error.

```
#include <Errors.hpp>
```

Inheritance diagram for Errors::ErrorButton:



Collaboration diagram for Errors::ErrorButton:



Public Member Functions

- [ErrorButton](#) (std::string m_what)
Constructor a new Error Button object.
- [~ErrorButton](#) () noexcept override=default
Destructor a new Error Button object.
- [ErrorButton](#) (const [ErrorButton](#) &other)=delete
Construct a new Error Button object by copy.
- [ErrorButton](#) & [operator=](#) (const [ErrorButton](#) &other)=delete
Assign a [ErrorButton](#) object by copy.
- [ErrorButton](#) ([ErrorButton](#) &&other)=default
Construct a new [ErrorButton](#) object by move.
- [ErrorButton](#) & [operator=](#) ([ErrorButton](#) &&other)=default
Assign a [ErrorButton](#) object by move.
- const char * [what](#) () const noexcept override
Get the What object.

Private Attributes

- std::string [name](#)

6.10.1 Detailed Description

A class for Button Error.

6.10.2 Constructor & Destructor Documentation

6.10.2.1 `ErrorButton()` [1/3]

```
Errors::ErrorButton::ErrorButton (
    std::string m_what ) [explicit]
```

Constructor a new Error Button object.

Parameters

<i>m_what</i>	The error message
---------------	-------------------

6.10.2.2 `~ErrorButton()`

```
Errors::ErrorButton::~~ErrorButton ( ) [override], [default], [noexcept]
```

Destructor a new Error Button object.

6.10.2.3 `ErrorButton()` [2/3]

```
Errors::ErrorButton::ErrorButton (
    const ErrorButton & other ) [delete]
```

Construct a new Error Button object by copy.

Parameters

<i>other</i>	
--------------	--

6.10.2.4 `ErrorButton()` [3/3]

```
Errors::ErrorButton::ErrorButton (
    ErrorButton && other ) [default]
```

Construct a new [ErrorButton](#) object by move.

Parameters

<i>other</i>	
--------------	--

6.10.3 Member Function Documentation

6.10.3.1 operator=() [1/2]

```
ErrorButton& Errors::ErrorButton::operator= (
    const ErrorButton & other ) [delete]
```

Assign a [ErrorButton](#) object by copy.

Parameters

<i>other</i>	
--------------	--

Returns

[ErrorButton&](#)

6.10.3.2 operator=() [2/2]

```
ErrorButton& Errors::ErrorButton::operator= (
    ErrorButton && other ) [default]
```

Assign a [ErrorButton](#) object by move.

Parameters

<i>other</i>	
--------------	--

Returns

[ErrorButton&](#)

6.10.3.3 what()

```
const char * Errors::ErrorButton::what ( ) const [override], [noexcept]
```

Get the What object.

Returns

const char* The error message

6.10.4 Field Documentation

6.10.4.1 name

```
std::string Errors::ErrorButton::name [private]
```

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

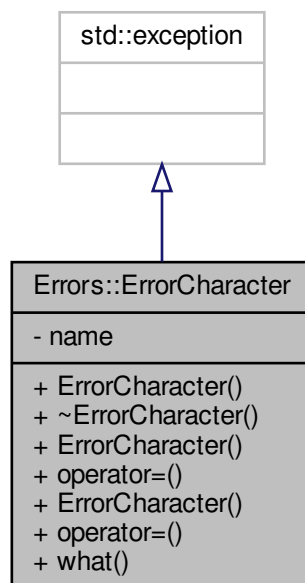
- [Errors/Errors.hpp](#)
- [Errors/Errors.cpp](#)

6.11 Errors::ErrorCharacter Class Reference

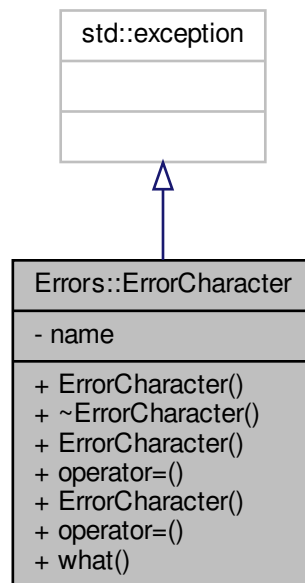
A class for Character Error.

```
#include <Errors.hpp>
```

Inheritance diagram for Errors::ErrorCharacter:



Collaboration diagram for Errors::ErrorCharacter:



Public Member Functions

- `ErrorCharacter` (`std::string m_what`)
Constructor a new Error Character object.
- `~ErrorCharacter` () noexcept override=default
Destructor a new Error Character object.
- `ErrorCharacter` (const `ErrorCharacter` &other)=delete
Construct a new Error Character object by copy.
- `ErrorCharacter` & `operator=` (const `ErrorCharacter` &other)=delete
Assign a `ErrorCharacter` object by copy.
- `ErrorCharacter` (`ErrorCharacter` &&other)=default
Construct a new `ErrorCharacter` object by move.
- `ErrorCharacter` & `operator=` (`ErrorCharacter` &&other)=default
Assign a `ErrorCharacter` object by move.
- const char * `what` () const noexcept override
Get the What object.

Private Attributes

- `std::string name`

6.11.1 Detailed Description

A class for Character Error.

6.11.2 Constructor & Destructor Documentation

6.11.2.1 ErrorCharacter() [1/3]

```
Errors::ErrorCharacter::ErrorCharacter (
    std::string m_what ) [explicit]
```

Constructor a new Error Character object.

Parameters

<i>m_what</i>	The error message
---------------	-------------------

6.11.2.2 ~ErrorCharacter()

```
Errors::ErrorCharacter::~~ErrorCharacter ( ) [override], [default], [noexcept]
```

Destructor a new Error Character object.

6.11.2.3 ErrorCharacter() [2/3]

```
Errors::ErrorCharacter::ErrorCharacter (
    const ErrorCharacter & other ) [delete]
```

Construct a new Error Character object by copy.

Parameters

<i>other</i>	
--------------	--

6.11.2.4 ErrorCharacter() [3/3]

```
Errors::ErrorCharacter::ErrorCharacter (
    ErrorCharacter && other ) [default]
```

Construct a new [ErrorCharacter](#) object by move.

Parameters

<i>other</i>	
--------------	--

6.11.3 Member Function Documentation

6.11.3.1 operator=() [1/2]

```
ErrorCharacter& Errors::ErrorCharacter::operator= (
    const ErrorCharacter & other ) [delete]
```

Assign a [ErrorCharacter](#) object by copy.

Parameters

<i>other</i>	
--------------	--

Returns

[ErrorCharacter&](#)

6.11.3.2 operator=() [2/2]

```
ErrorCharacter& Errors::ErrorCharacter::operator= (
    ErrorCharacter && other ) [default]
```

Assign a [ErrorCharacter](#) object by move.

Parameters

<i>other</i>	
--------------	--

Returns

[ErrorCharacter&](#)

6.11.3.3 what()

```
const char * Errors::ErrorCharacter::what ( ) const [override], [noexcept]
```

Get the What object.

Returns

const char* The error message

6.11.4 Field Documentation**6.11.4.1 name**

```
std::string Errors::ErrorCharacter::name [private]
```

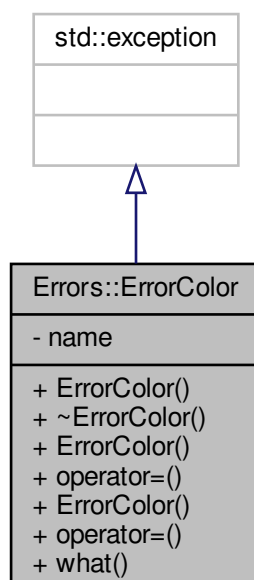
The documentation for this class was generated from the following files:

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

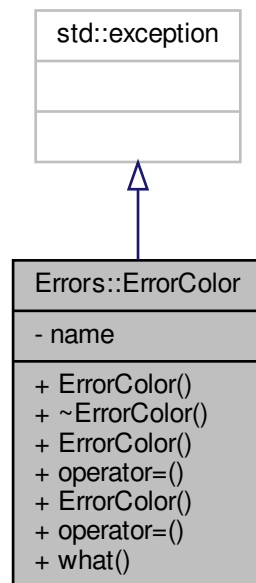
6.12 Errors::ErrorColor Class Reference

```
#include <Errors.hpp>
```

Inheritance diagram for Errors::ErrorColor:



Collaboration diagram for Errors::ErrorColor:



Public Member Functions

- `ErrorColor` (`std::string m_what`)
Constructor a new Error Color object.
- `~ErrorColor` () noexcept override=default
Destructor a new Error Color object.
- `ErrorColor` (const `ErrorColor` &other)=delete
Construct a new Error Color object by copy.
- `ErrorColor` & `operator=` (const `ErrorColor` &other)=delete
Assign a `ErrorColor` object by copy.
- `ErrorColor` (`ErrorColor` &&other)=default
Construct a new `ErrorColor` object by move.
- `ErrorColor` & `operator=` (`ErrorColor` &&other)=default
Assign a `ErrorColor` object by move.
- const char * `what` () const noexcept override
Get the What object.

Private Attributes

- std::string `name`

6.12.1 Constructor & Destructor Documentation

6.12.1.1 ErrorColor() [1/3]

```
Errors::ErrorColor::ErrorColor (
    std::string m_what ) [explicit]
```

Constructor a new Error Color object.

Parameters

<i>m_what</i>	The error message
---------------	-------------------

6.12.1.2 ~ErrorColor()

```
Errors::ErrorColor::~~ErrorColor ( ) [override], [default], [noexcept]
```

Destructor a new Error Color object.

6.12.1.3 ErrorColor() [2/3]

```
Errors::ErrorColor::ErrorColor (
    const ErrorColor & other ) [delete]
```

Construct a new Error Color object by copy.

Parameters

<i>other</i>	
--------------	--

6.12.1.4 ErrorColor() [3/3]

```
Errors::ErrorColor::ErrorColor (
    ErrorColor && other ) [default]
```

Construct a new [ErrorColor](#) object by move.

Parameters

<i>other</i>	
--------------	--

6.12.2 Member Function Documentation

6.12.2.1 operator=() [1/2]

```
ErrorColor& Errors::ErrorColor::operator= (
    const ErrorColor & other ) [delete]
```

Assign a [ErrorColor](#) object by copy.

Parameters

<i>other</i>	
--------------	--

Returns

[ErrorColor&](#)

6.12.2.2 operator=() [2/2]

```
ErrorColor& Errors::ErrorColor::operator= (
    ErrorColor && other ) [default]
```

Assign a [ErrorColor](#) object by move.

Parameters

<i>other</i>	
--------------	--

Returns

[ErrorColor&](#)

6.12.2.3 what()

```
const char * Errors::ErrorColor::what ( ) const [override], [noexcept]
```

Get the What object.

Returns

const char* The error message

6.12.3 Field Documentation

6.12.3.1 name

```
std::string Errors::ErrorColor::name [private]
```

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

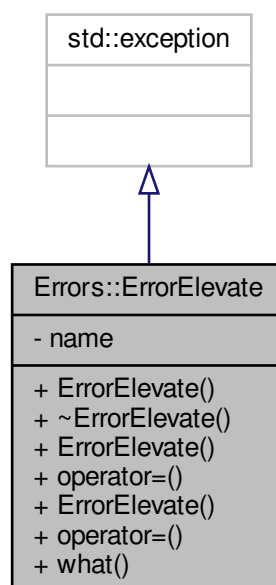
- [Errors/Errors.hpp](#)
- [Errors/Errors.cpp](#)

6.13 Errors::ErrorElevate Class Reference

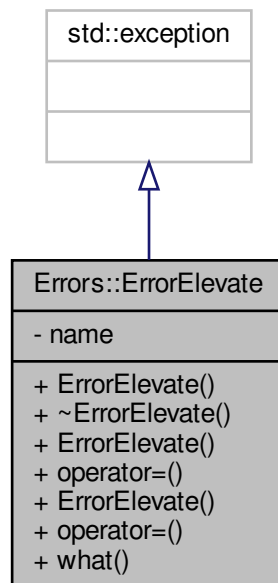
A class for Elevate Error.

```
#include <Errors.hpp>
```

Inheritance diagram for Errors::ErrorElevate:



Collaboration diagram for Errors::ErrorElevate:



Public Member Functions

- `ErrorElevate` (`std::string m_what`)
Constructor a new Error Elevate object.
- `~ErrorElevate` () noexcept override=default
Destructor a new Error Elevate object.
- `ErrorElevate` (const `ErrorElevate` &other)=delete
Construct a new Error Elevate object by copy.
- `ErrorElevate` & `operator=` (const `ErrorElevate` &other)=delete
Assign a `ErrorElevate` object by copy.
- `ErrorElevate` (`ErrorElevate` &&other)=default
Construct a new `ErrorElevate` object by move.
- `ErrorElevate` & `operator=` (`ErrorElevate` &&other)=default
Assign a `ErrorElevate` object by move.
- const char * `what` () const noexcept override
Get the What object.

Private Attributes

- `std::string name`

6.13.1 Detailed Description

A class for Elevate Error.

6.13.2 Constructor & Destructor Documentation

6.13.2.1 ErrorElevate() [1/3]

```
Errors::ErrorElevate::ErrorElevate (
    std::string m_what ) [explicit]
```

Constructor a new Error Elevate object.

Parameters

<i>m_what</i>	The error message
---------------	-------------------

6.13.2.2 ~ErrorElevate()

```
Errors::ErrorElevate::~~ErrorElevate ( ) [override], [default], [noexcept]
```

Destructor a new Error Elevate object.

6.13.2.3 ErrorElevate() [2/3]

```
Errors::ErrorElevate::ErrorElevate (
    const ErrorElevate & other ) [delete]
```

Construct a new Error Elevate object by copy.

Parameters

<i>other</i>	
--------------	--

6.13.2.4 ErrorElevate() [3/3]

```
Errors::ErrorElevate::ErrorElevate (
    ErrorElevate && other ) [default]
```

Construct a new [ErrorElevate](#) object by move.

Parameters

<i>other</i>	
--------------	--

6.13.3 Member Function Documentation

6.13.3.1 operator=() [1/2]

```
ErrorElevate& Errors::ErrorElevate::operator= (  
    const ErrorElevate & other ) [delete]
```

Assign a [ErrorElevate](#) object by copy.

Parameters

<i>other</i>	
--------------	--

Returns

[ErrorElevate&](#)

6.13.3.2 operator=() [2/2]

```
ErrorElevate& Errors::ErrorElevate::operator= (  
    ErrorElevate && other ) [default]
```

Assign a [ErrorElevate](#) object by move.

Parameters

<i>other</i>	
--------------	--

Returns

[ErrorElevate&](#)

6.13.3.3 what()

```
const char * Errors::ErrorElevate::what ( ) const [override], [noexcept]
```

Get the What object.

Returns

const char* The error message

6.13.4 Field Documentation

6.13.4.1 name

```
std::string Errors::ErrorElevate::name [private]
```

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

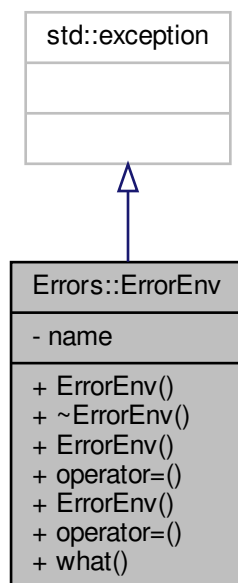
- [Errors/Errors.hpp](#)
- [Errors/Errors.cpp](#)

6.14 Errors::ErrorEnv Class Reference

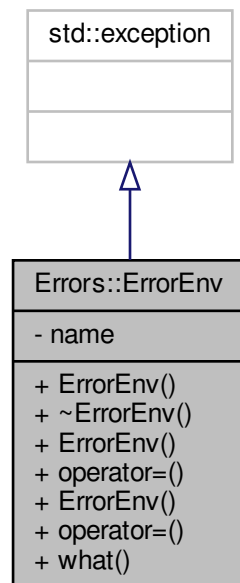
A class for env Error.

```
#include <Errors.hpp>
```

Inheritance diagram for Errors::ErrorEnv:



Collaboration diagram for Errors::ErrorEnv:



Public Member Functions

- `ErrorEnv` (`std::string m_what`)
Constructor a new Error Env object.
- `~ErrorEnv` () noexcept override=default
Destructor a new Error Env object.
- `ErrorEnv` (const `ErrorEnv` &other)=delete
Construct a new Error Env object by copy.
- `ErrorEnv` & `operator=` (const `ErrorEnv` &other)=delete
Assign a `ErrorEnv` object by copy.
- `ErrorEnv` (`ErrorEnv` &&other)=default
Construct a new `ErrorEnv` object by move.
- `ErrorEnv` & `operator=` (`ErrorEnv` &&other)=default
Assign a `ErrorEnv` object by move.
- const char * `what` () const noexcept override
Get the What object.

Private Attributes

- `std::string name`

6.14.1 Detailed Description

A class for env Error.

6.14.2 Constructor & Destructor Documentation

6.14.2.1 ErrorEnv() [1/3]

```
Errors::ErrorEnv::ErrorEnv (
    std::string m_what ) [explicit]
```

Constructor a new Error Env object.

Parameters

<i>m_what</i>	The error message
---------------	-------------------

6.14.2.2 ~ErrorEnv()

```
Errors::ErrorEnv::~~ErrorEnv ( ) [override], [default], [noexcept]
```

Destructor a new Error Env object.

6.14.2.3 ErrorEnv() [2/3]

```
Errors::ErrorEnv::ErrorEnv (
    const ErrorEnv & other ) [delete]
```

Construct a new Error Env object by copy.

Parameters

<i>other</i>	
--------------	--

6.14.2.4 ErrorEnv() [3/3]

```
Errors::ErrorEnv::ErrorEnv (
    ErrorEnv && other ) [default]
```

Construct a new [ErrorEnv](#) object by move.

Parameters

<i>other</i>	
--------------	--

6.14.3 Member Function Documentation

6.14.3.1 operator=() [1/2]

```
ErrorEnv& Errors::ErrorEnv::operator= (
    const ErrorEnv & other ) [delete]
```

Assign a [ErrorEnv](#) object by copy.

Parameters

<i>other</i>	
--------------	--

Returns

[ErrorEnv](#)&

6.14.3.2 operator=() [2/2]

```
ErrorEnv& Errors::ErrorEnv::operator= (
    ErrorEnv && other ) [default]
```

Assign a [ErrorEnv](#) object by move.

Parameters

<i>other</i>	
--------------	--

Returns

[ErrorEnv](#)&

6.14.3.3 what()

```
const char * Errors::ErrorEnv::what ( ) const [override], [noexcept]
```

Get the What object.

Returns

const char* The error message

6.14.4 Field Documentation

6.14.4.1 name

```
std::string Errors::ErrorEnv::name [private]
```

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

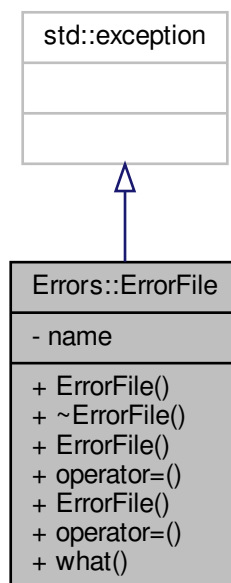
- Errors/[Errors.hpp](#)
- Errors/[Errors.cpp](#)

6.15 Errors::ErrorFile Class Reference

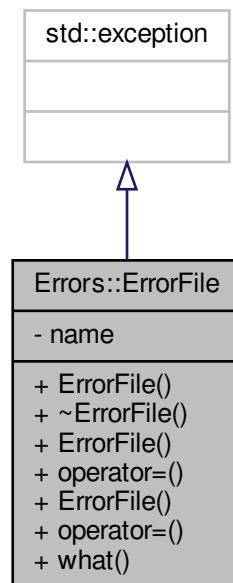
A class for file Error.

```
#include <Errors.hpp>
```

Inheritance diagram for Errors::ErrorFile:



Collaboration diagram for Errors::ErrorFile:



Public Member Functions

- `ErrorFile` (`std::string m_what`)
Constructor a new Error File object.
- `~ErrorFile` () noexcept override=default
Destructor a new Error File object.
- `ErrorFile` (const `ErrorFile` &other)=delete
Construct a new Error File object by copy.
- `ErrorFile` & `operator=` (const `ErrorFile` &other)=delete
Assign a `ErrorFile` object by copy.
- `ErrorFile` (`ErrorFile` &&other)=default
Construct a new `ErrorFile` object by move.
- `ErrorFile` & `operator=` (`ErrorFile` &&other)=default
Assign a `ErrorFile` object by move.
- const char * `what` () const noexcept override
Get the What object.

Private Attributes

- `std::string name`

6.15.1 Detailed Description

A class for file Error.

6.15.2 Constructor & Destructor Documentation

6.15.2.1 `ErrorFile()` [1/3]

```
Errors::ErrorFile::ErrorFile (
    std::string m_what ) [explicit]
```

Constructor a new Error File object.

Parameters

<i>m_what</i>	The error message
---------------	-------------------

6.15.2.2 `~ErrorFile()`

```
Errors::ErrorFile::~~ErrorFile ( ) [override], [default], [noexcept]
```

Destructor a new Error File object.

6.15.2.3 `ErrorFile()` [2/3]

```
Errors::ErrorFile::ErrorFile (
    const ErrorFile & other ) [delete]
```

Construct a new Error File object by copy.

Parameters

<i>other</i>	
--------------	--

6.15.2.4 `ErrorFile()` [3/3]

```
Errors::ErrorFile::ErrorFile (
    ErrorFile && other ) [default]
```

Construct a new [ErrorFile](#) object by move.

Parameters

<i>other</i>	
--------------	--

6.15.3 Member Function Documentation

6.15.3.1 operator=() [1/2]

```
ErrorFile& Errors::ErrorFile::operator= (
    const ErrorFile & other ) [delete]
```

Assign a [ErrorFile](#) object by copy.

Parameters

<i>other</i>	
--------------	--

Returns

[ErrorFile&](#)

6.15.3.2 operator=() [2/2]

```
ErrorFile& Errors::ErrorFile::operator= (
    ErrorFile && other ) [default]
```

Assign a [ErrorFile](#) object by move.

Parameters

<i>other</i>	
--------------	--

Returns

[ErrorFile&](#)

6.15.3.3 what()

```
const char * Errors::ErrorFile::what ( ) const [override], [noexcept]
```

Get the What object.

Returns

const char* The error message

6.15.4 Field Documentation

6.15.4.1 name

```
std::string Errors::ErrorFile::name [private]
```

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

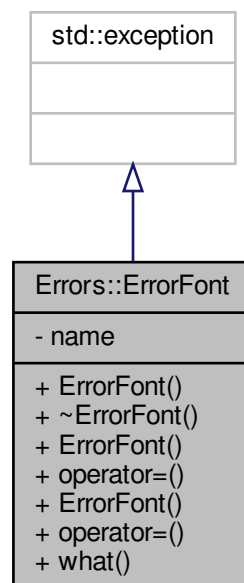
- [Errors/Errors.hpp](#)
- [Errors/Errors.cpp](#)

6.16 Errors::ErrorFont Class Reference

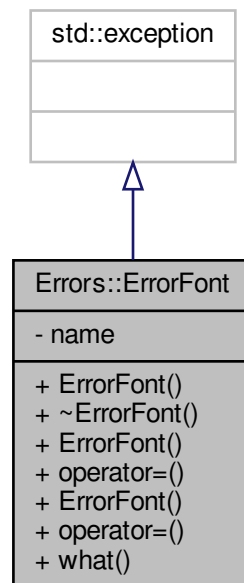
A class for font Error.

```
#include <Errors.hpp>
```

Inheritance diagram for Errors::ErrorFont:



Collaboration diagram for Errors::ErrorFont:



Public Member Functions

- `ErrorFont (std::string m_what)`
Constructor a new Error Font object.
- `~ErrorFont ()` noexcept override=default
Destructor a new Error Font object.
- `ErrorFont (const ErrorFont &other)=delete`
Construct a new Error Font object by copy.
- `ErrorFont & operator= (const ErrorFont &other)=delete`
Assign a ErrorFont object by copy.
- `ErrorFont (ErrorFont &&other)=default`
Construct a new ErrorFont object by move.
- `ErrorFont & operator= (ErrorFont &&other)=default`
Assign a ErrorFont object by move.
- `const char * what ()` const noexcept override
Get the What object.

Private Attributes

- `std::string name`

6.16.1 Detailed Description

A class for font Error.

6.16.2 Constructor & Destructor Documentation

6.16.2.1 ErrorFont() [1/3]

```
Errors::ErrorFont::ErrorFont (
    std::string m_what ) [explicit]
```

Constructor a new Error Font object.

Parameters

<i>m_what</i>	The error message
---------------	-------------------

6.16.2.2 ~ErrorFont()

```
Errors::ErrorFont::~~ErrorFont ( ) [override], [default], [noexcept]
```

Destructor a new Error Font object.

6.16.2.3 ErrorFont() [2/3]

```
Errors::ErrorFont::ErrorFont (
    const ErrorFont & other ) [delete]
```

Construct a new Error Font object by copy.

Parameters

<i>other</i>	
--------------	--

6.16.2.4 ErrorFont() [3/3]

```
Errors::ErrorFont::ErrorFont (
    ErrorFont && other ) [default]
```

Construct a new [ErrorFont](#) object by move.

Parameters

<i>other</i>	
--------------	--

6.16.3 Member Function Documentation

6.16.3.1 operator=() [1/2]

```
ErrorFont& Errors::ErrorFont::operator= (
    const ErrorFont & other ) [delete]
```

Assign a [ErrorFont](#) object by copy.

Parameters

<i>other</i>	
--------------	--

Returns

[ErrorFont&](#)

6.16.3.2 operator=() [2/2]

```
ErrorFont& Errors::ErrorFont::operator= (
    ErrorFont && other ) [default]
```

Assign a [ErrorFont](#) object by move.

Parameters

<i>other</i>	
--------------	--

Returns

[ErrorFont&](#)

6.16.3.3 what()

```
const char * Errors::ErrorFont::what ( ) const [override], [noexcept]
```

Get the What object.

Returns

const char* The error message

6.16.4 Field Documentation

6.16.4.1 name

```
std::string Errors::ErrorFont::name [private]
```

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

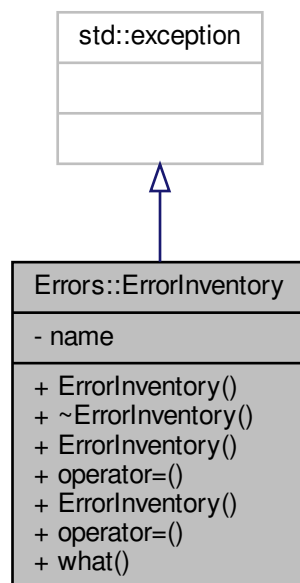
- [Errors/Errors.hpp](#)
- [Errors/Errors.cpp](#)

6.17 Errors::ErrorInventory Class Reference

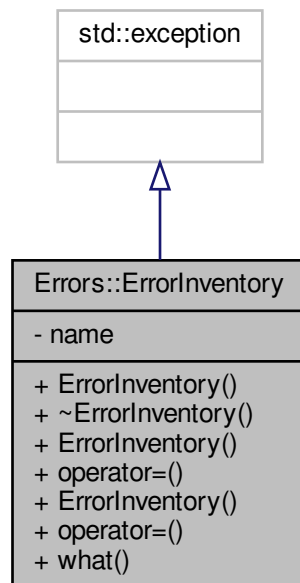
A class for Inventory Error.

```
#include <Errors.hpp>
```

Inheritance diagram for Errors::ErrorInventory:



Collaboration diagram for Errors::ErrorInventory:



Public Member Functions

- `ErrorInventory` (`std::string m_what`)
Constructor a new Error Inventory object.
- `~ErrorInventory` () noexcept override=default
Destructor a new Error Inventory object.
- `ErrorInventory` (const `ErrorInventory` &other)=delete
Construct a new Error Inventory object by copy.
- `ErrorInventory & operator=` (const `ErrorInventory` &other)=delete
Assign a `ErrorInventory` object by copy.
- `ErrorInventory` (`ErrorInventory` &&other)=default
Construct a new `ErrorInventory` object by move.
- `ErrorInventory & operator=` (`ErrorInventory` &&other)=default
Assign a `ErrorInventory` object by move.
- const char * `what` () const noexcept override
Get the What object.

Private Attributes

- std::string `name`

6.17.1 Detailed Description

A class for Inventory Error.

6.17.2 Constructor & Destructor Documentation

6.17.2.1 `ErrorInventory()` [1/3]

```
Errors::ErrorInventory::ErrorInventory (
    std::string m_what ) [explicit]
```

Constructor a new Error Inventory object.

Parameters

<i>m_what</i>	The error message
---------------	-------------------

6.17.2.2 `~ErrorInventory()`

```
Errors::ErrorInventory::~~ErrorInventory ( ) [override], [default], [noexcept]
```

Destructor a new Error Inventory object.

6.17.2.3 `ErrorInventory()` [2/3]

```
Errors::ErrorInventory::ErrorInventory (
    const ErrorInventory & other ) [delete]
```

Construct a new Error Inventory object by copy.

Parameters

<i>other</i>	
--------------	--

6.17.2.4 `ErrorInventory()` [3/3]

```
Errors::ErrorInventory::ErrorInventory (
    ErrorInventory && other ) [default]
```

Construct a new [ErrorInventory](#) object by move.

Parameters

<i>other</i>	
--------------	--

6.17.3 Member Function Documentation

6.17.3.1 operator=() [1/2]

```
ErrorInventory& Errors::ErrorInventory::operator= (  
    const ErrorInventory & other ) [delete]
```

Assign a [ErrorInventory](#) object by copy.

Parameters

<i>other</i>	
--------------	--

Returns

[ErrorInventory&](#)

6.17.3.2 operator=() [2/2]

```
ErrorInventory& Errors::ErrorInventory::operator= (  
    ErrorInventory && other ) [default]
```

Assign a [ErrorInventory](#) object by move.

Parameters

<i>other</i>	
--------------	--

Returns

[ErrorInventory&](#)

6.17.3.3 what()

```
const char * Errors::ErrorInventory::what ( ) const [override], [noexcept]
```

Get the What object.

Returns

const char* The error message

6.17.4 Field Documentation

6.17.4.1 name

```
std::string Errors::ErrorInventory::name [private]
```

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

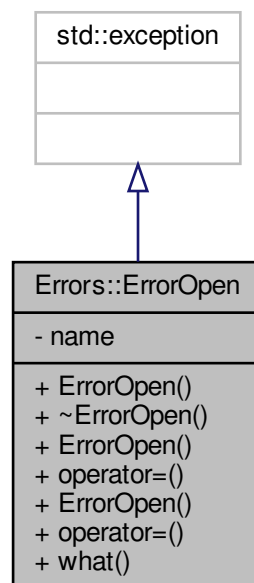
- [Errors/Errors.hpp](#)
- [Errors/Errors.cpp](#)

6.18 Errors::ErrorOpen Class Reference

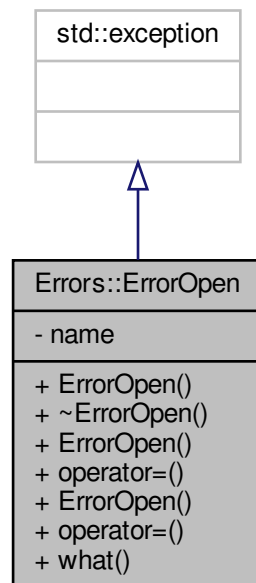
A class for open Error.

```
#include <Errors.hpp>
```

Inheritance diagram for Errors::ErrorOpen:



Collaboration diagram for Errors::ErrorOpen:



Public Member Functions

- `ErrorOpen` (`std::string m_what`)
Constructor a new Error Open object.
- `~ErrorOpen` () noexcept override=default
Destructor a new Error Open object.
- `ErrorOpen` (`const ErrorOpen &other`)=delete
Construct a new Error Open object by copy.
- `ErrorOpen & operator=` (`const ErrorOpen &other`)=delete
Assign a ErrorOpen object by copy.
- `ErrorOpen` (`ErrorOpen &&other`)=default
Construct a new ErrorOpen object by move.
- `ErrorOpen & operator=` (`ErrorOpen &&other`)=default
Assign a ErrorOpen object by move.
- `const char * what` () const noexcept override
Get the What object.

Private Attributes

- `std::string name`

6.18.1 Detailed Description

A class for open Error.

6.18.2 Constructor & Destructor Documentation

6.18.2.1 ErrorOpen() [1/3]

```
Errors::ErrorOpen::ErrorOpen (
    std::string m_what ) [explicit]
```

Constructor a new Error Open object.

Parameters

<i>m_what</i>	The error message
---------------	-------------------

6.18.2.2 ~ErrorOpen()

```
Errors::ErrorOpen::~~ErrorOpen ( ) [override], [default], [noexcept]
```

Destructor a new Error Open object.

6.18.2.3 ErrorOpen() [2/3]

```
Errors::ErrorOpen::ErrorOpen (
    const ErrorOpen & other ) [delete]
```

Construct a new Error Open object by copy.

Parameters

<i>other</i>	
--------------	--

6.18.2.4 ErrorOpen() [3/3]

```
Errors::ErrorOpen::ErrorOpen (
    ErrorOpen && other ) [default]
```

Construct a new [ErrorOpen](#) object by move.

Parameters

<i>other</i>	
--------------	--

6.18.3 Member Function Documentation

6.18.3.1 operator=() [1/2]

```
ErrorOpen& Errors::ErrorOpen::operator= (
    const ErrorOpen & other ) [delete]
```

Assign a [ErrorOpen](#) object by copy.

Parameters

<i>other</i>	
--------------	--

Returns

[ErrorOpen](#)&

6.18.3.2 operator=() [2/2]

```
ErrorOpen& Errors::ErrorOpen::operator= (
    ErrorOpen && other ) [default]
```

Assign a [ErrorOpen](#) object by move.

Parameters

<i>other</i>	
--------------	--

Returns

[ErrorOpen](#)&

6.18.3.3 what()

```
const char * Errors::ErrorOpen::what ( ) const [override], [noexcept]
```

Get the What object.

Returns

const char* The error message

6.18.4 Field Documentation

6.18.4.1 name

```
std::string Errors::ErrorOpen::name [private]
```

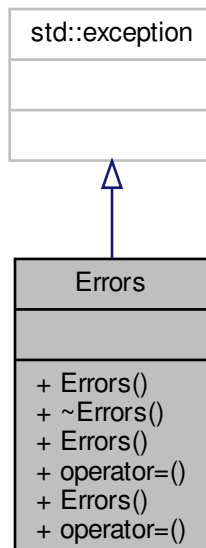
The documentation for this class was generated from the following files:

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

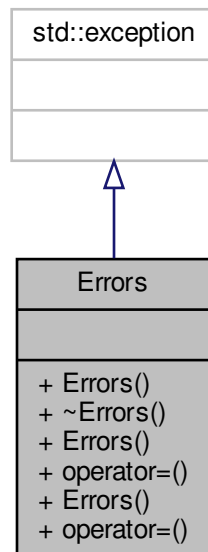
6.19 Errors Class Reference

```
#include <Errors.hpp>
```

Inheritance diagram for Errors:



Collaboration diagram for Errors:



Data Structures

- class [ErrorBroadcast](#)
A class for broadcast Error.
- class [ErrorButton](#)
A class for Button Error.
- class [ErrorCharacter](#)
A class for Character Error.
- class [ErrorColor](#)
- class [ErrorElevate](#)
A class for Elevate Error.
- class [ErrorEnv](#)
A class for env Error.
- class [ErrorFile](#)
A class for file Error.
- class [ErrorFont](#)
A class for font Error.
- class [ErrorInventory](#)
A class for Inventory Error.
- class [ErrorOpen](#)
A class for open Error.
- class [ErrorSocket](#)
A class for socket Error.
- class [ErrorTeam](#)
A class for Team Error.
- class [ErrorTexture](#)

A class for Texture Error.

- class [ErrorUsage](#)

A class for Usage Error.

Public Member Functions

- [Errors](#) ()=default
Default Constructor a new Error T object.
- [~Errors](#) ()=default
Default Destructor a new Error T object.
- [Errors](#) (const [Errors](#) &other)=delete
Construct a new [Errors](#) object by copy.
- [Errors](#) & operator= (const [Errors](#) &other)=delete
Assign a [Errors](#) object by copy.
- [Errors](#) ([Errors](#) &&other)=default
Construct a new [Errors](#) object by move.
- [Errors](#) & operator= ([Errors](#) &&other)=default
Assign a [Errors](#) object by move.

6.19.1 Constructor & Destructor Documentation

6.19.1.1 Errors() [1/3]

```
Errors::Errors ( ) [default]
```

Default Constructor a new Error T object.

6.19.1.2 ~Errors()

```
Errors::~~Errors ( ) [default]
```

Default Destructor a new Error T object.

6.19.1.3 Errors() [2/3]

```
Errors::Errors (
    const Errors & other ) [delete]
```

Construct a new [Errors](#) object by copy.

Parameters

<i>other</i>	
--------------	--

6.19.1.4 Errors() [3/3]

```
Errors::Errors (
    Errors && other ) [default]
```

Construct a new [Errors](#) object by move.

Parameters

<i>other</i>	
--------------	--

6.19.2 Member Function Documentation**6.19.2.1 operator=()** [1/2]

```
Errors& Errors::operator= (
    const Errors & other ) [delete]
```

Assign a [Errors](#) object by copy.

6.19.2.2 operator=() [2/2]

```
Errors& Errors::operator= (
    Errors && other ) [default]
```

Assign a [Errors](#) object by move.

Parameters

<i>other</i>	
--------------	--

Returns

[Errors&](#)

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

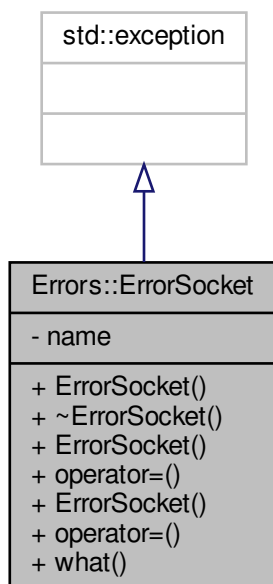
- Errors/[Errors.hpp](#)

6.20 Errors::ErrorSocket Class Reference

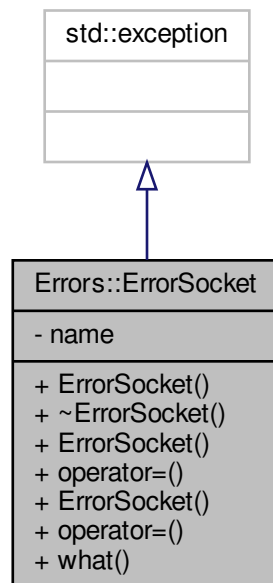
A class for socket Error.

```
#include <Errors.hpp>
```

Inheritance diagram for Errors::ErrorSocket:



Collaboration diagram for Errors::ErrorSocket:



Public Member Functions

- `ErrorSocket` (`std::string m_what`)
Constructor a new Error Socket object.
- `~ErrorSocket` () noexcept override=default
Destructor a new Error Socket object.
- `ErrorSocket` (`const ErrorSocket &other`)=delete
Construct a new Error Socket object by copy.
- `ErrorSocket & operator=` (`const ErrorSocket &other`)=delete
Assign a `ErrorSocket` object by copy.
- `ErrorSocket` (`ErrorSocket &&other`)=default
Construct a new `ErrorSocket` object by move.
- `ErrorSocket & operator=` (`ErrorSocket &&other`)=default
Assign a `ErrorSocket` object by move.
- `const char * what` () const noexcept override
Get the What object.

Private Attributes

- `std::string name`

6.20.1 Detailed Description

A class for socket Error.

6.20.2 Constructor & Destructor Documentation

6.20.2.1 ErrorSocket() [1/3]

```
Errors::ErrorSocket::ErrorSocket (
    std::string m_what ) [explicit]
```

Constructor a new Error Socket object.

Parameters

<i>m_what</i>	The error message
---------------	-------------------

6.20.2.2 ~ErrorSocket()

```
Errors::ErrorSocket::~~ErrorSocket ( ) [override], [default], [noexcept]
```

Destructor a new Error Socket object.

6.20.2.3 ErrorSocket() [2/3]

```
Errors::ErrorSocket::ErrorSocket (
    const ErrorSocket & other ) [delete]
```

Construct a new Error Socket object by copy.

Parameters

<i>other</i>	
--------------	--

6.20.2.4 ErrorSocket() [3/3]

```
Errors::ErrorSocket::ErrorSocket (
    ErrorSocket && other ) [default]
```

Construct a new [ErrorSocket](#) object by move.

Parameters

<i>other</i>	
--------------	--

6.20.3 Member Function Documentation

6.20.3.1 operator=() [1/2]

```
ErrorSocket& Errors::ErrorSocket::operator= (
    const ErrorSocket & other ) [delete]
```

Assign a [ErrorSocket](#) object by copy.

Parameters

<i>other</i>	
--------------	--

Returns

[ErrorSocket&](#)

6.20.3.2 operator=() [2/2]

```
ErrorSocket& Errors::ErrorSocket::operator= (
    ErrorSocket && other ) [default]
```

Assign a [ErrorSocket](#) object by move.

Parameters

<i>other</i>	
--------------	--

Returns

[ErrorSocket&](#)

6.20.3.3 what()

```
const char * Errors::ErrorSocket::what ( ) const [override], [noexcept]
```

Get the What object.

Returns

const char* The error message

6.20.4 Field Documentation

6.20.4.1 name

```
std::string Errors::ErrorSocket::name [private]
```

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

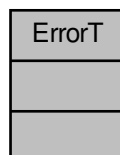
- [Errors/Errors.hpp](#)
- [Errors/Errors.cpp](#)

6.21 ErrorT Class Reference

A class to handle errors. This class provides a way to throw errors.

```
#include <Errors.hpp>
```

Collaboration diagram for ErrorT:



6.21.1 Detailed Description

A class to handle errors. This class provides a way to throw errors.

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

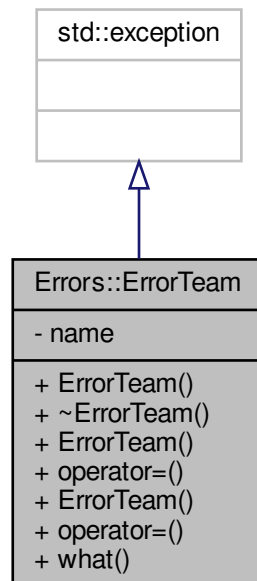
- [Errors/Errors.hpp](#)

6.22 Errors::ErrorTeam Class Reference

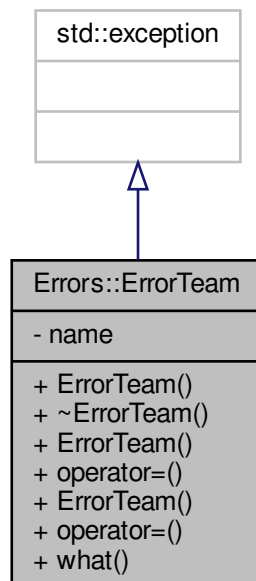
A class for Team Error.

```
#include <Errors.hpp>
```

Inheritance diagram for Errors::ErrorTeam:



Collaboration diagram for Errors::ErrorTeam:



Public Member Functions

- `ErrorTeam` (`std::string m_what`)
Constructor a new Error Team object.
- `~ErrorTeam` () noexcept override=default
Destructor a new Error Team object.
- `ErrorTeam` (const `ErrorTeam` &other)=delete
Construct a new Error Team object by copy.
- `ErrorTeam` & `operator=` (const `ErrorTeam` &other)=delete
Assign a `ErrorTeam` object by copy.
- `ErrorTeam` (`ErrorTeam` &&other)=default
Construct a new `ErrorTeam` object by move.
- `ErrorTeam` & `operator=` (`ErrorTeam` &&other)=default
Assign a `ErrorTeam` object by move.
- const char * `what` () const noexcept override
Get the What object.

Private Attributes

- `std::string name`

6.22.1 Detailed Description

A class for Team Error.

6.22.2 Constructor & Destructor Documentation

6.22.2.1 ErrorTeam() [1/3]

```
Errors::ErrorTeam::ErrorTeam (
    std::string m_what ) [explicit]
```

Constructor a new Error Team object.

Parameters

<i>m_what</i>	The error message
---------------	-------------------

6.22.2.2 ~ErrorTeam()

```
Errors::ErrorTeam::~~ErrorTeam ( ) [override], [default], [noexcept]
```

Destructor a new Error Team object.

6.22.2.3 ErrorTeam() [2/3]

```
Errors::ErrorTeam::ErrorTeam (
    const ErrorTeam & other ) [delete]
```

Construct a new Error Team object by copy.

Parameters

<i>other</i>	
--------------	--

6.22.2.4 ErrorTeam() [3/3]

```
Errors::ErrorTeam::ErrorTeam (
    ErrorTeam && other ) [default]
```

Construct a new [ErrorTeam](#) object by move.

Parameters

<i>other</i>	
--------------	--

6.22.3 Member Function Documentation

6.22.3.1 operator=() [1/2]

```
ErrorTeam& Errors::ErrorTeam::operator= (
    const ErrorTeam & other ) [delete]
```

Assign a [ErrorTeam](#) object by copy.

Parameters

<i>other</i>	
--------------	--

Returns

[ErrorTeam&](#)

6.22.3.2 operator=() [2/2]

```
ErrorTeam& Errors::ErrorTeam::operator= (
    ErrorTeam && other ) [default]
```

Assign a [ErrorTeam](#) object by move.

Parameters

<i>other</i>	
--------------	--

Returns

[ErrorTeam&](#)

6.22.3.3 what()

```
const char * Errors::ErrorTeam::what ( ) const [override], [noexcept]
```

Get the What object.

Returns

const char* The error message

6.22.4 Field Documentation

6.22.4.1 name

```
std::string Errors::ErrorTeam::name [private]
```

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

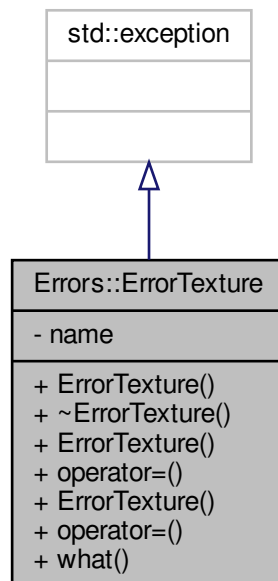
- Errors/[Errors.hpp](#)
- Errors/[Errors.cpp](#)

6.23 Errors::ErrorTexture Class Reference

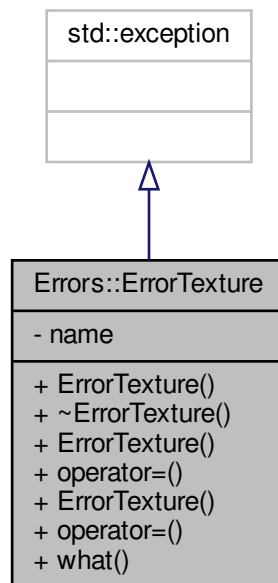
A class for Texture Error.

```
#include <Errors.hpp>
```

Inheritance diagram for Errors::ErrorTexture:



Collaboration diagram for Errors::ErrorTexture:



Public Member Functions

- `ErrorTexture` (`std::string m_what`)
Constructor a new Error Texture object.
- `~ErrorTexture` () noexcept override=default
Destructor a new Error Texture object.
- `ErrorTexture` (const `ErrorTexture` &other)=delete
Construct a new Error Texture object by copy.
- `ErrorTexture` & `operator=` (const `ErrorTexture` &other)=delete
Assign a ErrorTexture object by copy.
- `ErrorTexture` (`ErrorTexture` &&other)=default
Construct a new ErrorTexture object by move.
- `ErrorTexture` & `operator=` (`ErrorTexture` &&other)=default
Assign a ErrorTexture object by move.
- const char * `what` () const noexcept override
Get the What object.

Private Attributes

- `std::string name`

6.23.1 Detailed Description

A class for Texture Error.

6.23.2 Constructor & Destructor Documentation

6.23.2.1 ErrorTexture() [1/3]

```
Errors::ErrorTexture::ErrorTexture (
    std::string m_what ) [explicit]
```

Constructor a new Error Texture object.

Parameters

<i>m_what</i>	The error message
---------------	-------------------

6.23.2.2 ~ErrorTexture()

```
Errors::ErrorTexture::~~ErrorTexture ( ) [override], [default], [noexcept]
```

Destructor a new Error Texture object.

6.23.2.3 ErrorTexture() [2/3]

```
Errors::ErrorTexture::ErrorTexture (
    const ErrorTexture & other ) [delete]
```

Construct a new Error Texture object by copy.

Parameters

<i>other</i>	
--------------	--

6.23.2.4 ErrorTexture() [3/3]

```
Errors::ErrorTexture::ErrorTexture (
    ErrorTexture && other ) [default]
```

Construct a new [ErrorTexture](#) object by move.

Parameters

<i>other</i>	
--------------	--

6.23.3 Member Function Documentation

6.23.3.1 operator=() [1/2]

```
ErrorTexture& Errors::ErrorTexture::operator= (  
    const ErrorTexture & other ) [delete]
```

Assign a [ErrorTexture](#) object by copy.

Parameters

<i>other</i>	
--------------	--

Returns

[ErrorTexture&](#)

6.23.3.2 operator=() [2/2]

```
ErrorTexture& Errors::ErrorTexture::operator= (  
    ErrorTexture && other ) [default]
```

Assign a [ErrorTexture](#) object by move.

Parameters

<i>other</i>	
--------------	--

Returns

[ErrorTexture&](#)

6.23.3.3 what()

```
const char * Errors::ErrorTexture::what ( ) const [override], [noexcept]
```

Get the What object.

Returns

const char* The error message

6.23.4 Field Documentation

6.23.4.1 name

```
std::string Errors::ErrorTexture::name [private]
```

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

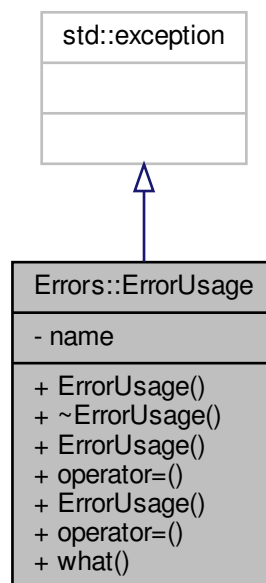
- Errors/[Errors.hpp](#)
- Errors/[Errors.cpp](#)

6.24 Errors::ErrorUsage Class Reference

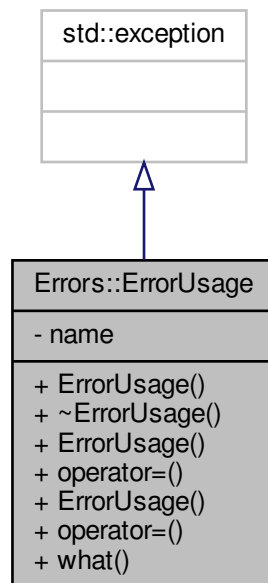
A class for Usage Error.

```
#include <Errors.hpp>
```

Inheritance diagram for Errors::ErrorUsage:



Collaboration diagram for Errors::ErrorUsage:



Public Member Functions

- [ErrorUsage](#) (std::string m_what)
Constructor a new Error Usage object.
- [~ErrorUsage](#) () noexcept override=default
Destructor a new Error Usage object.
- [ErrorUsage](#) (const [ErrorUsage](#) &other)=delete
Construct a new Error Usage object by copy.
- [ErrorUsage](#) & [operator=](#) (const [ErrorUsage](#) &other)=delete
Assign a [ErrorUsage](#) object by copy.
- [ErrorUsage](#) ([ErrorUsage](#) &&other)=default
Construct a new [ErrorUsage](#) object by move.
- [ErrorUsage](#) & [operator=](#) ([ErrorUsage](#) &&other)=default
Assign a [ErrorUsage](#) object by move.
- const char * [what](#) () const noexcept override
Get the What object.

Private Attributes

- std::string [name](#)

6.24.1 Detailed Description

A class for Usage Error.

6.24.2 Constructor & Destructor Documentation

6.24.2.1 ErrorUsage() [1/3]

```
Errors::ErrorUsage::ErrorUsage (
    std::string m_what ) [explicit]
```

Constructor a new Error Usage object.

Parameters

<i>m_what</i>	The error message
---------------	-------------------

6.24.2.2 ~ErrorUsage()

```
Errors::ErrorUsage::~~ErrorUsage ( ) [override], [default], [noexcept]
```

Destructor a new Error Usage object.

6.24.2.3 ErrorUsage() [2/3]

```
Errors::ErrorUsage::ErrorUsage (
    const ErrorUsage & other ) [delete]
```

Construct a new Error Usage object by copy.

Parameters

<i>other</i>	
--------------	--

6.24.2.4 ErrorUsage() [3/3]

```
Errors::ErrorUsage::ErrorUsage (
    ErrorUsage && other ) [default]
```

Construct a new [ErrorUsage](#) object by move.

Parameters

<i>other</i>	
--------------	--

6.24.3 Member Function Documentation

6.24.3.1 operator=() [1/2]

```
ErrorUsage& Errors::ErrorUsage::operator= (
    const ErrorUsage & other ) [delete]
```

Assign a [ErrorUsage](#) object by copy.

Parameters

<i>other</i>	
--------------	--

Returns

[ErrorUsage&](#)

6.24.3.2 operator=() [2/2]

```
ErrorUsage& Errors::ErrorUsage::operator= (
    ErrorUsage && other ) [default]
```

Assign a [ErrorUsage](#) object by move.

Parameters

<i>other</i>	
--------------	--

Returns

[ErrorUsage&](#)

6.24.3.3 what()

```
const char * Errors::ErrorUsage::what ( ) const [override], [noexcept]
```

Get the What object.

Returns

const char* The error message

6.24.4 Field Documentation

6.24.4.1 name

```
std::string Errors::ErrorUsage::name [private]
```

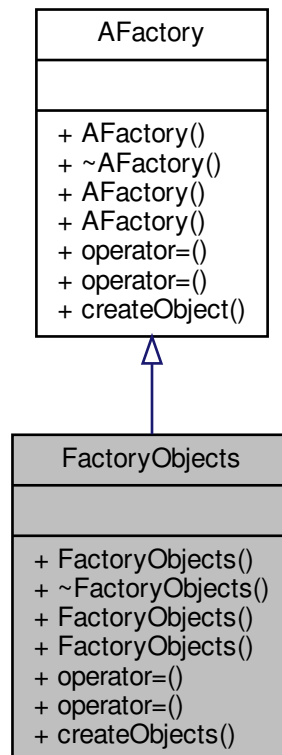
The documentation for this class was generated from the following files:

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

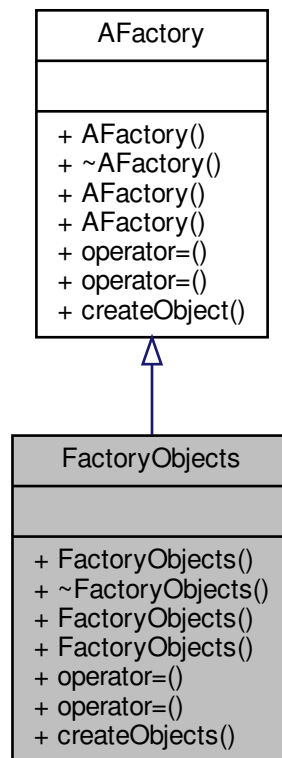
6.25 FactoryObjects Class Reference

```
#include <FactoryObjects.hpp>
```

Inheritance diagram for FactoryObjects:



Collaboration diagram for FactoryObjects:



Public Member Functions

- `FactoryObjects ()`=default
- `~FactoryObjects ()` noexcept=default
- `FactoryObjects (const FactoryObjects &other)`=delete
- `FactoryObjects (FactoryObjects &&other)`=default
- `FactoryObjects & operator= (const FactoryObjects &other)`=delete
- `FactoryObjects & operator= (FactoryObjects &&other)`=default
- `template<typename T, typename... Args>`
`std::unique_ptr< T > createObjects (Args &&... args)`

6.25.1 Constructor & Destructor Documentation

6.25.1.1 FactoryObjects() [1/3]

```
FactoryObjects::FactoryObjects ( ) [default]
```

6.25.1.2 ~FactoryObjects()

```
FactoryObjects::~~FactoryObjects ( ) [default], [noexcept]
```

6.25.1.3 FactoryObjects() [2/3]

```
FactoryObjects::FactoryObjects (
    const FactoryObjects & other ) [delete]
```

6.25.1.4 FactoryObjects() [3/3]

```
FactoryObjects::FactoryObjects (
    FactoryObjects && other ) [default]
```

6.25.2 Member Function Documentation

6.25.2.1 createObjects()

```
template<typename T , typename... Args>
std::unique_ptr<T> FactoryObjects::createObjects (
    Args &&... args ) [inline]
```

6.25.2.2 operator=() [1/2]

```
FactoryObjects& FactoryObjects::operator= (
    const FactoryObjects & other ) [delete]
```

6.25.2.3 operator=() [2/2]

```
FactoryObjects& FactoryObjects::operator= (
    FactoryObjects && other ) [default]
```

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

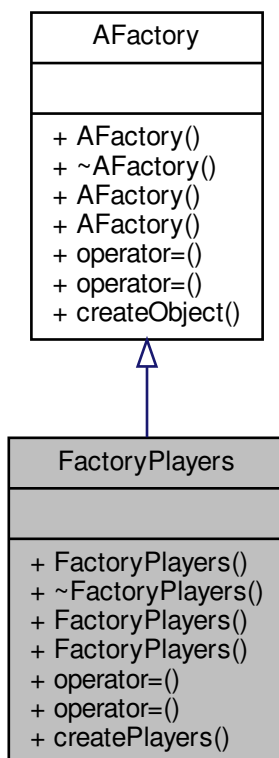
- Factory/FactoryObjects/[FactoryObjects.hpp](#)

6.26 FactoryPlayers Class Reference

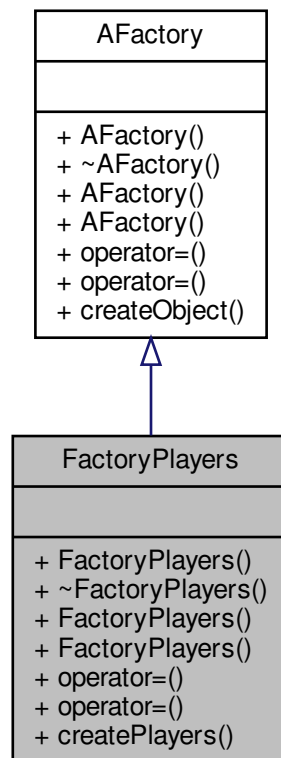
[FactoryPlayers](#) class.

```
#include <FactoryPlayers.hpp>
```

Inheritance diagram for FactoryPlayers:



Collaboration diagram for FactoryPlayers:



Public Member Functions

- `FactoryPlayers ()=default`
Construct a new Factory *Players* object.
- `~FactoryPlayers () noexcept=default`
Destroy the Factory *Players* object.
- `FactoryPlayers (const FactoryPlayers &other)=delete`
Construct a new Factory *Players* object by copy = delete.
- `FactoryPlayers (FactoryPlayers &&other)=default`
Construct a new Factory *Players* object by move.
- `FactoryPlayers & operator= (const FactoryPlayers &other)=delete`
Assign a Factory *Players* object by copy = delete.
- `FactoryPlayers & operator= (FactoryPlayers &&other)=default`
Assign a Factory *Players* object by move.
- `template<typename T, typename... Args>`
`std::unique_ptr< T > createPlayers (Args &&... args)`
 Create a Characters object.

6.26.1 Detailed Description

`FactoryPlayers` class.

6.26.2 Constructor & Destructor Documentation

6.26.2.1 FactoryPlayers() [1/3]

```
FactoryPlayers::FactoryPlayers ( ) [default]
```

Construct a new Factory [Players](#) object.

6.26.2.2 ~FactoryPlayers()

```
FactoryPlayers::~~FactoryPlayers ( ) [default], [noexcept]
```

Destroy the Factory [Players](#) object.

6.26.2.3 FactoryPlayers() [2/3]

```
FactoryPlayers::FactoryPlayers (
    const FactoryPlayers & other ) [delete]
```

Construct a new Factory [Players](#) object by copy = delete.

Parameters

<i>other</i>	
--------------	--

6.26.2.4 FactoryPlayers() [3/3]

```
FactoryPlayers::FactoryPlayers (
    FactoryPlayers && other ) [default]
```

Construct a new Factory [Players](#) object by move.

Parameters

<i>other</i>	
--------------	--

6.26.3 Member Function Documentation

6.26.3.1 createPlayers()

```
template<typename T , typename... Args>
std::unique_ptr<T> FactoryPlayers::createPlayers (
    Args &&... args ) [inline]
```

Create a Characters object.

Template Parameters

<i>T</i>	
<i>Args</i>	

Parameters

<i>args</i>	
-------------	--

Returns

std::unique_ptr<T>

6.26.3.2 operator=() [1/2]

```
FactoryPlayers& FactoryPlayers::operator= (
    const FactoryPlayers & other ) [delete]
```

Assign a Factory [Players](#) object by copy = delete.

Parameters

<i>other</i>	
--------------	--

Returns

[FactoryPlayers&](#)

6.26.3.3 operator=() [2/2]

```
FactoryPlayers& FactoryPlayers::operator= (
    FactoryPlayers && other ) [default]
```

Assign a Factory [Players](#) object by move.

Parameters

<i>other</i>	
--------------	--

Returns

[FactoryPlayers](#)&

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

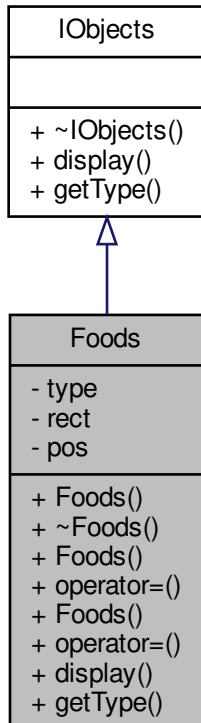
- [Factory/FactoryPlayers/FactoryPlayers.hpp](#)

6.27 Foods Class Reference

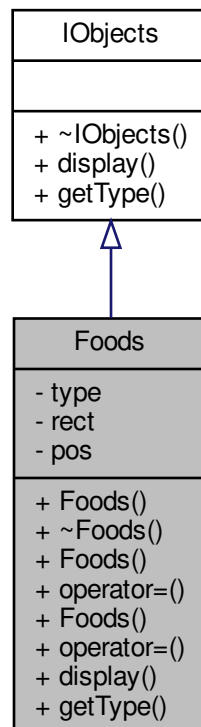
[Foods](#) class.

```
#include <Foods.hpp>
```

Inheritance diagram for Foods:



Collaboration diagram for Foods:



Public Member Functions

- **Foods** (sf::Vector2f **pos**, **TypeObject** typeObject)
Construct a new **Foods** object.
- **~Foods** () noexcept override=default
Destroy the **Foods** object.
- **Foods** (const **Foods** &stone)=default
Construct a new **Foods** object by copy.
- **Foods** & **operator=** (const **Foods** &stone)=default
Assign a **Foods** object by copy.
- **Foods** (**Foods** &&stone)=default
Construct a new **Foods** object by move.
- **Foods** & **operator=** (**Foods** &&stone)=default
Assign a **Foods** object by move.
- void **display** (sf::RenderWindow &>window, sf::Sprite &sprite, float size) override
Display the object.
- const **TypeObject** & **getType** () override
Get The type of the object.

Private Attributes

- [TypeObject](#) `type`
Get The Sprite of the object.
- `sf::IntRect` [rect](#)
- `sf::Vector2f` [pos](#)

6.27.1 Detailed Description

[Foods](#) class.

6.27.2 Constructor & Destructor Documentation

6.27.2.1 Foods() [1/3]

```
Foods::Foods (
    sf::Vector2f pos,
    TypeObject typeObject )
```

Construct a new [Foods](#) object.

Parameters

<i>pos</i>	
<i>typeObject</i>	

6.27.2.2 ~Foods()

```
Foods::~~Foods ( ) [override], [default], [noexcept]
```

Destroy the [Foods](#) object.

6.27.2.3 Foods() [2/3]

```
Foods::Foods (
    const Foods & stone ) [default]
```

Construct a new [Foods](#) object by copy.

Parameters

<i>stone</i>	
--------------	--

6.27.2.4 Foods() [3/3]

```
Foods::Foods (
    Foods && stone ) [default]
```

Construct a new [Foods](#) object by move.

Parameters

<i>stone</i>	
--------------	--

6.27.3 Member Function Documentation**6.27.3.1 display()**

```
void Foods::display (
    sf::RenderWindow & window,
    sf::Sprite & sprite,
    float size ) [override], [virtual]
```

Display the object.

Parameters

<i>window</i>	The window where the object will be displayed
<i>sprite</i>	The sprite of the object
<i>size</i>	The size of the object

Implements [IObjects](#).

6.27.3.2 getType()

```
const TypeObject & Foods::getType ( ) [override], [virtual]
```

Get The type of the object.

Returns

The type of the object

Implements [IObjects](#).

6.27.3.3 operator=() [1/2]

```
Foods& Foods::operator= (
    const Foods & stone ) [default]
```

Assign a [Foods](#) object by copy.

Parameters

<i>stone</i>	
--------------	--

Returns

[Foods&](#)

6.27.3.4 operator=() [2/2]

```
Foods& Foods::operator= (
    Foods && stone ) [default]
```

Assign a [Foods](#) object by move.

Parameters

<i>stone</i>	
--------------	--

Returns

[Foods&](#)

6.27.4 Field Documentation

6.27.4.1 pos

```
sf::Vector2f Foods::pos [private]
```

6.27.4.2 rect

```
sf::IntRect Foods::rect [private]
```

6.27.4.3 type

```
TypeObject Foods::type [private]
```

Get The Sprite of the object.

Returns

The Sprite of the object

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

- Objects/Foods/[Foods.hpp](#)
- Objects/Foods/[Foods.cpp](#)

6.28 gui::Handler Class Reference

[Handler](#) class.

```
#include <Handler.hpp>
```

Collaboration diagram for gui::Handler:

gui::Handler
- commands
+ Handler() + ~Handler() + Handler() + Handler() + operator=() + operator=() + handle() - mszCommand() - bctCommand() - tnaCommand() - pnwCommand() - ppoCommand() - plvCommand() - pinCommand() - pexCommand() - pbcCommand() - picCommand() - pieCommand() - pfkCommand() - pdrCommand() - pgtCommand() - pdiCommand() - enwCommand() - eboCommand() - ediCommand() - sgtCommand() - sstCommand() - segCommand() - smgCommand() - sucCommand() - sbpCommand()

Public Member Functions

- [Handler](#) ()=default
Construct a new [Handler](#) object.
- [~Handler](#) ()=default
Destroy the [Handler](#) object.
- [Handler](#) (const [Handler](#) &)=delete
Construct a new [Handler](#) object by copy.
- [Handler](#) ([Handler](#) &&)=default
Construct a new [Handler](#) object by move.
- [Handler](#) & [operator=](#) (const [Handler](#) &)=delete
Assign a [Handler](#) object by copy.

- `Handler & operator= (Handler &&)=default`
Assign a *Handler* object by move.
- `void handle (const std::string &keyword, gui::Data &game, std::istream &iss)`
handle the command

Static Private Member Functions

- `static void mszCommand (std::istream &iss, gui::Data &game)`
msz command
- `static void bctCommand (std::istream &iss, gui::Data &game)`
bct command
- `static void tnaCommand (std::istream &iss, gui::Data &game)`
tna command
- `static void pnwCommand (std::istream &iss, gui::Data &game)`
pnw command
- `static void ppoCommand (std::istream &iss, gui::Data &game)`
ppo command
- `static void plvCommand (std::istream &iss, gui::Data &game)`
plv command
- `static void pinCommand (std::istream &iss, gui::Data &game)`
pin command
- `static void pexCommand (std::istream &iss, gui::Data &game)`
pex command
- `static void pbcCommand (std::istream &iss, gui::Data &game)`
pbc command
- `static void picCommand (std::istream &iss, gui::Data &game)`
pic command
- `static void pieCommand (std::istream &iss, gui::Data &game)`
pie command
- `static void pfkCommand (std::istream &iss, gui::Data &game)`
pfk command
- `static void pdrCommand (std::istream &iss, gui::Data &game)`
pdr command
- `static void pgtCommand (std::istream &iss, gui::Data &game)`
pgt command
- `static void pdiCommand (std::istream &iss, gui::Data &game)`
pdi command
- `static void enwCommand (std::istream &iss, gui::Data &game)`
enw command
- `static void eboCommand (std::istream &iss, gui::Data &game)`
ebo command
- `static void ediCommand (std::istream &iss, gui::Data &game)`
edi command
- `static void sgtCommand (std::istream &iss, gui::Data &game)`
sgt command
- `static void sstCommand (std::istream &iss, gui::Data &game)`
sst command
- `static void segCommand (std::istream &iss, gui::Data &game)`
seg command
- `static void smgCommand (std::istream &iss, gui::Data &game)`

- smg command*
- static void [sucCommand](#) (std::istream &iss, [gui::Data](#) &game)
- suc command*
- static void [sbpCommand](#) (std::istream &iss, [gui::Data](#) &game)
- sbp command*

Private Attributes

- std::vector< [Command](#) > [commands](#)

6.28.1 Detailed Description

[Handler](#) class.

6.28.2 Constructor & Destructor Documentation

6.28.2.1 Handler() [1/3]

```
gui::Handler::Handler ( ) [default]
```

Construct a new [Handler](#) object.

6.28.2.2 ~Handler()

```
gui::Handler::~~Handler ( ) [default]
```

Destroy the [Handler](#) object.

6.28.2.3 Handler() [2/3]

```
gui::Handler::Handler (
    const Handler & ) [delete]
```

Construct a new [Handler](#) object by copy.

6.28.2.4 Handler() [3/3]

```
gui::Handler::Handler (
    Handler && ) [default]
```

Construct a new [Handler](#) object by move.

6.28.3 Member Function Documentation

6.28.3.1 bctCommand()

```
void gui::Handler::bctCommand (
    std::istringstream & iss,
    gui::Data & game ) [static], [private]
```

bct command

Parameters

<i>iss</i>	
<i>game</i>	

6.28.3.2 eboCommand()

```
void gui::Handler::eboCommand (
    std::istringstream & iss,
    gui::Data & game ) [static], [private]
```

ebo command

Parameters

<i>iss</i>	
<i>game</i>	

6.28.3.3 ediCommand()

```
void gui::Handler::ediCommand (
    std::istringstream & iss,
    gui::Data & game ) [static], [private]
```

edi command

Parameters

<i>iss</i>	
<i>game</i>	

6.28.3.4 enwCommand()

```
void gui::Handler::enwCommand (
    std::istream & iss,
    gui::Data & game ) [static], [private]
```

enw command

Parameters

<i>iss</i>	
<i>game</i>	

6.28.3.5 handle()

```
void gui::Handler::handle (
    const std::string & keyword,
    gui::Data & game,
    std::istream & iss )
```

handle the command

Parameters

<i>keyword</i>	
<i>game</i>	
<i>iss</i>	

6.28.3.6 mszCommand()

```
void gui::Handler::mszCommand (
    std::istream & iss,
    gui::Data & game ) [static], [private]
```

msz command

Parameters

<i>iss</i>	
<i>game</i>	

6.28.3.7 operator=() [1/2]

```
Handler& gui::Handler::operator= (
    const Handler & ) [delete]
```

Assign a [Handler](#) object by copy.

Returns

[Handler&](#)

6.28.3.8 operator=() [2/2]

```
Handler& gui::Handler::operator= (
    Handler && ) [default]
```

Assign a [Handler](#) object by move.

Returns

[Handler&](#)

6.28.3.9 pbcCommand()

```
void gui::Handler::pbcCommand (
    std::istringstream & iss,
    gui::Data & game ) [static], [private]
```

pbc command

Parameters

<i>iss</i>	
<i>game</i>	

6.28.3.10 pdiCommand()

```
void gui::Handler::pdiCommand (
    std::istream & iss,
    gui::Data & game ) [static], [private]
```

pdi command

Parameters

<i>iss</i>	
<i>game</i>	

6.28.3.11 pdrCommand()

```
void gui::Handler::pdrCommand (
    std::istream & iss,
    gui::Data & game ) [static], [private]
```

pdr command

Parameters

<i>iss</i>	
<i>game</i>	

6.28.3.12 pexCommand()

```
void gui::Handler::pexCommand (
    std::istream & iss,
    gui::Data & game ) [static], [private]
```

pex command

Parameters

<i>iss</i>	
<i>game</i>	

6.28.3.13 pfkCommand()

```
void gui::Handler::pfkCommand (
```

```
std::istringstream & iss,  
gui::Data & game ) [static], [private]
```

pfk command

Parameters

<i>iss</i>	
<i>game</i>	

6.28.3.14 pgtCommand()

```
void gui::Handler::pgtCommand (  
    std::istringstream & iss,  
    gui::Data & game ) [static], [private]
```

pgt command

Parameters

<i>iss</i>	
<i>game</i>	

6.28.3.15 picCommand()

```
void gui::Handler::picCommand (  
    std::istringstream & iss,  
    gui::Data & game ) [static], [private]
```

pic command

Parameters

<i>iss</i>	
<i>game</i>	

6.28.3.16 pieCommand()

```
void gui::Handler::pieCommand (  
    std::istringstream & iss,  
    gui::Data & game ) [static], [private]
```

pie command

Parameters

<i>iss</i>	
<i>game</i>	

6.28.3.17 pinCommand()

```
void gui::Handler::pinCommand (
    std::istream & iss,
    gui::Data & game ) [static], [private]
```

pin command

Parameters

<i>iss</i>	
<i>game</i>	

6.28.3.18 plvCommand()

```
void gui::Handler::plvCommand (
    std::istream & iss,
    gui::Data & game ) [static], [private]
```

plv command

Parameters

<i>iss</i>	
<i>game</i>	

6.28.3.19 pnwCommand()

```
void gui::Handler::pnwCommand (
    std::istream & iss,
    gui::Data & game ) [static], [private]
```

pnw command

Parameters

<i>iss</i>	
<i>game</i>	

6.28.3.20 ppoCommand()

```
void gui::Handler::ppoCommand (
    std::istream & iss,
    gui::Data & game ) [static], [private]
```

ppo command

Parameters

<i>iss</i>	
<i>game</i>	

6.28.3.21 sbpCommand()

```
void gui::Handler::sbpCommand (
    std::istream & iss,
    gui::Data & game ) [static], [private]
```

sbp command

Parameters

<i>iss</i>	
<i>game</i>	

6.28.3.22 segCommand()

```
void gui::Handler::segCommand (
    std::istream & iss,
    gui::Data & game ) [static], [private]
```

seg command

Parameters

<i>iss</i>	
<i>game</i>	

6.28.3.23 sgtCommand()

```
void gui::Handler::sgtCommand (
    std::istream & iss,
    gui::Data & game ) [static], [private]
```

sgt command

Parameters

<i>iss</i>	
<i>game</i>	

6.28.3.24 smgCommand()

```
void gui::Handler::smgCommand (
    std::istream & iss,
    gui::Data & game ) [static], [private]
```

smg command

Parameters

<i>iss</i>	
<i>game</i>	

6.28.3.25 sstCommand()

```
void gui::Handler::sstCommand (
    std::istream & iss,
    gui::Data & game ) [static], [private]
```

sst command

Parameters

<i>iss</i>	
<i>game</i>	

6.28.3.26 sucCommand()

```
void gui::Handler::sucCommand (
    std::istream & iss,
    gui::Data & game ) [static], [private]
```

suc command

Parameters

<i>iss</i>	
<i>game</i>	

6.28.3.27 tnaCommand()

```
void gui::Handler::tnaCommand (
    std::istream & iss,
    gui::Data & game ) [static], [private]
```

tna command

Parameters

<i>iss</i>	
<i>game</i>	

6.28.4 Field Documentation

6.28.4.1 commands

```
std::vector< Command > gui::Handler::commands [private]
```

Initial value:

```
= {
    { "msz", mszCommand },
    { "bct", bctCommand },
    { "tna", tnaCommand },
    { "pnw", pnwCommand },
    { "ppo", ppoCommand },
    { "plv", plvCommand },
    { "pin", pinCommand },
    { "pex", pexCommand },
    { "pbc", pbcCommand },
    { "pic", picCommand },
    { "pie", pieCommand },
    { "pfk", pfkCommand },
    { "pdr", pdrCommand },
    { "pgt", pgtCommand },
    { "pdi", pdiCommand },
    { "enw", enwCommand },
    { "ebo", eboCommand },
    { "edi", ediCommand },
    { "sgt", sgtCommand },
    { "sst", sstCommand },
    { "seg", segCommand },
    { "smg", smgCommand },
    { "suc", sucCommand },
    { "sbp", sbpCommand },
}
```

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

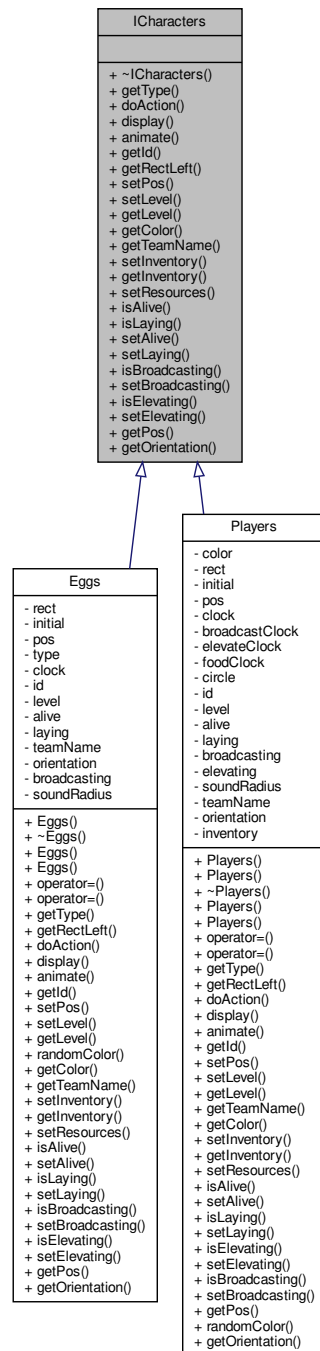
- [Network/Handlers/Handler.hpp](#)
- [Network/Handlers/Commands/bct.cpp](#)
- [Network/Handlers/Commands/ebo.cpp](#)
- [Network/Handlers/Commands/edi.cpp](#)
- [Network/Handlers/Commands/enw.cpp](#)
- [Network/Handlers/Commands/msz.cpp](#)
- [Network/Handlers/Commands/pbc.cpp](#)
- [Network/Handlers/Commands/pdi.cpp](#)
- [Network/Handlers/Commands/pdr.cpp](#)
- [Network/Handlers/Commands/pex.cpp](#)
- [Network/Handlers/Commands/pfk.cpp](#)
- [Network/Handlers/Commands/pgt.cpp](#)
- [Network/Handlers/Commands/pic.cpp](#)
- [Network/Handlers/Commands/pie.cpp](#)
- [Network/Handlers/Commands/pin.cpp](#)
- [Network/Handlers/Commands/plv.cpp](#)
- [Network/Handlers/Commands/pnw.cpp](#)
- [Network/Handlers/Commands/ppo.cpp](#)
- [Network/Handlers/Commands/sbp.cpp](#)
- [Network/Handlers/Commands/seg.cpp](#)
- [Network/Handlers/Commands/sgt.cpp](#)
- [Network/Handlers/Commands/msg.cpp](#)
- [Network/Handlers/Commands/sst.cpp](#)
- [Network/Handlers/Commands/suc.cpp](#)
- [Network/Handlers/Commands/tna.cpp](#)
- [Network/Handlers/Handler.cpp](#)

6.29 ICharacters Class Reference

Interface for the characters.

```
#include <ICharacters.hpp>
```

Inheritance diagram for ICharacters:



Collaboration diagram for ICharacters:



Public Member Functions

- virtual [~ICharacters](#) () noexcept=default
Destroy of the [ICharacters](#) object.
- virtual [CharacterType](#) [getType](#) () const =0
Get the Type object.
- virtual void [doAction](#) ()=0
Do specific action for the character.
- virtual void [display](#) (sf::RenderWindow &window, sf::Sprite &sprite, [gui::ItemDrawer](#) &drawer, int timeUnit, [SoundBox](#) &soundBox)=0
Display the character.
- virtual void [animate](#) (sf::Sprite &sprite)=0
Animate the character.
- virtual const int & [getId](#) ()=0
Get the Id of the character.
- virtual const int & [getRectLeft](#) ()=0
Get the Rect Left of the character.

- virtual void [setPos](#) (sf::Vector2f pos, Orientation orientation)=0
Set the Pos of the character.
- virtual void [setLevel](#) (int level)=0
Set the Level of the character.
- virtual const int & [getLevel](#) ()=0
Get the Level of the character.
- virtual const std::string & [getColor](#) ()=0
Get the Color of the character.
- virtual const std::string & [getTeamName](#) ()=0
Get the Team Name of the character.
- virtual void [setInventory](#) (const std::array< int, 7 > &res)=0
Set the Inventory of the character.
- virtual const std::array< int, 7 > & [getInventory](#) ()=0
Get the Inventory of the character.
- virtual void [setResources](#) (const int &res, const int &nb)=0
Set the Resources of the character.
- virtual const bool & [isAlive](#) ()=0
Check if the character is alive.
- virtual const bool & [isLaying](#) ()=0
Check if the character is laying.
- virtual void [setAlive](#) (bool alive)=0
Set the Alive boolean of the character.
- virtual void [setLaying](#) (bool laying)=0
Set the Laying boolean of the character.
- virtual const bool & [isBroadcasting](#) ()=0
Check if the character is broadcasting.
- virtual void [setBroadcasting](#) (bool broadcasting)=0
Set the Broadcasting boolean of the character.
- virtual const int & [isElevating](#) ()=0
Check if the character is elevating.
- virtual void [setElevating](#) (int elevating)=0
Set the Elevating process of the character.
- virtual const sf::Vector2f & [getPos](#) ()=0
Get the Pos of the character.
- virtual const Orientation & [getOrientation](#) ()=0
Get the Orientation of the character.

6.29.1 Detailed Description

Interface for the characters.

6.29.2 Constructor & Destructor Documentation

6.29.2.1 ~ICharacters()

```
virtual ICharacters::~ICharacters ( ) [virtual], [default], [noexcept]
```

Destroy of the [ICharacters](#) object.

6.29.3 Member Function Documentation

6.29.3.1 `animate()`

```
virtual void ICharacters::animate (
    sf::Sprite & sprite ) [pure virtual]
```

Animate the character.

Parameters

<i>sprite</i>	
---------------	--

Implemented in [Players](#), and [Eggs](#).

6.29.3.2 `display()`

```
virtual void ICharacters::display (
    sf::RenderWindow & window,
    sf::Sprite & sprite,
    gui::ItemDrawer & drawer,
    int timeUnit,
    SoundBox & soundBox ) [pure virtual]
```

Display the character.

Parameters

<i>window</i>	
<i>sprite</i>	
<i>drawer</i>	
<i>timeUnit</i>	
<i>soundbox</i>	

Implemented in [Players](#), and [Eggs](#).

6.29.3.3 `doAction()`

```
virtual void ICharacters::doAction ( ) [pure virtual]
```

Do specific action for the character.

Implemented in [Players](#), and [Eggs](#).

6.29.3.4 getColor()

```
virtual const std::string& ICharacters::getColor ( ) [pure virtual]
```

Get the Color of the character.

Returns

const std::string&

Implemented in [Players](#), and [Eggs](#).

6.29.3.5 getId()

```
virtual const int& ICharacters::getId ( ) [pure virtual]
```

Get the Id of the character.

Returns

const int&

Implemented in [Players](#), and [Eggs](#).

6.29.3.6 getInventory()

```
virtual const std::array<int, 7>& ICharacters::getInventory ( ) [pure virtual]
```

Get the Inventory of the character.

Returns

std::array<int, 7>&

Implemented in [Players](#), and [Eggs](#).

6.29.3.7 getLevel()

```
virtual const int& ICharacters::getLevel ( ) [pure virtual]
```

Get the Level of the character.

Returns

const int&

Implemented in [Players](#), and [Eggs](#).

6.29.3.8 getOrientation()

```
virtual const Orientation& ICharacters::getOrientation ( ) [pure virtual]
```

Get the Orientation of the character.

Returns

const Orientation&

Implemented in [Players](#), and [Eggs](#).

6.29.3.9 getPos()

```
virtual const sf::Vector2f& ICharacters::getPos ( ) [pure virtual]
```

Get the Pos of the character.

Returns

sf::Vector2f

Implemented in [Players](#), and [Eggs](#).

6.29.3.10 getRectLeft()

```
virtual const int& ICharacters::getRectLeft ( ) [pure virtual]
```

Get the Rect Left of the character.

Returns

const int&

Implemented in [Players](#), and [Eggs](#).

6.29.3.11 getTeamName()

```
virtual const std::string& ICharacters::getTeamName ( ) [pure virtual]
```

Get the Team Name of the character.

Returns

const std::string&

Implemented in [Players](#), and [Eggs](#).

6.29.3.12 `getType()`

```
virtual CharacterType ICharacters::getType ( ) const [pure virtual]
```

Get the Type object.

Returns

`CharacterType`

Implemented in [Players](#), and [Eggs](#).

6.29.3.13 `isAlive()`

```
virtual const bool& ICharacters::isAlive ( ) [pure virtual]
```

Check if the character is alive.

Returns

`true`

`false`

Implemented in [Players](#), and [Eggs](#).

6.29.3.14 `isBroadcasting()`

```
virtual const bool& ICharacters::isBroadcasting ( ) [pure virtual]
```

Check if the character is broadcasting.

Returns

`true`

`false`

Implemented in [Players](#), and [Eggs](#).

6.29.3.15 isElevating()

```
virtual const int& ICharacters::isElevating ( ) [pure virtual]
```

Check if the character is elevating.

Returns

int

Implemented in [Players](#), and [Eggs](#).

6.29.3.16 isLaying()

```
virtual const bool& ICharacters::isLaying ( ) [pure virtual]
```

Check if the character is laying.

Returns

true

false

Implemented in [Players](#), and [Eggs](#).

6.29.3.17 setAlive()

```
virtual void ICharacters::setAlive (
    bool alive ) [pure virtual]
```

Set the Alive boolean of the character.

Parameters

<i>alive</i>	
--------------	--

Implemented in [Players](#), and [Eggs](#).

6.29.3.18 setBroadcasting()

```
virtual void ICharacters::setBroadcasting (
    bool broadcasting ) [pure virtual]
```

Set the Broadcasting boolean of the character.

Parameters

<i>broadcasting</i>	
---------------------	--

Implemented in [Eggs](#), and [Players](#).

6.29.3.19 setElevating()

```
virtual void ICharacters::setElevating (
    int elevating ) [pure virtual]
```

Set the Elevating process of the character.

Parameters

<i>elevating</i>	
------------------	--

Implemented in [Players](#), and [Eggs](#).

6.29.3.20 setInventory()

```
virtual void ICharacters::setInventory (
    const std::array< int, 7 > & res ) [pure virtual]
```

Set the Inventory of the character.

Parameters

<i>res</i>	
------------	--

Implemented in [Players](#), and [Eggs](#).

6.29.3.21 setLaying()

```
virtual void ICharacters::setLaying (
    bool laying ) [pure virtual]
```

Set the Laying boolean of the character.

Parameters

<i>laying</i>	
---------------	--

Implemented in [Players](#), and [Eggs](#).

6.29.3.22 setLevel()

```
virtual void ICharacters::setLevel (
    int level ) [pure virtual]
```

Set the Level of the character.

Parameters

<i>level</i>	
--------------	--

Implemented in [Players](#), and [Eggs](#).

6.29.3.23 setPos()

```
virtual void ICharacters::setPos (
    sf::Vector2f pos,
    Orientation orientation ) [pure virtual]
```

Set the Pos of the character.

Parameters

<i>pos</i>	
<i>orientation</i>	

Implemented in [Players](#), and [Eggs](#).

6.29.3.24 setResources()

```
virtual void ICharacters::setResources (
    const int & res,
    const int & nb ) [pure virtual]
```

Set the Resources of the character.

Parameters

<i>res</i>	
<i>nb</i>	

Implemented in [Players](#), and [Eggs](#).

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

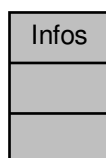
- Characters/[ICharacters.hpp](#)

6.30 Infos Struct Reference

[Infos](#) struct.

```
#include <Tile.hpp>
```

Collaboration diagram for Infos:



6.30.1 Detailed Description

[Infos](#) struct.

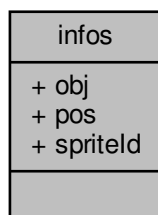
The documentation for this struct was generated from the following file:

- Map/[Tile.hpp](#)

6.31 infos Struct Reference

```
#include <Tile.hpp>
```

Collaboration diagram for infos:



Data Fields

- int [obj](#)
- sf::Vector2f [pos](#)
- int [spriteId](#)

6.31.1 Field Documentation

6.31.1.1 obj

```
int infos::obj
```

6.31.1.2 pos

```
sf::Vector2f infos::pos
```

6.31.1.3 spriteId

```
int infos::spriteId
```

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

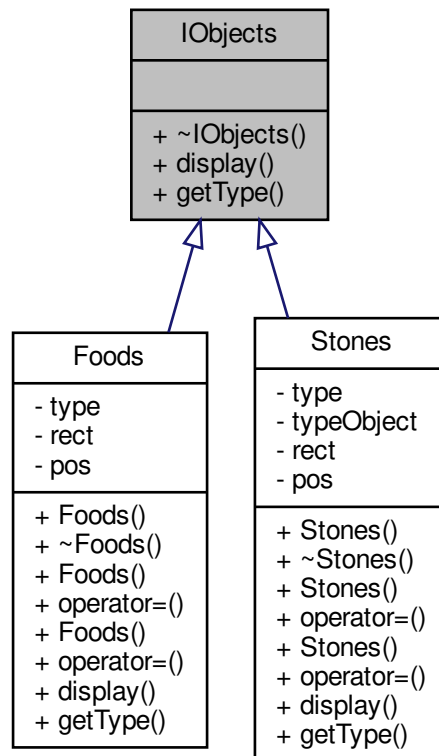
- Map/[Tile.hpp](#)

6.32 IObjects Class Reference

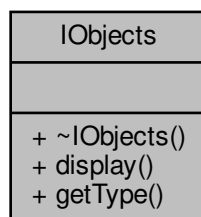
[IObjects](#) class.

```
#include <IObjects.hpp>
```

Inheritance diagram for IObjects:



Collaboration diagram for IObjects:



Public Member Functions

- virtual `~IObjects()` noexcept=default
Destroy the *IObjects* object.

- virtual void [display](#) (sf::RenderWindow &window, sf::Sprite &sprite, float size)=0
Display the object.
- virtual const [TypeObject](#) & [getType](#) ()=0
Get The type of the object.

6.32.1 Detailed Description

[IObjects](#) class.

6.32.2 Constructor & Destructor Documentation

6.32.2.1 ~IObjects()

```
virtual IObjects::~~IObjects ( ) [virtual], [default], [noexcept]
```

Destroy the [IObjects](#) object.

6.32.3 Member Function Documentation

6.32.3.1 display()

```
virtual void IObjects::display (
    sf::RenderWindow & window,
    sf::Sprite & sprite,
    float size ) [pure virtual]
```

Display the object.

Parameters

<i>window</i>	The window where the object will be displayed
<i>sprite</i>	The sprite of the object
<i>size</i>	The size of the object

Implemented in [Stones](#), and [Foods](#).

6.32.3.2 getType()

```
virtual const TypeObject& IObjects::getType ( ) [pure virtual]
```

Get The type of the object.

Returns

The type of the object

Implemented in [Stones](#), and [Foods](#).

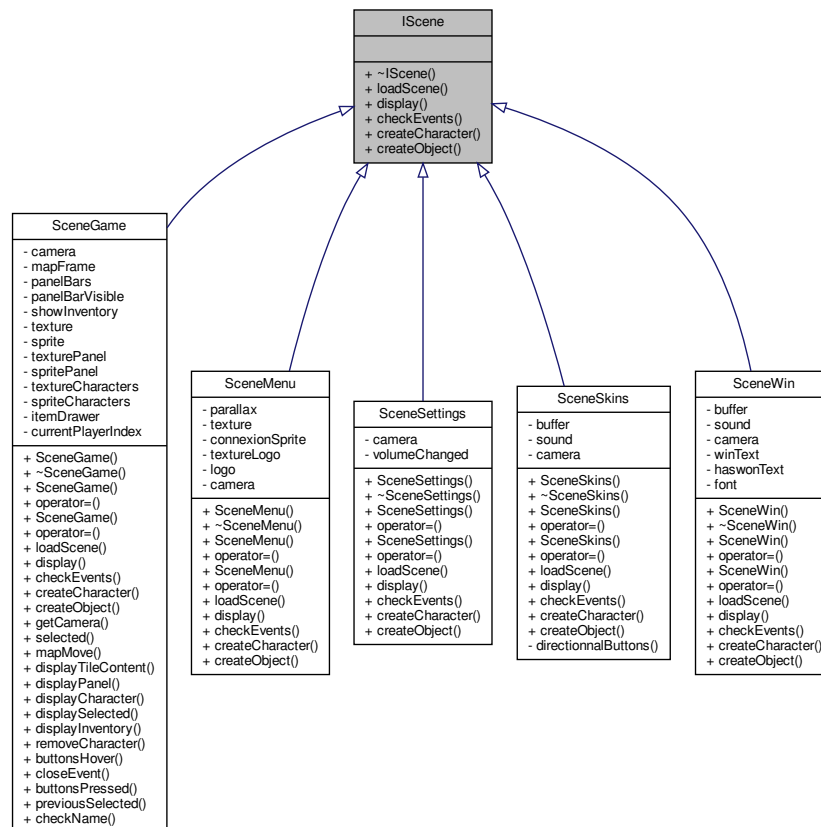
The documentation for this class was generated from the following file:

- [Objects/Objects.hpp](#)

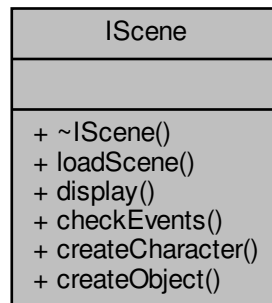
6.33 IScene Class Reference

```
#include <IScene.hpp>
```

Inheritance diagram for IScene:



Collaboration diagram for IScene:



Public Member Functions

- virtual [~IScene](#) () noexcept=default
- virtual void [loadScene](#) ([gui::SceneManager](#) &manager, [gui::Window](#) &>window, [gui::Data](#) &data)=0
- virtual void [display](#) ([gui::SceneManager](#) &manager, [gui::Window](#) &>window, [gui::Data](#) &data)=0
- virtual void [checkEvents](#) ([gui::SceneManager](#) &manager, [gui::Window](#) &>window, [gui::Data](#) &data)=0
- virtual void [createCharacter](#) ()=0
- virtual void [createObject](#) ([gui::Data](#) &data)=0

6.33.1 Constructor & Destructor Documentation

6.33.1.1 ~IScene()

```
virtual IScene::~~IScene ( ) [virtual], [default], [noexcept]
```

6.33.2 Member Function Documentation

6.33.2.1 checkEvents()

```
virtual void IScene::checkEvents (
    gui::SceneManager & manager,
    gui::Window & window,
    gui::Data & data ) [pure virtual]
```

Implemented in [SceneWin](#), [SceneSkins](#), [SceneSettings](#), [SceneMenu](#), and [SceneGame](#).

6.33.2.2 createCharacter()

```
virtual void IScene::createCharacter ( ) [pure virtual]
```

Implemented in [SceneWin](#), [SceneSkins](#), [SceneSettings](#), [SceneMenu](#), and [SceneGame](#).

6.33.2.3 createObject()

```
virtual void IScene::createObject (
    gui::Data & data ) [pure virtual]
```

Implemented in [SceneGame](#).

6.33.2.4 display()

```
virtual void IScene::display (
    gui::SceneManager & manager,
    gui::Window & window,
    gui::Data & data ) [pure virtual]
```

Implemented in [SceneWin](#), [SceneSkins](#), [SceneSettings](#), [SceneMenu](#), and [SceneGame](#).

6.33.2.5 loadScene()

```
virtual void IScene::loadScene (
    gui::SceneManager & manager,
    gui::Window & window,
    gui::Data & data ) [pure virtual]
```

Implemented in [SceneWin](#), [SceneSkins](#), [SceneSettings](#), [SceneMenu](#), and [SceneGame](#).

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

- [Scene/IScene.hpp](#)

6.34 gui::ItemDrawer Class Reference

[ItemDrawer](#) class.

```
#include <ItemDrawer.hpp>
```

Collaboration diagram for gui::ItemDrawer:

gui::ItemDrawer
<ul style="list-style-type: none"> - levelRect - itemRect - meteorRect - itemSprite
<ul style="list-style-type: none"> + ItemDrawer() + ~ItemDrawer() + ItemDrawer() + ItemDrawer() + operator=() + operator=() + setSprite() + drawItem() + drawLevel() + drawMeteor()

Public Member Functions

- [ItemDrawer](#) ()=default
Construct a new Item Drawer object.
- [~ItemDrawer](#) ()=default
Destroy the Item Drawer object.
- [ItemDrawer](#) (const [ItemDrawer](#) &other)=delete
Construct a new Item Drawer object by copy = deleted.
- [ItemDrawer](#) ([ItemDrawer](#) &&other)=default
Construct a new Item Drawer object by move = default.
- [ItemDrawer](#) & [operator=](#) (const [ItemDrawer](#) &other)=delete
Assign a Item Drawer object by copy = deleted.
- [ItemDrawer](#) & [operator=](#) ([ItemDrawer](#) &&other)=default
Assign a Item Drawer object by move = default.
- void [setSprite](#) (sf::Sprite sprite)
Set the Sprite.
- void [drawItem](#) (sf::RenderWindow &>window, int id, int x, int y, float scale)
draw the item
- void [drawLevel](#) (sf::RenderWindow &>window, int id, int x, int y, float scale)
draw the level
- void [drawMeteor](#) (sf::RenderWindow &>window, int id, int x, int y, float scale)
draw the Meteor

Private Attributes

- `std::vector< sf::IntRect > levelRect`
- `std::vector< sf::IntRect > itemRect`
- `std::vector< sf::IntRect > meteorRect`
- `sf::Sprite itemSprite`

6.34.1 Detailed Description

`ItemDrawer` class.

6.34.2 Constructor & Destructor Documentation

6.34.2.1 `ItemDrawer()` [1/3]

```
gui::ItemDrawer::ItemDrawer ( ) [default]
```

Construct a new Item Drawer object.

6.34.2.2 `~ItemDrawer()`

```
gui::ItemDrawer::~~ItemDrawer ( ) [default]
```

Destroy the Item Drawer object.

6.34.2.3 `ItemDrawer()` [2/3]

```
gui::ItemDrawer::ItemDrawer (
    const ItemDrawer & other ) [delete]
```

Construct a new Item Drawer object by copy = deleted.

Parameters

<i>other</i>	
--------------	--

6.34.2.4 ItemDrawer() [3/3]

```
gui::ItemDrawer::ItemDrawer (
    ItemDrawer && other ) [default]
```

Construct a new Item Drawer object by move = default.

Parameters

<i>other</i>	
--------------	--

Returns

[ItemDrawer&](#)

6.34.3 Member Function Documentation

6.34.3.1 drawItem()

```
void gui::ItemDrawer::drawItem (
    sf::RenderWindow & window,
    int id,
    int x,
    int y,
    float scale )
```

draw the item

Parameters

<i>window</i>	
<i>id</i>	
<i>x</i>	
<i>y</i>	
<i>scale</i>	

6.34.3.2 drawLevel()

```
void gui::ItemDrawer::drawLevel (
    sf::RenderWindow & window,
    int id,
    int x,
    int y,
    float scale )
```

draw the level

Parameters

<i>window</i>	
<i>id</i>	
<i>x</i>	
<i>y</i>	
<i>scale</i>	

6.34.3.3 drawMeteor()

```
void gui::ItemDrawer::drawMeteor (
    sf::RenderWindow & window,
    int id,
    int x,
    int y,
    float scale )
```

draw the Meteor

Parameters

<i>window</i>	
<i>id</i>	
<i>x</i>	
<i>y</i>	
<i>scale</i>	

6.34.3.4 operator=() [1/2]

```
ItemDrawer& gui::ItemDrawer::operator= (
    const ItemDrawer & other ) [delete]
```

Assign a Item Drawer object by copy = deleted.

Parameters

<i>other</i>	
--------------	--

Returns

[ItemDrawer&](#)

6.34.3.5 operator=() [2/2]

```
ItemDrawer& gui::ItemDrawer::operator= (
    ItemDrawer && other ) [default]
```

Assign a Item Drawer object by move = default.

Parameters

<i>other</i>	
--------------	--

Returns

ItemDrawer&

6.34.3.6 setSprite()

```
void gui::ItemDrawer::setSprite (
    sf::Sprite sprite )
```

Set the Sprite.

Parameters

<i>sprite</i>	
---------------	--

6.34.4 Field Documentation

6.34.4.1 itemRect

```
std::vector< sf::IntRect > gui::ItemDrawer::itemRect [private]
```

Initial value:

```
= {
    sf::IntRect(800, 1560, 75, 95),
    sf::IntRect(795, 1665, 75, 80),
    sf::IntRect(960, 1670, 80, 70),
    sf::IntRect(1190, 1670, 75, 80),
    sf::IntRect(1040, 1675, 75, 80),
    sf::IntRect(1115, 1675, 65, 80),
    sf::IntRect(885, 1660, 65, 80),
    sf::IntRect(800, 1560, 75, 95),
    sf::IntRect(895, 1560, 75, 95),
    sf::IntRect(995, 1560, 75, 95),
    sf::IntRect(1095, 1560, 75, 95),
    sf::IntRect(1200, 1560, 75, 95),
    sf::IntRect(1300, 1560, 75, 95),
    sf::IntRect(1400, 1560, 75, 95),
    sf::IntRect(1500, 1560, 75, 95),
    sf::IntRect(1610, 1560, 75, 95),
```

```
sf::IntRect(1720, 1560, 75, 95),
sf::IntRect(1815, 1560, 75, 95),
sf::IntRect(1920, 1560, 75, 95),
sf::IntRect(2025, 1560, 75, 95),
sf::IntRect(2120, 1560, 85, 95),
sf::IntRect(2230, 1560, 75, 95),
sf::IntRect(2330, 1560, 75, 95),
sf::IntRect(2432, 1560, 85, 95)
}
```

6.34.4.2 itemSprite

```
sf::Sprite gui::ItemDrawer::itemSprite [private]
```

6.34.4.3 levelRect

```
std::vector< sf::IntRect > gui::ItemDrawer::levelRect [private]
```

Initial value:

```
= {
    sf::IntRect(2215, 50, 170, 130),
    sf::IntRect(2410, 50, 170, 130),
    sf::IntRect(2610, 50, 170, 130),
    sf::IntRect(2310, 190, 170, 130),
    sf::IntRect(2520, 190, 170, 130),
    sf::IntRect(2215, 320, 170, 130),
    sf::IntRect(2410, 320, 170, 130),
    sf::IntRect(2610, 320, 170, 130)
}
```

6.34.4.4 meteorRect

```
std::vector< sf::IntRect > gui::ItemDrawer::meteorRect [private]
```

Initial value:

```
= {
    sf::IntRect(2775, 1265, 345, 255),
    sf::IntRect(3150, 1330, 360, 225),
    sf::IntRect(2710, 1575, 100, 95),
    sf::IntRect(2815, 1575, 115, 85),
    sf::IntRect(2950, 1570, 90, 110)
}
```

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

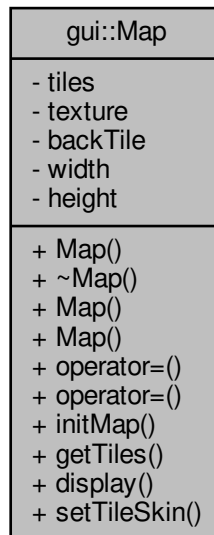
- [ItemDrawer/ItemDrawer.hpp](#)
- [ItemDrawer/ItemDrawer.cpp](#)

6.35 gui::Map Class Reference

Map Class.

```
#include <Map.hpp>
```

Collaboration diagram for gui::Map:



Public Member Functions

- **Map** ()=default
Construct a new Map object.
- **~Map** ()=default
Destroy the Map object.
- **Map** (const **Map** &other)=delete
Construct a new Map object by copy.
- **Map** (**Map** &&other)=default
Construct a new Map object.
- **Map** & **operator=** (const **Map** &other)=delete
Assign a Map object by copy.
- **Map** & **operator=** (**Map** &&other)=default
Assign a Map object by move.
- void **initMap** (unsigned int **width**, unsigned int **height**)
Init the map.
- std::vector< std::vector< **Tile** > > & **getTiles** ()
Get the tiles.
- void **display** (sf::RenderWindow &window, **gui::ItemDrawer** &itemDrawer)
Display the map.
- void **setTileSkin** (int tileSkin)
Set the Tile Skin.

Private Attributes

- `std::vector< std::vector< Tile > > tiles`
- `sf::Texture texture`
- `sf::Sprite backTile`
- `unsigned int width {0}`
- `unsigned int height {0}`

6.35.1 Detailed Description

[Map](#) Class.

6.35.2 Constructor & Destructor Documentation

6.35.2.1 [Map\(\)](#) [1/3]

```
gui::Map::Map ( ) [default]
```

Construct a new [Map](#) object.

6.35.2.2 [~Map\(\)](#)

```
gui::Map::~Map ( ) [default]
```

Destroy the [Map](#) object.

6.35.2.3 [Map\(\)](#) [2/3]

```
gui::Map::Map (
    const Map & other ) [delete]
```

Construct a new [Map](#) object by copy.

Parameters

<i>other</i>	
--------------	--

6.35.2.4 Map() [3/3]

```
gui::Map::Map (
    Map && other ) [default]
```

Construct a new [Map](#) object.

Parameters

<i>other</i>	
--------------	--

6.35.3 Member Function Documentation

6.35.3.1 display()

```
void gui::Map::display (
    sf::RenderWindow & window,
    gui::ItemDrawer & itemDrawer )
```

Display the map.

Parameters

<i>window</i>	The window to display the map
<i>itemDrawer</i>	The itemDrawer to display the map

6.35.3.2 getTiles()

```
std::vector< std::vector< Tile > > & gui::Map::getTiles ( )
```

Get the tiles.

Returns

```
const std::vector<std::vector<Tile>> &
```

6.35.3.3 initMap()

```
void gui::Map::initMap (
    unsigned int width,
    unsigned int height )
```

Init the map.

Parameters

<i>width</i>	
<i>height</i>	

Exceptions

<i>Errors::ErrorTexture</i>	
---	--

6.35.3.4 operator=() [1/2]

```
Map& gui::Map::operator= (  
    const Map & other ) [delete]
```

Assign a [Map](#) object by copy.

Parameters

<i>other</i>	
--------------	--

Returns

[Map](#)&

6.35.3.5 operator=() [2/2]

```
Map& gui::Map::operator= (  
    Map && other ) [default]
```

Assign a [Map](#) object by move.

Parameters

<i>other</i>	
--------------	--

Returns

[Map](#)&

6.35.3.6 setTileSkin()

```
void gui::Map::setTileSkin (
    int tileSkin )
```

Set the [Tile](#) Skin.

Parameters

<i>tileSkin</i>	
-----------------	--

6.35.4 Field Documentation

6.35.4.1 backTile

```
sf::Sprite gui::Map::backTile [private]
```

6.35.4.2 height

```
unsigned int gui::Map::height {0} [private]
```

6.35.4.3 texture

```
sf::Texture gui::Map::texture [private]
```

6.35.4.4 tiles

```
std::vector<std::vector<Tile> > gui::Map::tiles [private]
```

6.35.4.5 width

```
unsigned int gui::Map::width {0} [private]
```

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

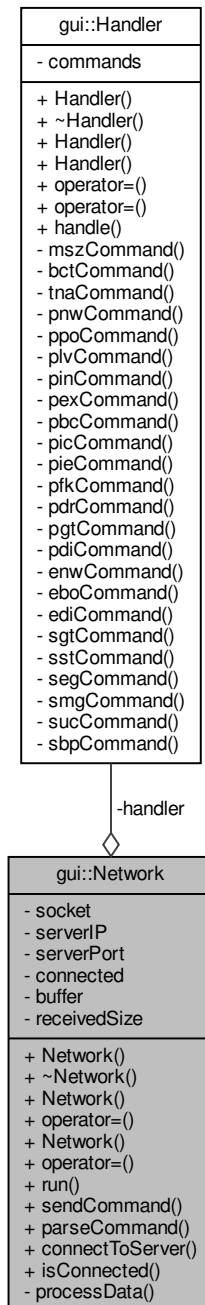
- [Map/Map.hpp](#)
- [Map/Map.cpp](#)

6.36 gui::Network Class Reference

[Network](#) class.

```
#include <Network.hpp>
```

Collaboration diagram for gui::Network:



Public Member Functions

- [Network](#) (unsigned int port, const char *path)
Construct a new [Network](#) object.
- [~Network](#) ()
Destroy the [Network](#) object.
- [Network](#) ([Network](#) &&)=default
Construct a new [Network](#) object by move.
- [Network](#) & operator= ([Network](#) &&)=default
Assign a [Network](#) object by move.
- [Network](#) (const [Network](#) &)=delete
Construct a new [Network](#) object.
- [Network](#) & operator= (const [Network](#) &)=delete
Assign a [Network](#) object by copy.
- void [run](#) ([gui::Data](#) &game)
run the network
- int [sendCommand](#) (const std::string &command)
send a command to the server
- void [parseCommand](#) (const std::string &command, [gui::Data](#) &game)
parse the command
- void [connectToServer](#) ()
connect to the server
- bool [isConnected](#) ()
Check if socket is disconnect.

Private Member Functions

- void [processData](#) (const char *data, std::size_t size)
process the data

Private Attributes

- sf::TcpSocket [socket](#)
- sf::IpAddress [serverIP](#)
- unsigned int [serverPort](#)
- bool [connected](#) = false
- char [buffer](#) [[maxBufferSize](#)] {0}
- std::size_t [receivedSize](#) {0}
- [gui::Handler](#) [handler](#)

6.36.1 Detailed Description

[Network](#) class.

6.36.2 Constructor & Destructor Documentation

6.36.2.1 [Network](#)() [1/3]

```
gui::Network::Network (
    unsigned int port,
    const char * path )
```

Construct a new [Network](#) object.

Parameters

<i>port</i>	
<i>path</i>	

6.36.2.2 ~Network()

```
gui::Network::~~Network ( )
```

Destroy the [Network](#) object.

6.36.2.3 Network() [2/3]

```
gui::Network::Network (
    Network && ) [default]
```

Construct a new [Network](#) object by move.

6.36.2.4 Network() [3/3]

```
gui::Network::Network (
    const Network & ) [delete]
```

Construct a new [Network](#) object.

6.36.3 Member Function Documentation**6.36.3.1 connectToServer()**

```
void gui::Network::connectToServer ( )
```

connect to the server

Exceptions

<i>Errors::ErrorNetwork</i>	
-----------------------------	--

6.36.3.2 isConnected()

```
bool gui::Network::isConnected ( )
```

Check if socket is disconnect.

6.36.3.3 operator=() [1/2]

```
Network& gui::Network::operator= (
    const Network & ) [delete]
```

Assign a [Network](#) object by copy.

Returns

[Network&](#)

6.36.3.4 operator=() [2/2]

```
Network& gui::Network::operator= (
    Network && ) [default]
```

Assign a [Network](#) object by move.

Returns

[Network&](#)

6.36.3.5 parseCommand()

```
void gui::Network::parseCommand (
    const std::string & command,
    gui::Data & game )
```

parse the command

Parameters

<i>command</i>	
<i>game</i>	

6.36.3.6 processData()

```
void gui::Network::processData (
    const char * data,
    std::size_t size ) [private]
```

process the data

Parameters

<i>data</i>	
<i>size</i>	

6.36.3.7 run()

```
void gui::Network::run (
    gui::Data & game )
```

run the network

Parameters

<i>game</i>	
-------------	--

6.36.3.8 sendCommand()

```
int gui::Network::sendCommand (
    const std::string & command )
```

send a command to the server

Parameters

<i>command</i>	
----------------	--

Returns

int

6.36.4 Field Documentation

6.36.4.1 buffer

```
char gui::Network::buffer {0} [private]
```

6.36.4.2 connected

```
bool gui::Network::connected = false [private]
```

6.36.4.3 handler

```
gui::Handler gui::Network::handler [private]
```

6.36.4.4 receivedSize

```
std::size_t gui::Network::receivedSize {0} [private]
```

6.36.4.5 serverIP

```
sf::IpAddress gui::Network::serverIP [private]
```

6.36.4.6 serverPort

```
unsigned int gui::Network::serverPort [private]
```

6.36.4.7 socket

```
sf::TcpSocket gui::Network::socket [private]
```

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

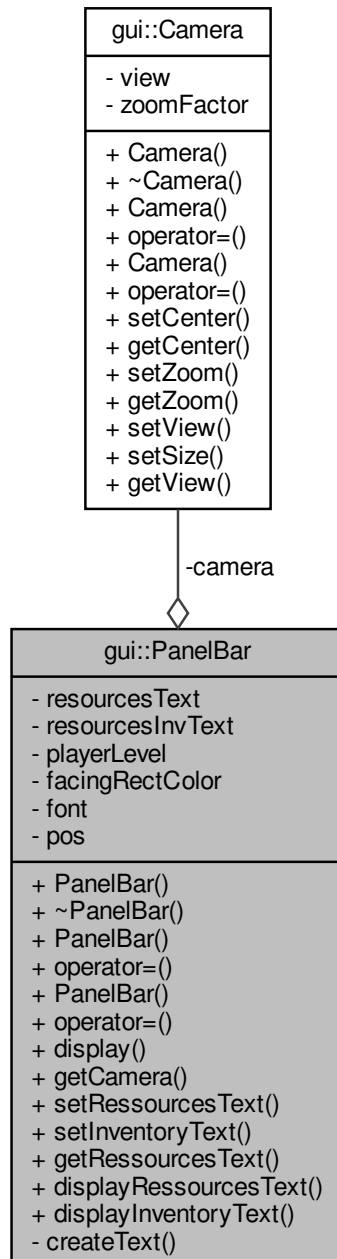
- Network/[Network.hpp](#)
- Network/[Network.cpp](#)

6.37 gui::PanelBar Class Reference

[PanelBar](#) class.

```
#include <PanelBar.hpp>
```

Collaboration diagram for gui::PanelBar:



Public Member Functions

- [PanelBar](#) (sf::Vector2f [pos](#))
Construct a new Panel Bar object.
- [~PanelBar](#) () noexcept=default
Destroy the Panel Bar object.
- [PanelBar](#) (const [PanelBar](#) &panelBar)=delete
Construct a new Panel Bar object by copy.
- [PanelBar](#) & [operator=](#) (const [PanelBar](#) &panelBar)=delete
Assign a Panel Bar object by copy.
- [PanelBar](#) ([PanelBar](#) &&panelBar)=default
Construct a new Panel Bar object by move.
- [PanelBar](#) & [operator=](#) ([PanelBar](#) &&panelBar)=default
Assign a Panel Bar object by move.
- void [display](#) (sf::RenderWindow &>window, sf::Sprite &sprite)
- [gui::Camera](#) & [getCamera](#) ()
- void [setRessourcesText](#) (std::vector< [infos](#) > &resources)
Set the Ressources Text.
- void [setInventoryText](#) (ICharacters &character)
Set the Inventory Text.
- const std::vector< sf::Text > & [getRessourcesText](#) () const
Get the Ressources Text object.
- void [displayRessourcesText](#) ([gui::Window](#) &window, [gui::ItemDrawer](#) &drawer)
Display the Ressources Text.
- void [displayInventoryText](#) ([gui::Window](#) &window, [gui::ItemDrawer](#) &drawer, sf::Sprite &sprite)
Display the Inventory Text.

Private Member Functions

- sf::Text [createText](#) (const std::string &str, const sf::Vector2f &[pos](#))
Create a Text object.

Private Attributes

- std::vector< sf::Text > [resourcesText](#)
- std::vector< sf::Text > [resourcesInvText](#)
- int [playerLevel](#)
- int [facingRectColor](#)
- sf::Font [font](#)
- [gui::Camera](#) [camera](#) {1920, 1080}
- sf::Vector2f [pos](#)

6.37.1 Detailed Description

[PanelBar](#) class.

6.37.2 Constructor & Destructor Documentation

6.37.2.1 PanelBar() [1/3]

```
gui::PanelBar::PanelBar (  
    sf::Vector2f pos )
```

Construct a new Panel Bar object.

Parameters

<i>pos</i>	
------------	--

Exceptions

<i>Errors::ErrorTexture</i>	
---	--

6.37.2.2 ~PanelBar()

```
gui::PanelBar::~~PanelBar ( ) [default], [noexcept]
```

Destroy the Panel Bar object.

6.37.2.3 PanelBar() [2/3]

```
gui::PanelBar::PanelBar (
    const PanelBar & panelBar ) [delete]
```

Construct a new Panel Bar object by copy.

Parameters

<i>panelBar</i>	
-----------------	--

6.37.2.4 PanelBar() [3/3]

```
gui::PanelBar::PanelBar (
    PanelBar && panelBar ) [default]
```

Construct a new Panel Bar object by move.

Parameters

<i>panelBar</i>	
-----------------	--

6.37.3 Member Function Documentation

6.37.3.1 createText()

```
sf::Text gui::PanelBar::createText (
    const std::string & str,
    const sf::Vector2f & pos ) [private]
```

Create a Text object.

Parameters

<i>str</i>	
<i>pos</i>	

Returns

sf::Text

6.37.3.2 display()

```
void gui::PanelBar::display (
    sf::RenderWindow & window,
    sf::Sprite & sprite )
```

6.37.3.3 displayInventoryText()

```
void gui::PanelBar::displayInventoryText (
    gui::Window & window,
    gui::ItemDrawer & drawer,
    sf::Sprite & sprite )
```

Display the Inventory Text.

Parameters

<i>window</i>	
<i>drawer</i>	
<i>sprite</i>	

6.37.3.4 displayRessourcesText()

```
void gui::PanelBar::displayRessourcesText (
    gui::Window & window,
    gui::ItemDrawer & drawer )
```

Display the Ressources Text.

Parameters

<i>window</i>	
<i>drawer</i>	

6.37.3.5 getCamera()

```
gui::Camera & gui::PanelBar::getCamera ( )
```

6.37.3.6 getRessourcesText()

```
const std::vector< sf::Text > & gui::PanelBar::getRessourcesText ( ) const
```

Get the Ressources Text object.

Returns

```
const std::vector<sf::Text>&
```

6.37.3.7 operator=() [1/2]

```
PanelBar& gui::PanelBar::operator= (
    const PanelBar & panelBar ) [delete]
```

Assign a Panel Bar object by copy.

Parameters

<i>panelBar</i>	
-----------------	--

Returns

```
PanelBar&
```

6.37.3.8 operator=() [2/2]

```
PanelBar& gui::PanelBar::operator= (
    PanelBar && panelBar ) [default]
```

Assign a Panel Bar object by move.

Parameters

<i>panelBar</i>	
-----------------	--

Returns

[PanelBar](#)&

6.37.3.9 setInventoryText()

```
void gui::PanelBar::setInventoryText (
    ICharacters & character )
```

Set the Inventory Text.

Parameters

<i>character</i>	
------------------	--

6.37.3.10 setRessourcesText()

```
void gui::PanelBar::setRessourcesText (
    std::vector< infos > & resources )
```

Set the Ressources Text.

Parameters

<i>resources</i>	
------------------	--

6.37.4 Field Documentation

6.37.4.1 camera

```
gui::Camera gui::PanelBar::camera {1920, 1080} [private]
```

6.37.4.2 facingRectColor

```
int gui::PanelBar::facingRectColor [private]
```

6.37.4.3 font

```
sf::Font gui::PanelBar::font [private]
```

6.37.4.4 playerLevel

```
int gui::PanelBar::playerLevel [private]
```

6.37.4.5 pos

```
sf::Vector2f gui::PanelBar::pos [private]
```

6.37.4.6 resourcesInvText

```
std::vector< sf::Text > gui::PanelBar::resourcesInvText [private]
```

6.37.4.7 resourcesText

```
std::vector< sf::Text > gui::PanelBar::resourcesText [private]
```

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

- Components/PanelBar/[PanelBar.hpp](#)
- Components/PanelBar/[PanelBar.cpp](#)

6.38 gui::Parallax Class Reference

[Parallax](#) class.

```
#include <Parallax.hpp>
```

Collaboration diagram for gui::Parallax:

gui::Parallax
- parallax - clock
+ Parallax() + Parallax() + operator=() + Parallax() + operator=() + ~Parallax() + move() + load() + display() + reset()

Public Member Functions

- [Parallax](#) ()
Construct a new [Parallax](#) object.
- [Parallax](#) (const [Parallax](#) &other)=delete
Construct a new [Parallax](#) object by copy.
- [Parallax](#) & [operator=](#) (const [Parallax](#) &other)=delete
Assign a [Parallax](#) object by copy.
- [Parallax](#) ([Parallax](#) &&other)=default
Construct a new [Parallax](#) object by move.
- [Parallax](#) & [operator=](#) ([Parallax](#) &&other)=default
Assign a [Parallax](#) object by move.
- [~Parallax](#) ()=default
Destroy the [Parallax](#) object.
- void [move](#) ()
Move each layer of the parallax by a certain speed.
- void [load](#) ()
Load the parallax by setting the position of each layer.
- void [display](#) ([gui::Window](#) &window)
Display the parallax.
- void [reset](#) ()
Reset the parallax to make it loop.

Private Attributes

- `std::vector< std::unique_ptr< Background > > parallax`
- `sf::Clock clock`

6.38.1 Detailed Description

[Parallax](#) class.

6.38.2 Constructor & Destructor Documentation

6.38.2.1 [Parallax\(\)](#) [1/3]

```
gui::Parallax::Parallax ( )
```

Construct a new [Parallax](#) object.

Exceptions

Errors::ErrorTexture	
--------------------------------------	--

6.38.2.2 [Parallax\(\)](#) [2/3]

```
gui::Parallax::Parallax (
    const Parallax & other ) [delete]
```

Construct a new [Parallax](#) object by copy.

Parameters

<i>other</i>	
--------------	--

6.38.2.3 [Parallax\(\)](#) [3/3]

```
gui::Parallax::Parallax (
    Parallax && other ) [default]
```

Construct a new [Parallax](#) object by move.

Parameters

<i>other</i>	
--------------	--

6.38.2.4 ~Parallax()

```
gui::Parallax::~~Parallax ( ) [default]
```

Destroy the [Parallax](#) object.

6.38.3 Member Function Documentation

6.38.3.1 display()

```
void gui::Parallax::display (
    gui::Window & window )
```

Display the parallax.

Parameters

<i>window</i>	The window where the parallax will be displayed
---------------	---

6.38.3.2 load()

```
void gui::Parallax::load ( )
```

Load the parallax by setting the position of each layer.

6.38.3.3 move()

```
void gui::Parallax::move ( )
```

Move each layer of the parallax by a certain speed.

6.38.3.4 operator=() [1/2]

```
Parallax& gui::Parallax::operator= (
    const Parallax & other ) [delete]
```

Assign a [Parallax](#) object by copy.

Parameters

<i>other</i>	
--------------	--

Returns

[Parallax](#)&

6.38.3.5 operator=() [2/2]

```
Parallax& gui::Parallax::operator= (  
    Parallax && other ) [default]
```

Assign a [Parallax](#) object by move.

Parameters

<i>other</i>	
--------------	--

Returns

[Parallax](#)&

6.38.3.6 reset()

```
void gui::Parallax::reset ( )
```

Reset the parallax to make it loop.

6.38.4 Field Documentation**6.38.4.1 clock**

```
sf::Clock gui::Parallax::clock [private]
```

6.38.4.2 parallax

```
std::vector<std::unique_ptr<Background> > gui::Parallax::parallax [private]
```

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

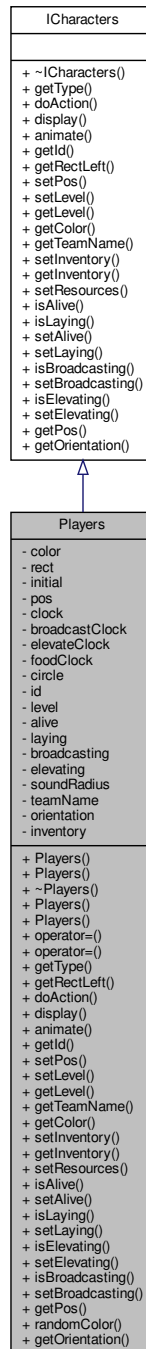
- Components/Parallax/[Parallax.hpp](#)
- Components/Parallax/[Parallax.cpp](#)

6.39 Players Class Reference

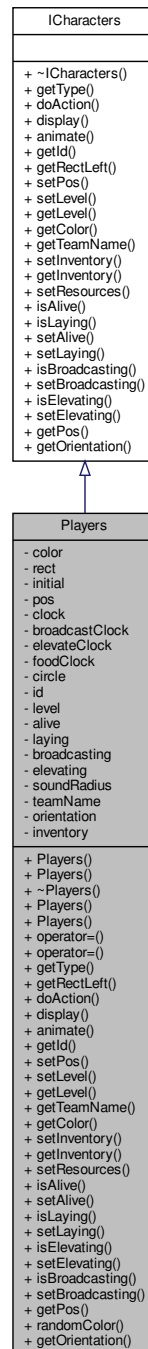
[Players](#) class.

```
#include <Players.hpp>
```

Inheritance diagram for Players:



Collaboration diagram for Players:



Public Member Functions

- `Players` (sf::IntRect `rect`, sf::Vector2f `pos`)
Construct a new `Players` object.
- `Players` (sf::Vector2f `pos`, int `id`, int `level`, const std::string &`teamName`, Orientation `orientation`, gui::Data &`data`, bool `broadcastingParams`)
Construct a new `Players` object.

- `~Players ()` noexcept override=default
Destroy the `Players` object.
- `Players (const Players &other)=delete`
Construct a new `Players` object by copy deleted.
- `Players (Players &&other)=default`
Construct a new `Players` object by move.
- `Players & operator= (const Players &other)=delete`
Assign a `Players` object by copy deleted.
- `Players & operator= (Players &&other)=default`
Assign a `Players` object by move.
- `CharacterType getType ()` const override
Get the Type of the Characters.
- `const int & getRectLeft ()` override
Get the Rect Left of the Player.
- `void doAction ()` override
Do the action of the Player.
- `void display (sf::RenderWindow &window, sf::Sprite &sprite, gui::ItemDrawer &drawer, int timeUnit, SoundBox &soundbox)` override
Display the Player.
- `void animate (sf::Sprite &sprite)` override
Animate the Player.
- `const int & getId ()` override
Get the Id of the Player.
- `void setPos (sf::Vector2f pos, Orientation orientation)` override
Set the Pos of the Player.
- `void setLevel (int level)` override
Set the Level of the Player.
- `const int & getLevel ()` override
Get the Level of the Player.
- `const std::string & getTeamName ()` override
Get the Team Name of the Player.
- `const std::string & getColor ()` override
Get the Color of the Player.
- `void setInventory (const std::array< int, 7 > &res)` override
Set the Inventory of the Player.
- `const std::array< int, 7 > & getInventory ()` override
Get the Inventory of the Player.
- `void setResources (const int &res, const int &nb)` override
Set the Resources of the Player.
- `const bool & isAlive ()` override
Check if the Player is Alive.
- `void setAlive (bool alive)` override
Set the Alive boolean of the Player.
- `const bool & isLaying ()` override
Check if the Player is Laying.
- `void setLaying (bool laying)` override
Set the Laying boolean of the Player.
- `const int & isElevating ()` override
Check if the Player is Elevating.
- `void setElevating (int elevating)` override
Set the Elevating process of the Player.

- const bool & [isBroadcasting](#) () override
Check if the Player is Broadcasting.
- void [setBroadcasting](#) (bool [broadcasting](#)) override
Set the Broadcasting boolean of the Player.
- const sf::Vector2f & [getPos](#) () override
Get the Pos of the Player.
- std::string [randomColor](#) (gui::Data &data)
Get a random color.
- const Orientation & [getOrientation](#) () override
Get the Orientation of the Player.

Private Attributes

- std::string [color](#)
- sf::IntRect [rect](#)
- sf::IntRect [initial](#)
- sf::Vector2f [pos](#)
- sf::Clock [clock](#)
- sf::Clock [broadcastClock](#)
- sf::Clock [elevateClock](#)
- sf::Clock [foodClock](#)
- sf::CircleShape [circle](#)
- int [id](#)
- int [level](#)
- bool [alive](#)
- bool [laying](#)
- bool [broadcasting](#)
- int [elevating](#)
- int [soundRadius](#)
- std::string [teamName](#)
- Orientation [orientation](#)
- std::array< int, 7 > [inventory](#) {0, 0, 0, 0, 0, 0, 0}

6.39.1 Detailed Description

[Players](#) class.

6.39.2 Constructor & Destructor Documentation

6.39.2.1 [Players\(\)](#) [1/4]

```
Players::Players (
    sf::IntRect rect,
    sf::Vector2f pos )
```

Construct a new [Players](#) object.

Parameters

<i>rect</i>	
<i>pos</i>	

6.39.2.2 Players() [2/4]

```

Players::Players (
    sf::Vector2f pos,
    int id,
    int level,
    const std::string & teamName,
    Orientation orientation,
    gui::Data & data,
    bool broadcastingParams )

```

Construct a new [Players](#) object.

Parameters

<i>pos</i>	
<i>id</i>	
<i>level</i>	
<i>teamName</i>	
<i>orientation</i>	
<i>data</i>	
<i>broadcastingParams</i>	

6.39.2.3 ~Players()

```

Players::~~Players ( ) [override], [default], [noexcept]

```

Destroy the [Players](#) object.

6.39.2.4 Players() [3/4]

```

Players::Players (
    const Players & other ) [delete]

```

Construct a new [Players](#) object by copy deleted.

Parameters

<i>other</i>	
--------------	--

6.39.2.5 Players() [4/4]

```
Players::Players (
    Players && other ) [default]
```

Construct a new [Players](#) object by move.

Parameters

<i>other</i>	
--------------	--

6.39.3 Member Function Documentation**6.39.3.1 animate()**

```
void Players::animate (
    sf::Sprite & sprite ) [override], [virtual]
```

Animate the Player.

Parameters

<i>sprite</i>	
---------------	--

Implements [ICharacters](#).

6.39.3.2 display()

```
void Players::display (
    sf::RenderWindow & window,
    sf::Sprite & sprite,
    gui::ItemDrawer & drawer,
    int timeUnit,
    SoundBox & soundbox ) [override], [virtual]
```

Display the Player.

Parameters

<i>window</i>	
<i>sprite</i>	
<i>drawer</i>	
<i>timeUnit</i>	
<i>soundbox</i>	

Implements [ICharacters](#).

6.39.3.3 doAction()

```
void Players::doAction ( ) [override], [virtual]
```

Do the action of the Player.

Implements [ICharacters](#).

6.39.3.4 getColor()

```
const std::string & Players::getColor ( ) [override], [virtual]
```

Get the Color of the Player.

Returns

const std::string&

Implements [ICharacters](#).

6.39.3.5 getId()

```
const int & Players::getId ( ) [override], [virtual]
```

Get the Id of the Player.

Returns

const int&

Implements [ICharacters](#).

6.39.3.6 getInventory()

```
const std::array< int, 7 > & Players::getInventory ( ) [override], [virtual]
```

Get the Inventory of the Player.

Returns

std::array<int, 7>&

Implements [ICharacters](#).

6.39.3.7 getLevel()

```
const int & Players::getLevel ( ) [override], [virtual]
```

Get the Level of the Player.

Returns

const int&

Implements [ICharacters](#).

6.39.3.8 getOrientation()

```
const Orientation & Players::getOrientation ( ) [override], [virtual]
```

Get the Orientation of the Player.

Returns

const Orientation&

Implements [ICharacters](#).

6.39.3.9 getPos()

```
const sf::Vector2f & Players::getPos ( ) [override], [virtual]
```

Get the Pos of the Player.

Returns

sf::Vector2f

Implements [ICharacters](#).

6.39.3.10 getRectLeft()

```
const int & Players::getRectLeft ( ) [override], [virtual]
```

Get the Rect Left of the Player.

Returns

const int&

Implements [ICharacters](#).

6.39.3.11 getTeamName()

```
const std::string & Players::getTeamName ( ) [override], [virtual]
```

Get the Team Name of the Player.

Returns

const std::string&

Implements [ICharacters](#).

6.39.3.12 getType()

```
CharacterType Players::getType ( ) const [override], [virtual]
```

Get the Type of the Characters.

Returns

CharacterType

Implements [ICharacters](#).

6.39.3.13 isAlive()

```
const bool & Players::isAlive ( ) [override], [virtual]
```

Check if the Player is Alive.

Returns

true

false

Implements [ICharacters](#).

6.39.3.14 isBroadcasting()

```
const bool & Players::isBroadcasting ( ) [override], [virtual]
```

Check if the Player is Broadcasting.

Returns

true
false

Implements [ICharacters](#).

6.39.3.15 isElevating()

```
const int & Players::isElevating ( ) [override], [virtual]
```

Check if the Player is Elevating.

Returns

int

Implements [ICharacters](#).

6.39.3.16 isLaying()

```
const bool & Players::isLaying ( ) [override], [virtual]
```

Check if the Player is Laying.

Returns

true
false

Implements [ICharacters](#).

6.39.3.17 operator=() [1/2]

```
Players& Players::operator= (
    const Players & other ) [delete]
```

Assign a [Players](#) object by copy deleted.

Parameters

<i>other</i>	
--------------	--

Returns

[Players](#)&

6.39.3.18 operator=() [2/2]

```
Players& Players::operator= (
    Players && other ) [default]
```

Assign a [Players](#) object by move.

Parameters

<i>other</i>	
--------------	--

Returns

[Players](#)&

6.39.3.19 randomColor()

```
std::string Players::randomColor (
    gui::Data & data )
```

Get a random color.

Parameters

<i>data</i>	
-------------	--

Returns

std::string

6.39.3.20 setAlive()

```
void Players::setAlive (
    bool alive ) [override], [virtual]
```

Set the Alive boolean of the Player.

Parameters

<i>alive</i>	
--------------	--

Implements [ICharacters](#).

6.39.3.21 setBroadcasting()

```
void Players::setBroadcasting (
    bool broadcasting ) [override], [virtual]
```

Set the Broadcasting boolean of the Player.

Parameters

<i>broadcasting</i>	
---------------------	--

Implements [ICharacters](#).

6.39.3.22 setElevating()

```
void Players::setElevating (
    int elevating ) [override], [virtual]
```

Set the Elevating process of the Player.

Parameters

<i>elevating</i>	
------------------	--

Implements [ICharacters](#).

6.39.3.23 setInventory()

```
void Players::setInventory (
    const std::array< int, 7 > & res ) [override], [virtual]
```

Set the Inventory of the Player.

Parameters

<i>res</i>	
------------	--

Implements [ICharacters](#).

6.39.3.24 setLaying()

```
void Players::setLaying (
    bool laying ) [override], [virtual]
```

Set the Laying boolean of the Player.

Parameters

<i>laying</i>	
---------------	--

Implements [ICharacters](#).

6.39.3.25 setLevel()

```
void Players::setLevel (
    int level ) [override], [virtual]
```

Set the Level of the Player.

Parameters

<i>level</i>	
--------------	--

Implements [ICharacters](#).

6.39.3.26 setPos()

```
void Players::setPos (
    sf::Vector2f pos,
    Orientation orientation ) [override], [virtual]
```

Set the Pos of the Player.

Parameters

<i>pos</i>	
<i>orientation</i>	

Implements [ICharacters](#).

6.39.3.27 setResources()

```
void Players::setResources (
    const int & res,
    const int & nb ) [override], [virtual]
```

Set the Resources of the Player.

Parameters

<i>res</i>	
<i>nb</i>	

Implements [ICharacters](#).

6.39.4 Field Documentation

6.39.4.1 alive

```
bool Players::alive [private]
```

6.39.4.2 broadcastClock

```
sf::Clock Players::broadcastClock [private]
```

6.39.4.3 broadcasting

```
bool Players::broadcasting [private]
```

6.39.4.4 circle

```
sf::CircleShape Players::circle [private]
```

6.39.4.5 clock

```
sf::Clock Players::clock [private]
```

6.39.4.6 color

```
std::string Players::color [private]
```

6.39.4.7 elevateClock

```
sf::Clock Players::elevateClock [private]
```

6.39.4.8 elevating

```
int Players::elevating [private]
```

6.39.4.9 foodClock

```
sf::Clock Players::foodClock [private]
```

6.39.4.10 id

```
int Players::id [private]
```

6.39.4.11 initial

```
sf::IntRect Players::initial [private]
```

6.39.4.12 inventory

```
std::array< int, 7 > Players::inventory {0, 0, 0, 0, 0, 0, 0} [private]
```

6.39.4.13 laying

```
bool Players::laying [private]
```

6.39.4.14 level

```
int Players::level [private]
```

6.39.4.15 orientation

```
Orientation Players::orientation [private]
```

6.39.4.16 pos

```
sf::Vector2f Players::pos [private]
```

6.39.4.17 rect

```
sf::IntRect Players::rect [private]
```

6.39.4.18 soundRadius

```
int Players::soundRadius [private]
```

6.39.4.19 teamName

```
std::string Players::teamName [private]
```

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

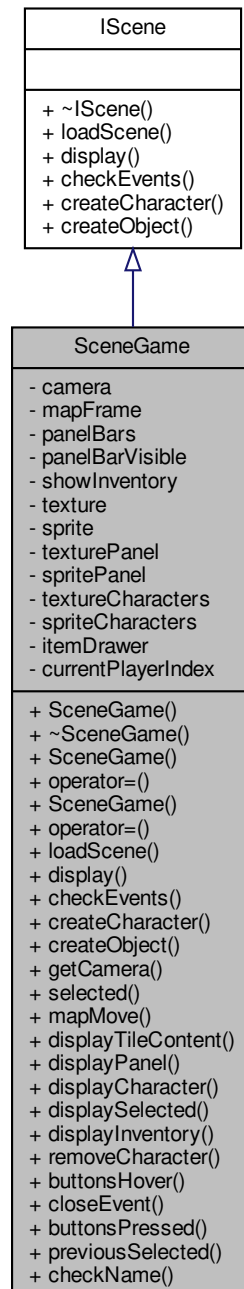
- Characters/Players/[Players.hpp](#)
- Characters/Players/[Players.cpp](#)

6.40 SceneGame Class Reference

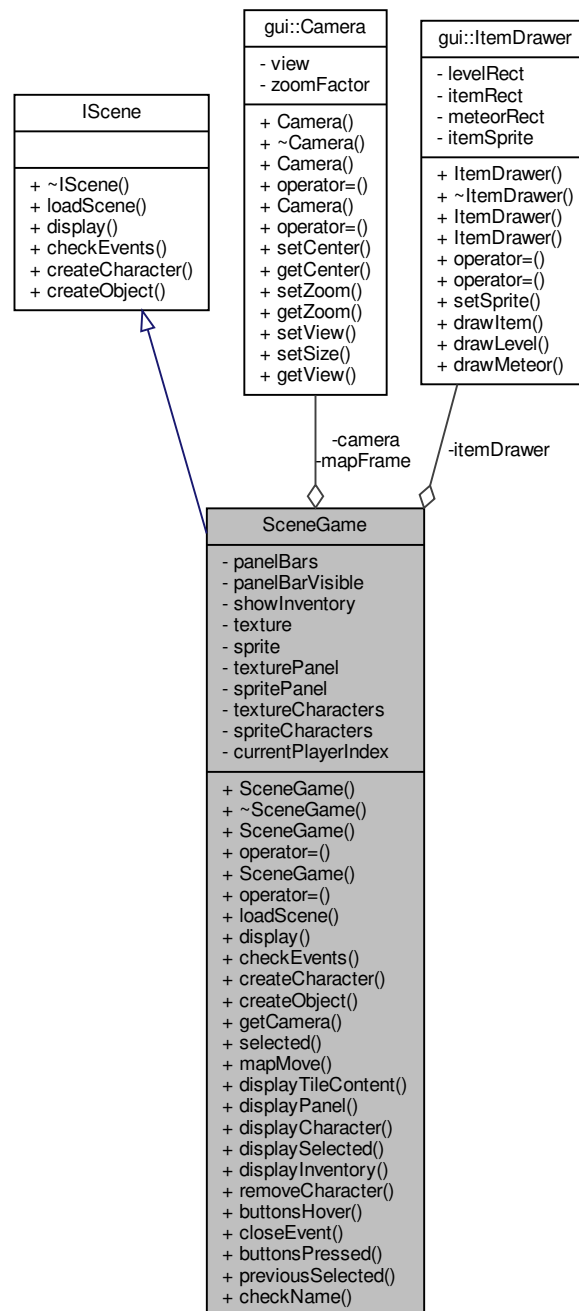
Game scene class.

```
#include <SceneGame.hpp>
```

Inheritance diagram for SceneGame:



Collaboration diagram for SceneGame:



Public Member Functions

- **SceneGame** ()=default
Construct a new Scene Game object.
- **~SceneGame** () noexcept override=default
Destroy the Scene Game object.
- **SceneGame** (const **SceneGame** &other)=delete

- Construct a new Scene Game object by copy.*

 - `SceneGame & operator= (const SceneGame &other)=delete`

Assign a Scene Game object by copy.
- `SceneGame (SceneGame &&other)=default`

Construct a new Scene Game object by move.
- `SceneGame & operator= (SceneGame &&other)=default`

Assign a Scene Game object by move.
- void `loadScene (gui::SceneManager &manager, gui::Window &window, gui::Data &data)` override

Load the scene.
- void `display (gui::SceneManager &manager, gui::Window &window, gui::Data &data)` override

Display the scene.
- void `checkEvents (gui::SceneManager &manager, gui::Window &window, gui::Data &data)` override

Check the events of the scene.
- void `createCharacter ()` override

Create the character of the scene.
- void `createObject (gui::Data &data)` override

Create the objects of the scene.
- const `gui::Camera & getCamera ()`

Get the camera of the scene.
- void `selected (std::vector< std::vector< gui::Tile >> &tiles, sf::Vector2f pos)`

Set the outline of the selected tile.
- void `mapMove (gui::Window &window, gui::Data &data)`

Move the map.
- void `displayTileContent (std::vector< std::vector< gui::Tile >> &map, gui::Window &window, gui::ItemDrawer &itemDrawer, gui::Data &data)`

Display tile of the Map.
- void `displayPanel (gui::SceneManager &manager, gui::Window &window, gui::Data &data)`

Display the panel bar.
- void `displayCharacter (gui::Window &window, gui::Data &data, gui::SceneManager &manager)`

Display the Character.
- void `displaySelected (gui::Window &window, gui::Data &data, gui::ItemDrawer &itemDrawer, gui::Tile &tile)`

Display the selected tile.
- void `displayInventory (gui::Data &data, gui::Tile &tile)`

Display the inventory.
- void `removeCharacter (gui::SceneManager &manager, gui::Data &data)`

Remove the character.
- void `buttonsHover (gui::SceneManager &manager, gui::Window &window)`

Hover state and the buttons.
- void `closeEvent (gui::Window &window, gui::Data &data)`

Handle close event.
- void `buttonsPressed (gui::SceneManager &manager, gui::Window &window, gui::Data &data)`

Handle buttons pressed.
- void `previousSelected (std::vector< std::vector< gui::Tile >> &map)`

Handle buttons released.
- void `checkName (const std::unique_ptr< gui::Button > &button, gui::Data &data)`

Check if the name is valid.

Private Attributes

- `gui::Camera camera {0, 0}`
- `gui::Camera mapFrame {0, 0}`
- `std::vector< gui::PanelBar > panelBars`
- `bool panelBarVisible {false}`
- `bool showInventory {false}`
- `sf::Texture texture`
- `sf::Sprite sprite`
- `sf::Texture texturePanel`
- `sf::Sprite spritePanel`
- `sf::Texture textureCharacters`
- `sf::Sprite spriteCharacters`
- `gui::ItemDrawer itemDrawer`
- `int currentPlayerIndex`

6.40.1 Detailed Description

Game scene class.

6.40.2 Constructor & Destructor Documentation

6.40.2.1 SceneGame() [1/3]

```
SceneGame::SceneGame ( ) [default]
```

Construct a new Scene Game object.

6.40.2.2 ~SceneGame()

```
SceneGame::~~SceneGame ( ) [override], [default], [noexcept]
```

Destroy the Scene Game object.

6.40.2.3 SceneGame() [2/3]

```
SceneGame::SceneGame (
    const SceneGame & other ) [delete]
```

Construct a new Scene Game object by copy.

Parameters

<i>other</i>	
--------------	--

6.40.2.4 SceneGame() [3/3]

```
SceneGame::SceneGame (  
    SceneGame && other ) [default]
```

Construct a new Scene Game object by move.

Parameters

<i>other</i>	
--------------	--

6.40.3 Member Function Documentation**6.40.3.1 buttonsHover()**

```
void SceneGame::buttonsHover (  
    gui::SceneManager & manager,  
    gui::Window & window )
```

Hover state and the buttons.

Parameters

<i>manager</i>	
<i>window</i>	

6.40.3.2 buttonsPressed()

```
void SceneGame::buttonsPressed (  
    gui::SceneManager & manager,  
    gui::Window & window,  
    gui::Data & data )
```

Handle buttons pressed.

Parameters

<i>manager</i>	
<i>window</i>	
<i>data</i>	

6.40.3.3 checkEvents()

```
void SceneGame::checkEvents (
    gui::SceneManager & manager,
    gui::Window & window,
    gui::Data & data ) [override], [virtual]
```

Check the events of the scene.

Parameters

<i>window</i>	The window where the events will be checked
<i>data</i>	The data of the game
<i>manager</i>	The scene manager

Implements [IScene](#).

6.40.3.4 checkName()

```
void SceneGame::checkName (
    const std::unique_ptr< gui::Button > & button,
    gui::Data & data )
```

Check if the name is valid.

Parameters

<i>button</i>	
<i>data</i>	

6.40.3.5 closeEvent()

```
void SceneGame::closeEvent (
    gui::Window & window,
    gui::Data & data )
```

Handle close event.

Parameters

<i>window</i>	
<i>data</i>	

6.40.3.6 createCharacter()

```
void SceneGame::createCharacter ( ) [inline], [override], [virtual]
```

Create the character of the scene.

Implements [IScene](#).

6.40.3.7 createObject()

```
void SceneGame::createObject (
    gui::Data & data ) [override], [virtual]
```

Create the objects of the scene.

Parameters

<i>data</i>	The data of the game
-------------	----------------------

Exceptions

Errors::ErrorTexture	
--------------------------------------	--

Implements [IScene](#).

6.40.3.8 display()

```
void SceneGame::display (
    gui::SceneManager & manager,
    gui::Window & window,
    gui::Data & data ) [override], [virtual]
```

Display the scene.

Parameters

<i>window</i>	The window where the scene will be displayed
<i>data</i>	The data of the game
<i>manager</i>	The scene manager

Implements [IScene](#).

6.40.3.9 displayCharacter()

```
void SceneGame::displayCharacter (
    gui::Window & window,
    gui::Data & data,
    gui::SceneManager & manager )
```

Display the Character.

Parameters

<i>window</i>	
<i>data</i>	
<i>manager</i>	

6.40.3.10 displayInventory()

```
void SceneGame::displayInventory (
    gui::Data & data,
    gui::Tile & tile )
```

Display the inventory.

Parameters

<i>data</i>	
<i>tile</i>	

6.40.3.11 displayPanel()

```
void SceneGame::displayPanel (
    gui::SceneManager & manager,
    gui::Window & window,
    gui::Data & data )
```

Display the panel bar.

Parameters

<i>manager</i>	
<i>window</i>	
<i>data</i>	

6.40.3.12 displaySelected()

```
void SceneGame::displaySelected (
    gui::Window & window,
    gui::Data & data,
    gui::ItemDrawer & itemDrawer,
    gui::Tile & tile )
```

Display the selected tile.

Parameters

<i>window</i>	
<i>data</i>	
<i>itemDrawer</i>	
<i>tile</i>	

6.40.3.13 displayTileContent()

```
void SceneGame::displayTileContent (
    std::vector< std::vector< gui::Tile >> & map,
    gui::Window & window,
    gui::ItemDrawer & itemDrawer,
    gui::Data & data )
```

Display tile of the Map.

Parameters

<i>map</i>	
<i>window</i>	
<i>itemDrawer</i>	
<i>data</i>	

6.40.3.14 getCamera()

```
const gui::Camera & SceneGame::getCamera ( )
```

Get the camera of the scene.

Returns

The camera of the scene

6.40.3.15 loadScene()

```
void SceneGame::loadScene (
    gui::SceneManager & manager,
    gui::Window & window,
    gui::Data & data ) [override], [virtual]
```

Load the scene.

Parameters

<i>window</i>	The window where the scene will be loaded
<i>data</i>	The data of the game
<i>manager</i>	The scene manager

Implements [IScene](#).

6.40.3.16 mapMove()

```
void SceneGame::mapMove (
    gui::Window & window,
    gui::Data & data )
```

Move the map.

Parameters

<i>window</i>	The window where the map will be moved
<i>data</i>	The data of the game

6.40.3.17 operator=() [1/2]

```
SceneGame& SceneGame::operator= (
    const SceneGame & other ) [delete]
```

Assign a Scene Game object by copy.

Parameters

<i>other</i>	
--------------	--

Returns

[SceneGame&](#)

6.40.3.18 operator=() [2/2]

```
SceneGame& SceneGame::operator= (
    SceneGame && other ) [default]
```

Assign a Scene Game object by move.

Parameters

<i>other</i>	
--------------	--

Returns

SceneGame&

6.40.3.19 previousSelected()

```
void SceneGame::previousSelected (
    std::vector< std::vector< gui::Tile >> & map )
```

Handle buttons released.

Parameters

<i>map</i>	
------------	--

6.40.3.20 removeCharacter()

```
void SceneGame::removeCharacter (
    gui::SceneManager & manager,
    gui::Data & data )
```

Remove the character.

Parameters

<i>manager</i>	
<i>data</i>	

6.40.3.21 selected()

```
void SceneGame::selected (
```

```
std::vector< std::vector< gui::Tile >> & tiles,  
sf::Vector2f pos )
```

Set the outline of the selected tile.

Parameters

<i>tiles</i>	The tiles of the map
<i>pos</i>	The position of the mouse

6.40.4 Field Documentation

6.40.4.1 camera

```
gui::Camera SceneGame::camera {0, 0} [private]
```

6.40.4.2 currentPlayerIndex

```
int SceneGame::currentPlayerIndex [private]
```

6.40.4.3 itemDrawer

```
gui::ItemDrawer SceneGame::itemDrawer [private]
```

6.40.4.4 mapFrame

```
gui::Camera SceneGame::mapFrame {0, 0} [private]
```

6.40.4.5 panelBars

```
std::vector< gui::PanelBar > SceneGame::panelBars [private]
```

6.40.4.6 panelBarVisible

```
bool SceneGame::panelBarVisible {false} [private]
```

6.40.4.7 showInventory

```
bool SceneGame::showInventory {false} [private]
```

6.40.4.8 sprite

```
sf::Sprite SceneGame::sprite [private]
```

6.40.4.9 spriteCharacters

```
sf::Sprite SceneGame::spriteCharacters [private]
```

6.40.4.10 spritePanel

```
sf::Sprite SceneGame::spritePanel [private]
```

6.40.4.11 texture

```
sf::Texture SceneGame::texture [private]
```

6.40.4.12 textureCharacters

```
sf::Texture SceneGame::textureCharacters [private]
```

6.40.4.13 texturePanel

```
sf::Texture SceneGame::texturePanel [private]
```

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

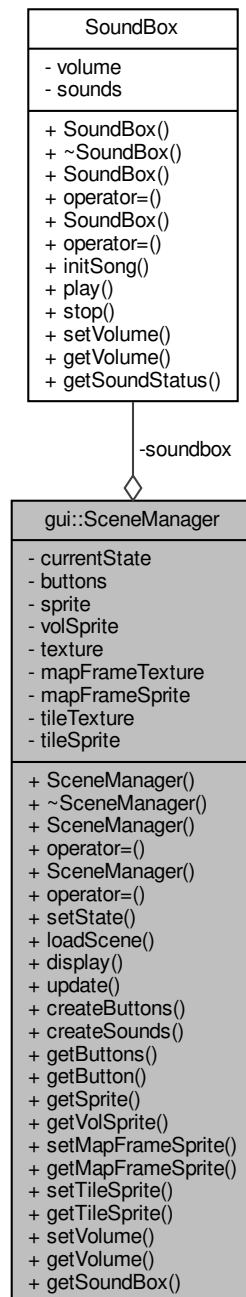
- Scene/Game/[SceneGame.hpp](#)
- Scene/Game/Display/[Display.cpp](#)
- Scene/Game/Display/[DisplayTile.cpp](#)
- Scene/Game/Event/[Event.cpp](#)
- Scene/Game/Event/[MapMove.cpp](#)
- Scene/Game/Event/[Selected.cpp](#)
- Scene/Game/[SceneGame.cpp](#)

6.41 gui::SceneManager Class Reference

[SceneManager](#) class.

```
#include <SceneManager.hpp>
```

Collaboration diagram for gui::SceneManager:



Public Member Functions

- [SceneManager](#) ()
Construct a new Scene Manager object.
- [~SceneManager](#) () noexcept=default
Destroy the Scene Manager object.
- [SceneManager](#) (const [SceneManager](#) &other)=delete

- Construct a new Scene Manager object by copy.*

 - `SceneManager & operator= (const SceneManager &other)=delete`

Assign a Scene Manager object by copy.

 - `SceneManager (SceneManager &&other)=default`

Construct a new Scene Manager object by move.

 - `SceneManager & operator= (SceneManager &&other)=default`

Assign a Scene Manager object by move.

 - `void setState (std::unique_ptr< IScene > state)`

Set the state of the scene.

 - `void loadScene (gui::SceneManager &manager, gui::Window &window, gui::Data &data)`
 - `void display (gui::Window &window, gui::Data &data)`
 - `void update (gui::Window &window, gui::Data &data)`

Check the events of the scene.

 - `void createButtons ()`

Create a Buttons object.

 - `void createSounds ()`

Create a Sounds object.

 - `std::vector< std::unique_ptr< gui::Button > > & getButtons ()`

Get the Buttons object.

 - `gui::Button & getButton (std::string name)`

Get the Button.

 - `sf::Sprite & getSprite ()`

Get the Sprite.

 - `sf::Sprite & getVolSprite ()`

Get the Vol Sprite.

 - `void setMapFrameSprite (int rect)`

Set the Map Frame Sprite.

 - `sf::Sprite & getMapFrameSprite ()`

Get the Map Frame Sprite.

 - `void setTileSprite (int rect)`

Set the Tile Sprite.

 - `const sf::Sprite & getTileSprite ()`

Get the Tile Sprite.

 - `void setVolume (float volume)`

Set the Volume.

 - `int getVolume ()`

Get the Volume.

 - `SoundBox & getSoundBox ()`

Get the Sound Box.

Private Attributes

- `std::unique_ptr< IScene > currentState`
- `std::vector< std::unique_ptr< gui::Button > > buttons`
- `sf::Sprite sprite`
- `sf::Sprite volSprite`
- `sf::Texture texture`
- `sf::Texture mapFrameTexture`
- `sf::Sprite mapFrameSprite`
- `sf::Texture tileTexture`
- `sf::Sprite tileSprite`
- `SoundBox soundbox`

6.41.1 Detailed Description

[SceneManager](#) class.

6.41.2 Constructor & Destructor Documentation

6.41.2.1 SceneManager() [1/3]

```
gui::SceneManager::SceneManager ( )
```

Construct a new Scene Manager object.

Exceptions

Errors::ErrorTexture	
--------------------------------------	--

6.41.2.2 ~SceneManager()

```
gui::SceneManager::~SceneManager ( ) [default], [noexcept]
```

Destroy the Scene Manager object.

6.41.2.3 SceneManager() [2/3]

```
gui::SceneManager::SceneManager (
    const SceneManager & other ) [delete]
```

Construct a new Scene Manager object by copy.

Parameters

<i>other</i>	
--------------	--

6.41.2.4 SceneManager() [3/3]

```
gui::SceneManager::SceneManager (
    SceneManager && other ) [default]
```

Construct a new Scene Manager object by move.

Parameters

<i>other</i>	
--------------	--

6.41.3 Member Function Documentation

6.41.3.1 createButtons()

```
void gui::SceneManager::createButtons ( )
```

Create a Buttons object.

6.41.3.2 createSounds()

```
void gui::SceneManager::createSounds ( )
```

Create a Sounds object.

6.41.3.3 display()

```
void gui::SceneManager::display (
    gui::Window & window,
    gui::Data & data )
```

6.41.3.4 getButton()

```
gui::Button & gui::SceneManager::getButton (
    std::string name )
```

Get the [Button](#).

Parameters

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

Returns

[gui::Button](#)&

6.41.3.5 getButtons()

```
std::vector< std::unique_ptr< gui::Button > > & gui::SceneManager::getButtons ( )
```

Get the Buttons object.

Returns

std::vector<std::unique_ptr<gui::Button>>&

6.41.3.6 getMapFrameSprite()

```
sf::Sprite & gui::SceneManager::getMapFrameSprite ( )
```

Get the [Map](#) Frame Sprite.

Returns

sf::Sprite&

6.41.3.7 getSoundBox()

```
SoundBox & gui::SceneManager::getSoundBox ( )
```

Get the Sound Box.

Returns

[SoundBox](#)&

6.41.3.8 getSprite()

```
sf::Sprite & gui::SceneManager::getSprite ( )
```

Get the Sprite.

Returns

sf::Sprite&

6.41.3.9 getTileSprite()

```
const sf::Sprite & gui::SceneManager::getTileSprite ( )
```

Get the [Tile](#) Sprite.

Returns

sf::Sprite&

6.41.3.10 getVolSprite()

```
sf::Sprite & gui::SceneManager::getVolSprite ( )
```

Get the Vol Sprite.

Returns

sf::Sprite&

6.41.3.11 getVolume()

```
int gui::SceneManager::getVolume ( )
```

Get the Volume.

Returns

const int&

6.41.3.12 loadScene()

```
void gui::SceneManager::loadScene (
    gui::SceneManager & manager,
    gui::Window & window,
    gui::Data & data )
```

6.41.3.13 operator=() [1/2]

```
SceneManager& gui::SceneManager::operator= (
    const SceneManager & other ) [delete]
```

Assign a Scene Manager object by copy.

Parameters

<i>other</i>	
--------------	--

Returns

[SceneManager](#)&

6.41.3.14 operator=() [2/2]

```
SceneManager& gui::SceneManager::operator= (  
    SceneManager && other ) [default]
```

Assign a Scene Manager object by move.

Parameters

<i>other</i>	
--------------	--

Returns

[SceneManager](#)&

6.41.3.15 setMapFrameSprite()

```
void gui::SceneManager::setMapFrameSprite (  
    int rect )
```

Set the [Map](#) Frame Sprite.

Parameters

<i>rect</i>	
-------------	--

6.41.3.16 setState()

```
void gui::SceneManager::setState (  
    std::unique_ptr< IScene > state )
```

Set the state of the scene.

Parameters

<i>state</i>	The state of the scene
--------------	------------------------

6.41.3.17 setTileSprite()

```
void gui::SceneManager::setTileSprite (
    int rect )
```

Set the [Tile](#) Sprite.

Parameters

<i>rect</i>	
-------------	--

6.41.3.18 setVolume()

```
void gui::SceneManager::setVolume (
    float volume )
```

Set the Volume.

Parameters

<i>volume</i>	
---------------	--

6.41.3.19 update()

```
void gui::SceneManager::update (
    gui::Window & window,
    gui::Data & data )
```

Check the events of the scene.

Parameters

<i>window</i>	The window where the events will be checked
<i>data</i>	The data of the game

6.41.4 Field Documentation

6.41.4.1 buttons

```
std::vector<std::unique_ptr<gui::Button> > gui::SceneManager::buttons [private]
```

6.41.4.2 currentState

```
std::unique_ptr< IScene > gui::SceneManager::currentState [private]
```

6.41.4.3 mapFrameSprite

```
sf::Sprite gui::SceneManager::mapFrameSprite [private]
```

6.41.4.4 mapFrameTexture

```
sf::Texture gui::SceneManager::mapFrameTexture [private]
```

6.41.4.5 soundbox

```
SoundBox gui::SceneManager::soundbox [private]
```

6.41.4.6 sprite

```
sf::Sprite gui::SceneManager::sprite [private]
```

6.41.4.7 texture

```
sf::Texture gui::SceneManager::texture [private]
```

6.41.4.8 tileSprite

```
sf::Sprite gui::SceneManager::tileSprite [private]
```

6.41.4.9 tileTexture

```
sf::Texture gui::SceneManager::tileTexture [private]
```

6.41.4.10 volSprite

```
sf::Sprite gui::SceneManager::volSprite [private]
```

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

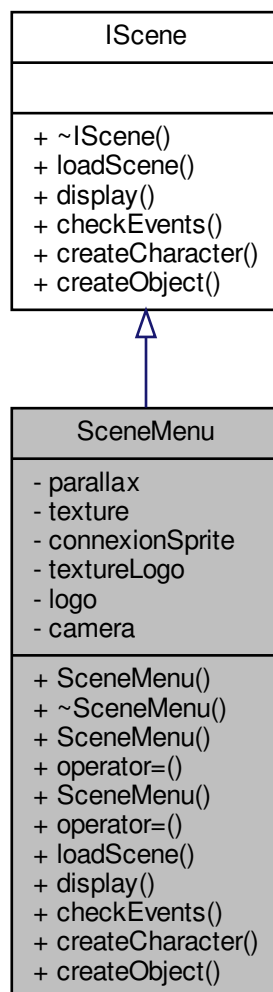
- Scene/Manager/[SceneManager.hpp](#)
- Scene/Manager/[SceneManager.cpp](#)

6.42 SceneMenu Class Reference

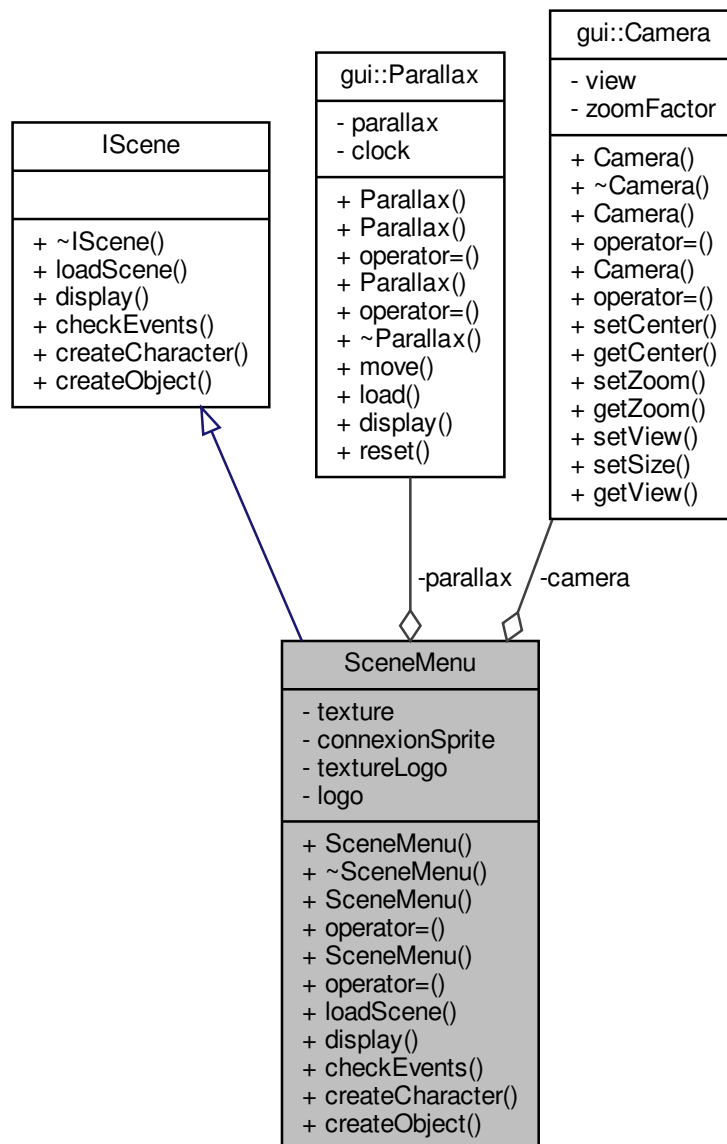
[SceneMenu](#) class.

```
#include <SceneMenu.hpp>
```


Inheritance diagram for SceneMenu:



Collaboration diagram for SceneMenu:



Public Member Functions

- [SceneMenu](#) ()=default
Construct a new Scene Menu object.
- [~SceneMenu](#) () noexcept override=default
Destroy the Scene Menu object.
- [SceneMenu](#) (const [SceneMenu](#) &other)=delete
Construct a new Scene Menu object by copy.
- [SceneMenu](#) & [operator=](#) (const [SceneMenu](#) &other)=delete
Assign a Scene Menu object by copy.

- [SceneMenu](#) ([SceneMenu](#) &&other)=default
Construct a new Scene Menu object by move.
- [SceneMenu](#) & [operator=](#) ([SceneMenu](#) &&other)=default
Assign a Scene Menu object by move.
- void [loadScene](#) ([gui::SceneManager](#) &manager, [gui::Window](#) &>window, [gui::Data](#) &data) override
Load the scene.
- void [display](#) ([gui::SceneManager](#) &manager, [gui::Window](#) &>window, [gui::Data](#) &data) override
Display the scene.
- void [checkEvents](#) ([gui::SceneManager](#) &manager, [gui::Window](#) &>window, [gui::Data](#) &data) override
Check the events of the scene.
- void [createCharacter](#) () override
Create the character of the scene.
- void [createObject](#) (__attribute__((unused)) [gui::Data](#) &data) override
Create the objects of the scene.

Private Attributes

- [gui::Parallax](#) [parallax](#)
- [sf::Texture](#) [texture](#)
- [sf::Sprite](#) [connexionSprite](#)
- [sf::Texture](#) [textureLogo](#)
- [sf::Sprite](#) [logo](#)
- [gui::Camera](#) [camera](#) {0, 0}

6.42.1 Detailed Description

[SceneMenu](#) class.

6.42.2 Constructor & Destructor Documentation

6.42.2.1 [SceneMenu\(\)](#) [1/3]

```
SceneMenu::SceneMenu ( ) [default]
```

Construct a new Scene Menu object.

6.42.2.2 [~SceneMenu\(\)](#)

```
SceneMenu::~SceneMenu ( ) [override], [default], [noexcept]
```

Destroy the Scene Menu object.

6.42.2.3 [SceneMenu\(\)](#) [2/3]

```
SceneMenu::SceneMenu (
    const SceneMenu & other ) [delete]
```

Construct a new Scene Menu object by copy.

Parameters

<i>other</i>	
--------------	--

6.42.2.4 SceneMenu() [3/3]

```
SceneMenu::SceneMenu (  
    SceneMenu && other ) [default]
```

Construct a new Scene Menu object by move.

Parameters

<i>other</i>	
--------------	--

6.42.3 Member Function Documentation**6.42.3.1 checkEvents()**

```
void SceneMenu::checkEvents (  
    gui::SceneManager & manager,  
    gui::Window & window,  
    gui::Data & data ) [override], [virtual]
```

Check the events of the scene.

Parameters

<i>window</i>	The window where the events will be checked
<i>data</i>	The data of the game
<i>manager</i>	The scene manager

Implements [IScene](#).

6.42.3.2 createCharacter()

```
void SceneMenu::createCharacter ( ) [inline], [override], [virtual]
```

Create the character of the scene.

Implements [IScene](#).

6.42.3.3 createObject()

```
void SceneMenu::createObject (
    __attribute__((unused)) gui::Data & data ) [inline], [override]
```

Create the objects of the scene.

Parameters

<i>data</i>	The data of the game
-------------	----------------------

6.42.3.4 display()

```
void SceneMenu::display (
    gui::SceneManager & manager,
    gui::Window & window,
    gui::Data & data ) [override], [virtual]
```

Display the scene.

Parameters

<i>window</i>	The window where the scene will be displayed
<i>data</i>	The data of the game
<i>manager</i>	The scene manager

Implements [IScene](#).

6.42.3.5 loadScene()

```
void SceneMenu::loadScene (
    gui::SceneManager & manager,
    gui::Window & window,
    gui::Data & data ) [override], [virtual]
```

Load the scene.

Parameters

<i>window</i>	The window where the scene will be loaded
<i>data</i>	The data of the game
<i>manager</i>	The scene manager

Exceptions

<i>Error::ErrorTexture</i>	
----------------------------	--

Implements [IScene](#).

6.42.3.6 operator=() [1/2]

```
SceneMenu& SceneMenu::operator= (
    const SceneMenu & other ) [delete]
```

Assign a Scene Menu object by copy.

Parameters

<i>other</i>	
--------------	--

Returns

[SceneMenu&](#)

6.42.3.7 operator=() [2/2]

```
SceneMenu& SceneMenu::operator= (
    SceneMenu && other ) [default]
```

Assign a Scene Menu object by move.

Parameters

<i>other</i>	
--------------	--

Returns

[SceneMenu&](#)

6.42.4 Field Documentation**6.42.4.1 camera**

```
gui::Camera SceneMenu::camera {0, 0} [private]
```

6.42.4.2 connexionSprite

```
sf::Sprite SceneMenu::connexionSprite [private]
```

6.42.4.3 logo

```
sf::Sprite SceneMenu::logo [private]
```

6.42.4.4 parallax

```
gui::Parallax SceneMenu::parallax [private]
```

6.42.4.5 texture

```
sf::Texture SceneMenu::texture [private]
```

6.42.4.6 textureLogo

```
sf::Texture SceneMenu::textureLogo [private]
```

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

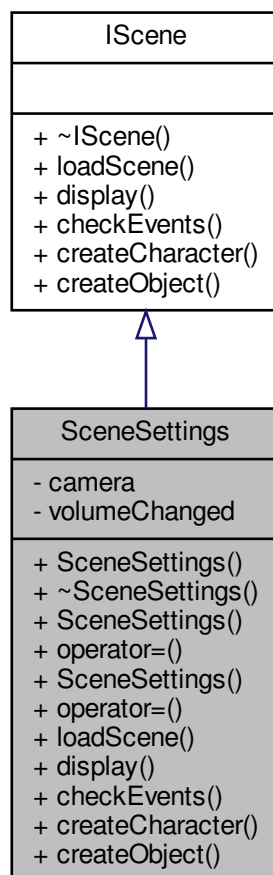
- Scene/Menu/[SceneMenu.hpp](#)
- Scene/Menu/[SceneMenu.cpp](#)

6.43 SceneSettings Class Reference

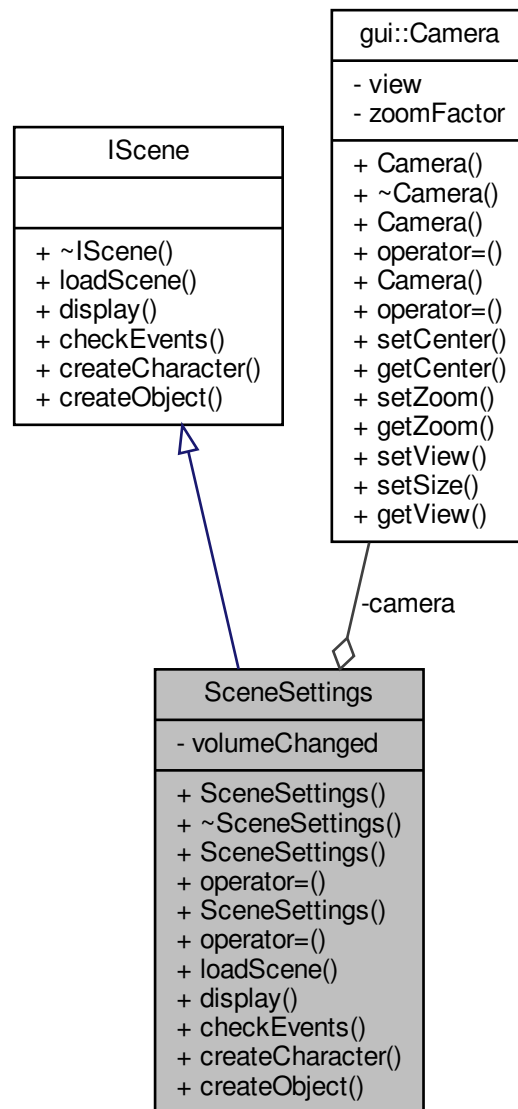
Settings scene class.

```
#include <SceneSettings.hpp>
```

Inheritance diagram for SceneSettings:



Collaboration diagram for SceneSettings:



Public Member Functions

- [SceneSettings](#) ()=default
Construct a new Scene Settings object.
- [~SceneSettings](#) () noexcept override=default
Destroy the Scene Settings object.
- [SceneSettings](#) (const [SceneSettings](#) &other)=delete
Construct a new Scene Settings object by copy.
- [SceneSettings](#) & [operator=](#) (const [SceneSettings](#) &other)=delete
Assign a Scene Settings object by copy.

- [SceneSettings](#) ([SceneSettings](#) &&other)=default
Construct a new Scene Settings object by move.
- [SceneSettings](#) & operator= ([SceneSettings](#) &&other)=default
Assign a Scene Settings object by move.
- void [loadScene](#) ([gui::SceneManager](#) &manager, [gui::Window](#) &>window, [gui::Data](#) &data) override
Load the scene.
- void [display](#) ([gui::SceneManager](#) &manager, [gui::Window](#) &>window, [gui::Data](#) &data) override
Display the scene.
- void [checkEvents](#) ([gui::SceneManager](#) &manager, [gui::Window](#) &>window, [gui::Data](#) &data) override
Check the events of the scene.
- void [createCharacter](#) () override
Create the character of the scene.
- void [createObject](#) (__attribute__((unused)) [gui::Data](#) &data) override
Create the objects of the scene.

Private Attributes

- [gui::Camera](#) camera {0, 0}
- bool [volumeChanged](#) {false}

6.43.1 Detailed Description

Settings scene class.

6.43.2 Constructor & Destructor Documentation

6.43.2.1 [SceneSettings\(\)](#) [1/3]

```
SceneSettings::SceneSettings ( ) [default]
```

Construct a new Scene Settings object.

6.43.2.2 [~SceneSettings\(\)](#)

```
SceneSettings::~~SceneSettings ( ) [override], [default], [noexcept]
```

Destroy the Scene Settings object.

6.43.2.3 [SceneSettings\(\)](#) [2/3]

```
SceneSettings::SceneSettings (
    const SceneSettings & other ) [delete]
```

Construct a new Scene Settings object by copy.

Parameters

<i>other</i>	
--------------	--

6.43.2.4 SceneSettings() [3/3]

```
SceneSettings::SceneSettings (
    SceneSettings && other ) [default]
```

Construct a new Scene Settings object by move.

Parameters

<i>other</i>	
--------------	--

6.43.3 Member Function Documentation**6.43.3.1 checkEvents()**

```
void SceneSettings::checkEvents (
    gui::SceneManager & manager,
    gui::Window & window,
    gui::Data & data ) [override], [virtual]
```

Check the events of the scene.

Parameters

<i>window</i>	The window where the events will be checked
<i>data</i>	The data of the game
<i>manager</i>	The scene manager

Implements [IScene](#).

6.43.3.2 createCharacter()

```
void SceneSettings::createCharacter ( ) [inline], [override], [virtual]
```

Create the character of the scene.

Implements [IScene](#).

6.43.3.3 createObject()

```
void SceneSettings::createObject (
    __attribute__((unused)) gui::Data & data ) [inline], [override]
```

Create the objects of the scene.

Parameters

<i>data</i>	The data of the game
-------------	----------------------

6.43.3.4 display()

```
void SceneSettings::display (
    gui::SceneManager & manager,
    gui::Window & window,
    gui::Data & data ) [override], [virtual]
```

Display the scene.

Parameters

<i>window</i>	The window where the scene will be displayed
<i>data</i>	The data of the game
<i>manager</i>	The scene manager

Implements [IScene](#).

6.43.3.5 loadScene()

```
void SceneSettings::loadScene (
    gui::SceneManager & manager,
    gui::Window & window,
    gui::Data & data ) [override], [virtual]
```

Load the scene.

Parameters

<i>window</i>	The window where the scene will be loaded
<i>data</i>	The data of the game
<i>manager</i>	The scene manager

Implements [IScene](#).

6.43.3.6 operator=() [1/2]

```
SceneSettings& SceneSettings::operator= (
    const SceneSettings & other ) [delete]
```

Assign a Scene Settings object by copy.

Parameters

<i>other</i>	
--------------	--

Returns

[SceneSettings&](#)

6.43.3.7 operator=() [2/2]

```
SceneSettings& SceneSettings::operator= (
    SceneSettings && other ) [default]
```

Assign a Scene Settings object by move.

Parameters

<i>other</i>	
--------------	--

Returns

[SceneSettings&](#)

6.43.4 Field Documentation

6.43.4.1 camera

```
gui::Camera SceneSettings::camera {0, 0} [private]
```

6.43.4.2 volumeChanged

```
bool SceneSettings::volumeChanged {false} [private]
```

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

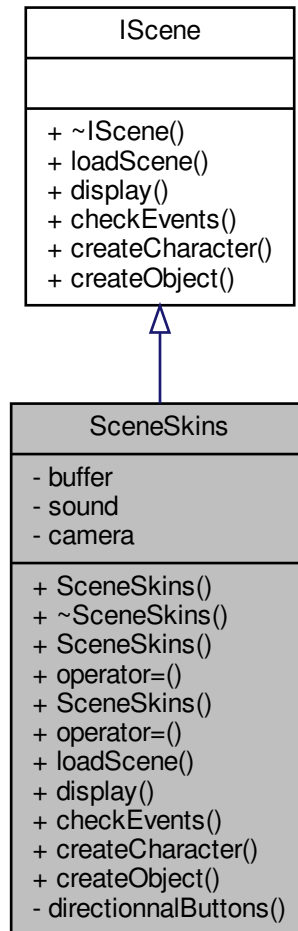
- Scene/Settings/[SceneSettings.hpp](#)
- Scene/Settings/[SceneSettings.cpp](#)

6.44 SceneSkins Class Reference

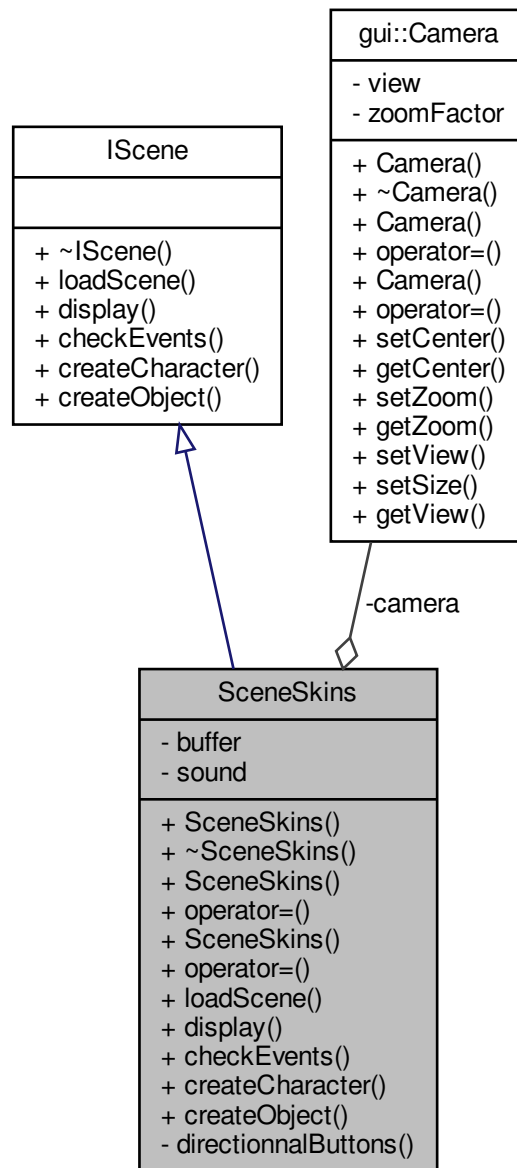
Skins scene class.

```
#include <SceneSkins.hpp>
```

Inheritance diagram for SceneSkins:



Collaboration diagram for SceneSkins:



Public Member Functions

- [SceneSkins](#) ()=default
Construct a new Scene Skins object.
- [~SceneSkins](#) () noexcept override=default
Destroy the Scene Skins object.
- [SceneSkins](#) (const [SceneSkins](#) &other)=delete
Construct a new Scene Skins.
- [SceneSkins](#) & [operator=](#) (const [SceneSkins](#) &other)=delete

- Assign a Scene Skins object by copy.*
 - `SceneSkins (SceneSkins &&other)=default`
- Construct a new Scene Skins object by move.*
 - `SceneSkins & operator= (SceneSkins &&other)=default`
- Assign a Scene Skins object by move.*
 - void `loadScene (gui::SceneManager &manager, gui::Window &window, gui::Data &data)` override
- Load the scene.*
 - void `display (gui::SceneManager &manager, gui::Window &window, gui::Data &data)` override
- Display the scene.*
 - void `checkEvents (gui::SceneManager &manager, gui::Window &window, gui::Data &data)` override
- Check the events of the scene.*
 - void `createCharacter ()` override
- Create the character of the scene.*
 - void `createObject (__attribute__((unused)) gui::Data &data)` override
- Create the objects of the scene.*

Private Member Functions

- void `directionnalButtons (gui::SceneManager &manager, gui::Button &button, gui::Data &dat, gui::Window &window)`
- Set Directionnal Buttons object.*

Private Attributes

- sf::SoundBuffer `buffer`
- sf::Sound `sound`
- gui::Camera `camera` {0, 0}

6.44.1 Detailed Description

Skins scene class.

6.44.2 Constructor & Destructor Documentation

6.44.2.1 SceneSkins() [1/3]

```
SceneSkins::SceneSkins ( ) [default]
```

Construct a new Scene Skins object.

6.44.2.2 ~SceneSkins()

```
SceneSkins::~~SceneSkins ( ) [override], [default], [noexcept]
```

Destroy the Scene Skins object.

6.44.2.3 SceneSkins() [2/3]

```
SceneSkins::SceneSkins (
    const SceneSkins & other ) [delete]
```

Construct a new Scene Skins.

Parameters

<i>other</i>	
--------------	--

6.44.2.4 SceneSkins() [3/3]

```
SceneSkins::SceneSkins (
    SceneSkins && other ) [default]
```

Construct a new Scene Skins object by move.

Parameters

<i>other</i>	
--------------	--

6.44.3 Member Function Documentation

6.44.3.1 checkEvents()

```
void SceneSkins::checkEvents (
    gui::SceneManager & manager,
    gui::Window & window,
    gui::Data & data ) [override], [virtual]
```

Check the events of the scene.

Parameters

<i>window</i>	The window where the events will be checked
<i>data</i>	The data of the game
<i>manager</i>	The scene manager

Implements [IScene](#).

6.44.3.2 createCharacter()

```
void SceneSkins::createCharacter ( ) [inline], [override], [virtual]
```

Create the character of the scene.

Implements [IScene](#).

6.44.3.3 createObject()

```
void SceneSkins::createObject (
    __attribute__((unused)) gui::Data & data ) [inline], [override]
```

Create the objects of the scene.

Parameters

<i>data</i>	The data of the game
-------------	----------------------

6.44.3.4 directionnalButtons()

```
void SceneSkins::directionnalButtons (
    gui::SceneManager & manager,
    gui::Button & button,
    gui::Data & dat,
    gui::Window & window ) [private]
```

Set Directionnal Buttons object.

Parameters

<i>manager</i>	
<i>button</i>	
<i>data</i>	

6.44.3.5 display()

```
void SceneSkins::display (
    gui::SceneManager & manager,
```

```
gui::Window & window,  
gui::Data & data ) [override], [virtual]
```

Display the scene.

Parameters

<i>window</i>	The window where the scene will be displayed
<i>data</i>	The data of the game
<i>manager</i>	The scene manager

Implements [IScene](#).

6.44.3.6 loadScene()

```
void SceneSkins::loadScene (  
    gui::SceneManager & manager,  
    gui::Window & window,  
    gui::Data & data ) [override], [virtual]
```

Load the scene.

Parameters

<i>window</i>	The window where the scene will be loaded
<i>data</i>	The data of the game
<i>manager</i>	The scene manager

Implements [IScene](#).

6.44.3.7 operator=() [1/2]

```
SceneSkins& SceneSkins::operator= (  
    const SceneSkins & other ) [delete]
```

Assign a Scene Skins object by copy.

Parameters

<i>other</i>	
--------------	--

Returns

[SceneSkins&](#)

6.44.3.8 operator=() [2/2]

```
SceneSkins& SceneSkins::operator= (
    SceneSkins && other ) [default]
```

Assign a Scene Skins object by move.

Parameters

<i>other</i>	
--------------	--

Returns

SceneSkins&

6.44.4 Field Documentation

6.44.4.1 buffer

```
sf::SoundBuffer SceneSkins::buffer [private]
```

6.44.4.2 camera

```
gui::Camera SceneSkins::camera {0, 0} [private]
```

6.44.4.3 sound

```
sf::Sound SceneSkins::sound [private]
```

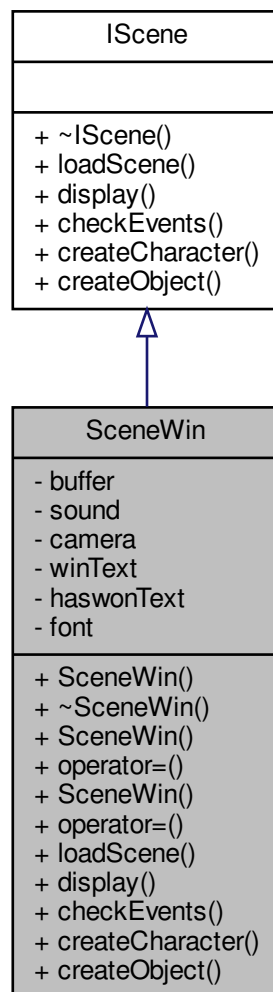
The documentation for this class was generated from the following files:

- Scene/Skins/[SceneSkins.hpp](#)
- Scene/Skins/[SceneSkins.cpp](#)

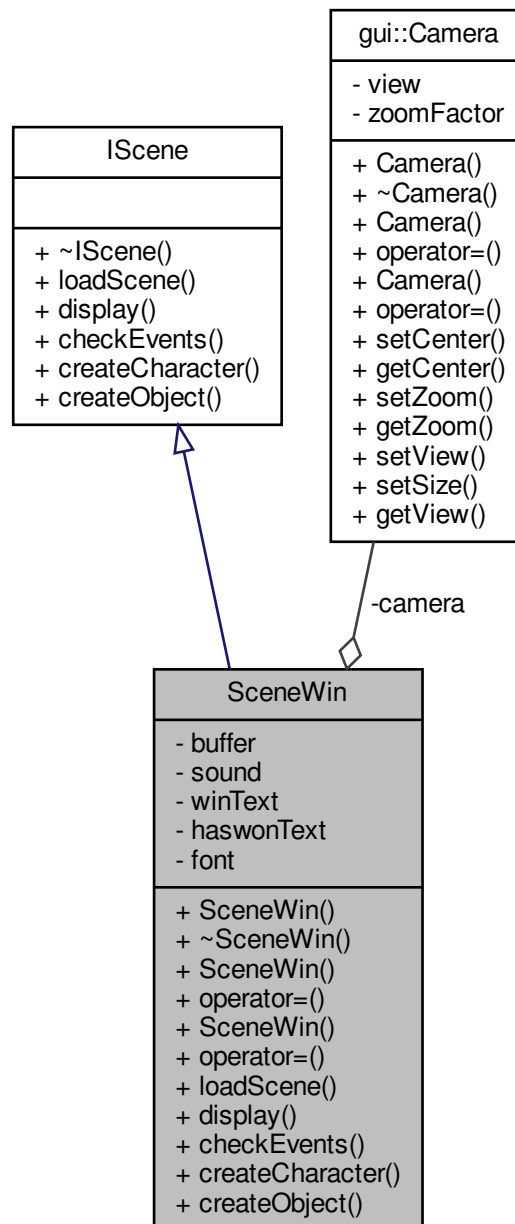
6.45 SceneWin Class Reference

```
#include <SceneWin.hpp>
```

Inheritance diagram for SceneWin:



Collaboration diagram for SceneWin:



Public Member Functions

- [SceneWin](#) ()=default
Construct a new Scene Win object.
- [~SceneWin](#) () noexcept override=default
Destroy the Scene Win object.
- [SceneWin](#) (const [SceneWin](#) &other)=delete

- Construct a new Scene Win object by copy.*

 - `SceneWin & operator= (const SceneWin &other)=delete`

Assign a Scene Win object by copy.

 - `SceneWin (SceneWin &&other)=default`

Construct a new Scene Win object by move.

 - `SceneWin & operator= (SceneWin &&other)=default`

Assign a Scene Win object by move.

 - `void loadScene (gui::SceneManager &manager, gui::Window &window, gui::Data &data) override`

Load the scene.

 - `void display (gui::SceneManager &manager, gui::Window &window, gui::Data &data) override`

Display the scene.

 - `void checkEvents (gui::SceneManager &manager, gui::Window &window, gui::Data &data) override`

Check the events of the scene.

 - `void createCharacter () override`

Create the character of the scene.

 - `void createObject (__attribute__((unused)) gui::Data &data) override`

Create the objects of the scene.

Private Attributes

- `sf::SoundBuffer buffer`
- `sf::Sound sound`
- `gui::Camera camera {0, 0}`
- `sf::Text winText`
- `sf::Text haswonText`
- `sf::Font font`

6.45.1 Constructor & Destructor Documentation

6.45.1.1 SceneWin() [1/3]

```
SceneWin::SceneWin ( ) [default]
```

Construct a new Scene Win object.

6.45.1.2 ~SceneWin()

```
SceneWin::~SceneWin ( ) [override], [default], [noexcept]
```

Destroy the Scene Win object.

6.45.1.3 SceneWin() [2/3]

```
SceneWin::SceneWin (
    const SceneWin & other ) [delete]
```

Construct a new Scene Win object by copy.

Parameters

<i>other</i>	
--------------	--

6.45.1.4 SceneWin() [3/3]

```
SceneWin::SceneWin (
    SceneWin && other ) [default]
```

Construct a new Scene Win object by move.

Parameters

<i>other</i>	
--------------	--

6.45.2 Member Function Documentation**6.45.2.1 checkEvents()**

```
void SceneWin::checkEvents (
    gui::SceneManager & manager,
    gui::Window & window,
    gui::Data & data ) [override], [virtual]
```

Check the events of the scene.

Parameters

<i>window</i>	The window where the events will be checked
<i>data</i>	The data of the game
<i>manager</i>	The scene manager

Implements [IScene](#).

6.45.2.2 createCharacter()

```
void SceneWin::createCharacter ( ) [inline], [override], [virtual]
```

Create the character of the scene.

Implements [IScene](#).

6.45.2.3 createObject()

```
void SceneWin::createObject (
    __attribute__((unused)) gui::Data & data ) [inline], [override]
```

Create the objects of the scene.

6.45.2.4 display()

```
void SceneWin::display (
    gui::SceneManager & manager,
    gui::Window & window,
    gui::Data & data ) [override], [virtual]
```

Display the scene.

Parameters

<i>window</i>	The window where the scene will be displayed
<i>data</i>	The data of the game
<i>manager</i>	The scene manager

Implements [IScene](#).

6.45.2.5 loadScene()

```
void SceneWin::loadScene (
    gui::SceneManager & manager,
    gui::Window & window,
    gui::Data & data ) [override], [virtual]
```

Load the scene.

Parameters

<i>window</i>	The window where the scene will be loaded
<i>data</i>	The data of the game
<i>manager</i>	The scene manager

Exceptions

Errors::ErrorFont	
-----------------------------------	--

Implements [IScene](#).

6.45.2.6 operator=() [1/2]

```
SceneWin& SceneWin::operator= (  
    const SceneWin & other ) [delete]
```

Assign a Scene Win object by copy.

Parameters

<i>other</i>	
--------------	--

Returns

SceneWin&

6.45.2.7 operator=() [2/2]

```
SceneWin& SceneWin::operator= (  
    SceneWin && other ) [default]
```

Assign a Scene Win object by move.

Parameters

<i>other</i>	
--------------	--

Returns

SceneWin&

6.45.3 Field Documentation

6.45.3.1 buffer

```
sf::SoundBuffer SceneWin::buffer [private]
```

6.45.3.2 camera

```
gui::Camera SceneWin::camera {0, 0} [private]
```

6.45.3.3 font

```
sf::Font SceneWin::font [private]
```

6.45.3.4 haswonText

```
sf::Text SceneWin::haswonText [private]
```

6.45.3.5 sound

```
sf::Sound SceneWin::sound [private]
```

6.45.3.6 winText

```
sf::Text SceneWin::winText [private]
```

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

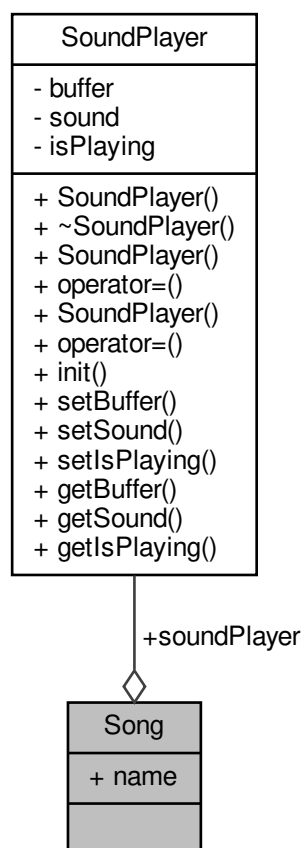
- Scene/Win/[SceneWin.hpp](#)
- Scene/Win/[SceneWin.cpp](#)

6.46 Song Struct Reference

[Song](#) struct.

```
#include <SoundBox.hpp>
```

Collaboration diagram for Song:



Data Fields

- `std::string` [name](#)
- [SoundPlayer](#) `soundPlayer`

6.46.1 Detailed Description

[Song](#) struct.

6.46.2 Field Documentation

6.46.2.1 name

```
std::string Song::name
```

6.46.2.2 soundPlayer

```
SoundPlayer Song::soundPlayer
```

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

- Components/SoundBox/[SoundBox.hpp](#)

6.47 SoundBox Class Reference

[SoundBox](#) class.

```
#include <SoundBox.hpp>
```

Collaboration diagram for SoundBox:

SoundBox
<ul style="list-style-type: none">- volume- sounds
<ul style="list-style-type: none">+ SoundBox()+ ~SoundBox()+ SoundBox()+ operator=()+ SoundBox()+ operator=()+ initSong()+ play()+ stop()+ setVolume()+ getVolume()+ getSoundStatus()

Public Member Functions

- [SoundBox](#) ()=default
Construct a new [SoundBox](#) object.
- [~SoundBox](#) () noexcept=default
Destroy the [SoundBox](#) object.
- [SoundBox](#) (const [SoundBox](#) &other)=delete
Construct a new [SoundBox](#) object by copy.
- [SoundBox](#) & [operator=](#) (const [SoundBox](#) &other)=delete
Assign a [SoundBox](#) object by copy.
- [SoundBox](#) ([SoundBox](#) &&other)=default
Construct a new [SoundBox](#) object by move.
- [SoundBox](#) & [operator=](#) ([SoundBox](#) &&other)=default
Assign a [SoundBox](#) object by move.
- void [initSong](#) (const std::string &songName, const std::string &filePath, bool looping)
Init a song.
- void [play](#) (const std::string &songName)
Play a song.
- void [stop](#) (const std::string &songName)
Stop a song.
- void [setVolume](#) (float [volume](#))
Set the Volume object.
- int [getVolume](#) ()
Get the Volume object.
- sf::SoundSource::Status [getSoundStatus](#) (const std::string &songName)

Private Attributes

- float [volume](#) {30}
- std::vector< [Song](#) > [sounds](#)

6.47.1 Detailed Description

[SoundBox](#) class.

6.47.2 Constructor & Destructor Documentation

6.47.2.1 [SoundBox](#)() [1/3]

```
SoundBox::SoundBox ( ) [default]
```

Construct a new [SoundBox](#) object.

6.47.2.2 ~SoundBox()

```
SoundBox::~SoundBox ( ) [default], [noexcept]
```

Destroy the [SoundBox](#) object.

6.47.2.3 SoundBox() [2/3]

```
SoundBox::SoundBox (
    const SoundBox & other ) [delete]
```

Construct a new [SoundBox](#) object by copy.

Parameters

<i>other</i>	
--------------	--

6.47.2.4 SoundBox() [3/3]

```
SoundBox::SoundBox (
    SoundBox && other ) [default]
```

Construct a new [SoundBox](#) object by move.

Parameters

<i>other</i>	
--------------	--

6.47.3 Member Function Documentation

6.47.3.1 getSoundStatus()

```
sf::SoundSource::Status SoundBox::getSoundStatus (
    const std::string & songName )
```

6.47.3.2 `getVolume()`

```
int SoundBox::getVolume ( )
```

Get the Volume object.

Returns

int

6.47.3.3 `initSong()`

```
void SoundBox::initSong (
    const std::string & songName,
    const std::string & filePath,
    bool looping )
```

Init a song.

Parameters

<i>songName</i>	
<i>filePath</i>	

6.47.3.4 `operator=()` [1/2]

```
SoundBox& SoundBox::operator= (
    const SoundBox & other ) [delete]
```

Assign a [SoundBox](#) object by copy.

Parameters

<i>other</i>	
--------------	--

Returns

[SoundBox&](#)

6.47.3.5 `operator=()` [2/2]

```
SoundBox& SoundBox::operator= (
    SoundBox && other ) [default]
```


Assign a [SoundBox](#) object by move.

Parameters

<i>other</i>	
--------------	--

Returns

[SoundBox](#)&

6.47.3.6 play()

```
void SoundBox::play (
    const std::string & songName )
```

Play a song.

Parameters

<i>songName</i>	
-----------------	--

6.47.3.7 setVolume()

```
void SoundBox::setVolume (
    float volume )
```

Set the Volume object.

Parameters

<i>songName</i>	
<i>volume</i>	

6.47.3.8 stop()

```
void SoundBox::stop (
    const std::string & songName )
```

Stop a song.

Parameters

<i>songName</i>	
-----------------	--

6.47.4 Field Documentation

6.47.4.1 sounds

```
std::vector<Song> SoundBox::sounds [private]
```

6.47.4.2 volume

```
float SoundBox::volume {30} [private]
```

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

- Components/SoundBox/[SoundBox.hpp](#)
- Components/SoundBox/[SoundBox.cpp](#)

6.48 SoundPlayer Class Reference

[SoundPlayer](#) class.

```
#include <SoundPlayers.hpp>
```

Collaboration diagram for SoundPlayer:

SoundPlayer
<ul style="list-style-type: none">- buffer- sound- isPlaying
<ul style="list-style-type: none">+ SoundPlayer()+ ~SoundPlayer()+ SoundPlayer()+ operator=()+ SoundPlayer()+ operator=()+ init()+ setBuffer()+ setSound()+ setIsPlaying()+ getBuffer()+ getSound()+ setIsPlaying()

Public Member Functions

- [SoundPlayer](#) ()=default
Construct a new Sound Player object.
- [~SoundPlayer](#) () noexcept=default
Destroy the Sound Player object.
- [SoundPlayer](#) (const [SoundPlayer](#) &other)=delete
Construct a new Sound Player object by copy.
- [SoundPlayer](#) & [operator=](#) (const [SoundPlayer](#) &other)=delete
Assign a Sound Player object by copy.
- [SoundPlayer](#) ([SoundPlayer](#) &&other)=default
Construct a new Sound Player object by move.
- [SoundPlayer](#) & [operator=](#) ([SoundPlayer](#) &&other)=default
Assign a Sound Player object by move.
- void [init](#) (const std::string &filePath)
Play the sound.
- void [setBuffer](#) (const sf::SoundBuffer &[buffer](#))
- void [setSound](#) (const sf::Sound &[sound](#))
- void [setIsPlaying](#) (bool [isPlaying](#))
- const sf::SoundBuffer & [getBuffer](#) () const
- sf::Sound & [getSound](#) ()
- const bool & [getIsPlaying](#) () const

Private Attributes

- sf::SoundBuffer [buffer](#)
- sf::Sound [sound](#)
- bool [isPlaying](#)

6.48.1 Detailed Description

[SoundPlayer](#) class.

6.48.2 Constructor & Destructor Documentation

6.48.2.1 [SoundPlayer](#)() [1/3]

```
SoundPlayer::SoundPlayer ( ) [default]
```

Construct a new Sound Player object.

6.48.2.2 ~SoundPlayer()

```
SoundPlayer::~~SoundPlayer ( ) [default], [noexcept]
```

Destroy the Sound Player object.

6.48.2.3 SoundPlayer() [2/3]

```
SoundPlayer::SoundPlayer (
    const SoundPlayer & other ) [delete]
```

Construct a new Sound Player object by copy.

Parameters

<i>other</i>	
--------------	--

6.48.2.4 SoundPlayer() [3/3]

```
SoundPlayer::SoundPlayer (
    SoundPlayer && other ) [default]
```

Construct a new Sound Player object by move.

Parameters

<i>other</i>	
--------------	--

6.48.3 Member Function Documentation

6.48.3.1 getBuffer()

```
const sf::SoundBuffer & SoundPlayer::getBuffer ( ) const
```

6.48.3.2 getIsPlaying()

```
const bool & SoundPlayer::getIsPlaying ( ) const
```

6.48.3.3 getSound()

```
sf::Sound & SoundPlayer::getSound ( )
```

6.48.3.4 init()

```
void SoundPlayer::init (
    const std::string & filePath )
```

Play the sound.

Parameters

<i>filePath</i>	
-----------------	--

6.48.3.5 operator=() [1/2]

```
SoundPlayer& SoundPlayer::operator= (
    const SoundPlayer & other ) [delete]
```

Assign a Sound Player object by copy.

Parameters

<i>other</i>	
--------------	--

6.48.3.6 operator=() [2/2]

```
SoundPlayer& SoundPlayer::operator= (
    SoundPlayer && other ) [default]
```

Assign a Sound Player object by move.

Parameters

<i>other</i>	
--------------	--

6.48.3.7 setBuffer()

```
void SoundPlayer::setBuffer (
```

```
const sf::SoundBuffer & buffer )
```

6.48.3.8 setIsPlaying()

```
void SoundPlayer::setIsPlaying (
    bool isPlaying )
```

6.48.3.9 setSound()

```
void SoundPlayer::setSound (
    const sf::Sound & sound )
```

6.48.4 Field Documentation

6.48.4.1 buffer

```
sf::SoundBuffer SoundPlayer::buffer [private]
```

6.48.4.2 isPlaying

```
bool SoundPlayer::isPlaying [private]
```

6.48.4.3 sound

```
sf::Sound SoundPlayer::sound [private]
```

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

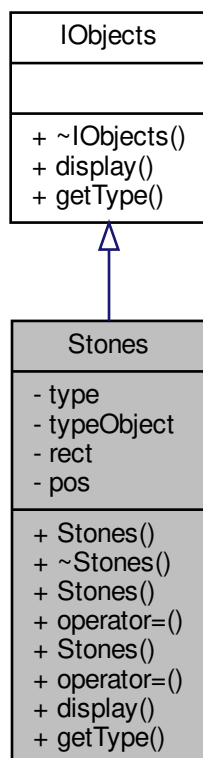
- Components/SoundBox/[SoundPlayers.hpp](#)
- Components/SoundBox/[SoundPlayers.cpp](#)

6.49 Stones Class Reference

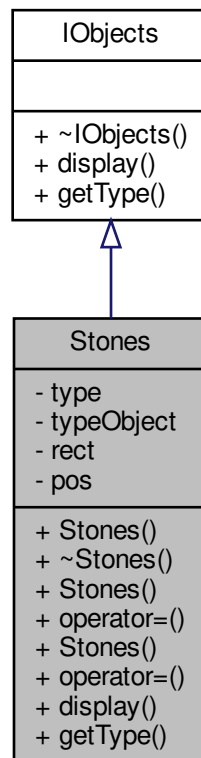
[Stones](#) class.

```
#include <Stones.hpp>
```

Inheritance diagram for Stones:



Collaboration diagram for Stones:



Public Member Functions

- `Stones (TypeStone type, sf::Vector2f pos, TypeObject typeObject)`
Construct a new `Stones` object.
- `~Stones ()` noexcept override=default
Destroy the `Stones` object.
- `Stones (const Stones &stone)=default`
Construct a new `Stones` object by copy.
- `Stones & operator= (const Stones &stone)=default`
Assign a `Stones` object by copy.
- `Stones (Stones &&stone)=default`
Construct a new `Stones` object by move.
- `Stones & operator= (Stones &&stone)=default`
Assign a `Stones` object by move.
- `void display (sf::RenderWindow &window, sf::Sprite &sprite, float size) override`
Display the object.
- `const TypeObject & getType () override`
Get The type of the object.

Private Attributes

- [TypeStone](#) `type`
Get The Sprite of the object.
- [TypeObject](#) `typeObject`
- `sf::IntRect` `rect`
- `sf::Vector2f` `pos`

6.49.1 Detailed Description

[Stones](#) class.

6.49.2 Constructor & Destructor Documentation

6.49.2.1 `Stones()` [1/3]

```
Stones::Stones (
    TypeStone type,
    sf::Vector2f pos,
    TypeObject typeObject )
```

Construct a new [Stones](#) object.

Parameters

<i>type</i>	
<i>pos</i>	
<i>typeObject</i>	

6.49.2.2 `~Stones()`

```
Stones::~~Stones ( ) [override], [default], [noexcept]
```

Destroy the [Stones](#) object.

6.49.2.3 `Stones()` [2/3]

```
Stones::Stones (
    const Stones & stone ) [default]
```

Construct a new [Stones](#) object by copy.

Parameters

<i>stone</i>	
--------------	--

6.49.2.4 Stones() [3/3]

```
Stones::Stones (
    Stones && stone ) [default]
```

Construct a new [Stones](#) object by move.

Parameters

<i>stone</i>	
--------------	--

6.49.3 Member Function Documentation**6.49.3.1 display()**

```
void Stones::display (
    sf::RenderWindow & window,
    sf::Sprite & sprite,
    float size ) [override], [virtual]
```

Display the object.

Parameters

<i>window</i>	The window where the object will be displayed
---------------	---

Implements [IObjects](#).

6.49.3.2 getType()

```
const TypeObject & Stones::getType ( ) [override], [virtual]
```

Get The type of the object.

Returns

The type of the object

Implements [IObjects](#).

6.49.3.3 operator=() [1/2]

```
Stones& Stones::operator= (
    const Stones & stone ) [default]
```

Assign a [Stones](#) object by copy.

Parameters

<i>stone</i>	
--------------	--

Returns

[Stones&](#)

6.49.3.4 operator=() [2/2]

```
Stones& Stones::operator= (
    Stones && stone ) [default]
```

Assign a [Stones](#) object by move.

Parameters

<i>stone</i>	
--------------	--

Returns

[Stones&](#)

6.49.4 Field Documentation

6.49.4.1 pos

```
sf::Vector2f Stones::pos [private]
```

6.49.4.2 rect

```
sf::IntRect Stones::rect [private]
```

6.49.4.3 type

`TypeStone` `Stones::type` [private]

Get The Sprite of the object.

Returns

The Sprite of the object

6.49.4.4 typeObject

`TypeObject` `Stones::typeObject` [private]

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

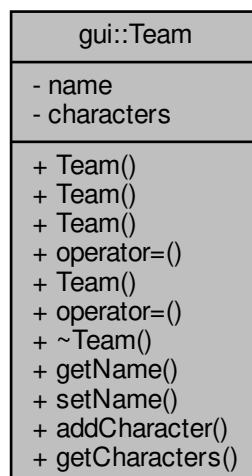
- Objects/Stones/[Stones.hpp](#)
- Objects/Stones/[Stones.cpp](#)

6.50 gui::Team Class Reference

`Team` class.

```
#include <Team.hpp>
```

Collaboration diagram for gui::Team:



Public Member Functions

- `Team ()`=default
Construct a new `Team` object.
- `Team (const std::string &name)`
Construct a new `Team` object.
- `Team (const Team &other)`=delete
Construct a new `Team` object by copy.
- `Team & operator= (const Team &other)`=delete
Assign a `Team` object by copy.
- `Team (Team &&other)`=default
Construct a new `Team` object by move.
- `Team & operator= (Team &&other)`=default
Assign a `Team` object by move.
- `~Team ()` noexcept=default
Destroy the `Team` object.
- `const std::string & getName ()` const
Get the Name.
- `void setName (const std::string &name)`
Set the Name object.
- `void addCharacter (const int &id)`
Add a character to the team.
- `const std::vector< int > & getCharacters ()`
Get the Characters.

Private Attributes

- `std::string name` {""}
- `std::vector< int > characters`

6.50.1 Detailed Description

`Team` class.

6.50.2 Constructor & Destructor Documentation

6.50.2.1 `Team()` [1/4]

```
gui::Team::Team ( ) [default]
```

Construct a new `Team` object.

6.50.2.2 `Team()` [2/4]

```
gui::Team::Team (
    const std::string & name ) [inline], [explicit]
```

Construct a new `Team` object.

Parameters

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

6.50.2.3 Team() [3/4]

```
gui::Team::Team (  
    const Team & other ) [delete]
```

Construct a new [Team](#) object by copy.

Parameters

<i>other</i>	
--------------	--

6.50.2.4 Team() [4/4]

```
gui::Team::Team (  
    Team && other ) [default]
```

Construct a new [Team](#) object by move.

Parameters

<i>other</i>	
--------------	--

6.50.2.5 ~Team()

```
gui::Team::~~Team ( ) [default], [noexcept]
```

Destroy the [Team](#) object.

6.50.3 Member Function Documentation**6.50.3.1 addCharacter()**

```
void gui::Team::addCharacter (  
    const int & id )
```

Add a character to the team.

Parameters

<i>id</i>	
-----------	--

6.50.3.2 getCharacters()

```
const std::vector< int > & gui::Team::getCharacters ( )
```

Get the Characters.

Returns

const std::vector<int>&

6.50.3.3 getName()

```
const std::string & gui::Team::getName ( ) const
```

Get the Name.

Returns

const std::string&

6.50.3.4 operator=() [1/2]

```
Team& gui::Team::operator= (
    const Team & other ) [delete]
```

Assign a [Team](#) object by copy.

Parameters

<i>other</i>	
--------------	--

Returns

[Team](#)&

6.50.3.5 operator=() [2/2]

```
Team& gui::Team::operator= (
    Team && other ) [default]
```

Assign a [Team](#) object by move.

Parameters

<i>other</i>	
--------------	--

Returns

[Team](#)&

6.50.3.6 setName()

```
void gui::Team::setName (
    const std::string & name )
```

Set the Name object.

Parameters

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

6.50.4 Field Documentation

6.50.4.1 characters

```
std::vector< int > gui::Team::characters [private]
```

6.50.4.2 name

```
std::string gui::Team::name {""} [private]
```

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

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

6.51 TextBox Class Reference

[TextBox](#) class.

```
#include <TextBox.hpp>
```

Collaboration diagram for TextBox:

TextBox
<ul style="list-style-type: none"> - descriptionText - text - inputText - box - inputMode - finished - rect
<ul style="list-style-type: none"> + TextBox() + ~TextBox() + TextBox() + operator=() + TextBox() + operator=() + textEvent() + isPressed() + draw() + getString() + setString() + initTextBox() + activate() + clear() + getPosition() + isFinished() + isInputMode()

Public Member Functions

- [TextBox](#) ()=default
Construct a new Text Box object.
- [~TextBox](#) () noexcept=default
Destroy the Text Box object.
- [TextBox](#) (const [TextBox](#) &other)=delete
Construct a new Text Box object by copy.
- [TextBox](#) & [operator=](#) (const [TextBox](#) &other)=delete
Assign a Text Box object by copy.
- [TextBox](#) ([TextBox](#) &&other)=default
Construct a new Text Box object by move.
- [TextBox](#) & [operator=](#) ([TextBox](#) &&other)=default

- Assign a Text Box object by move.*
 - int [textEvent](#) (sf::Event event)
- Handle the text event.*
 - bool [isPressed](#) (sf::Vector2f mousePos)
- Check if the mouse is pressed.*
 - void [draw](#) (sf::RenderWindow &>window, sf::Sprite &sprite)
- Draw the text box.*
 - const sf::String & [getString](#) ()
- Get the String.*
 - void [setString](#) (const sf::String &str)
- Set the String.*
 - void [initWithBox](#) (const sf::Font &font, unsigned int characterSize, sf::Vector2f size, sf::Vector2f position)
- Init the text box.*
 - void [activate](#) (bool mode)
- Activate the text box.*
 - void [clear](#) ()
- Clear the text box.*
 - sf::Vector2f [getPosition](#) ()
- Get the Position.*
 - const bool & [isFinished](#) ()
- Check if the text box is finished.*
 - const bool & [isInputMode](#) ()
- Check if the text box is in input mode.*

Private Attributes

- sf::Text [descriptionText](#)
- sf::Text [text](#)
- sf::String [inputText](#)
- sf::RectangleShape [box](#)
- bool [inputMode](#)
- bool [finished](#)
- sf::IntRect [rect](#)

6.51.1 Detailed Description

[TextBox](#) class.

6.51.2 Constructor & Destructor Documentation

6.51.2.1 [TextBox\(\)](#) [1/3]

```
TextBox::TextBox ( ) [default]
```

Construct a new Text Box object.

6.51.2.2 ~TextBox()

```
TextBox::~TextBox ( ) [default], [noexcept]
```

Destroy the Text Box object.

6.51.2.3 TextBox() [2/3]

```
TextBox::TextBox (
    const TextBox & other ) [delete]
```

Construct a new Text Box object by copy.

Parameters

<i>other</i>	
--------------	--

6.51.2.4 TextBox() [3/3]

```
TextBox::TextBox (
    TextBox && other ) [default]
```

Construct a new Text Box object by move.

Parameters

<i>other</i>	
--------------	--

6.51.3 Member Function Documentation

6.51.3.1 activate()

```
void TextBox::activate (
    bool mode )
```

Activate the text box.

Parameters

<i>mode</i>	
-------------	--

6.51.3.2 clear()

```
void TextBox::clear ( )
```

Clear the text box.

6.51.3.3 draw()

```
void TextBox::draw (
    sf::RenderWindow & window,
    sf::Sprite & sprite )
```

Draw the text box.

Parameters

<i>window</i>	
<i>sprite</i>	

6.51.3.4 getPosition()

```
sf::Vector2f TextBox::getPosition ( )
```

Get the Position.

Returns

sf::Vector2f

6.51.3.5 getString()

```
const sf::String & TextBox::getString ( )
```

Get the String.

Returns

const sf::String&

6.51.3.6 initTextBox()

```
void TextBox::initTextBox (
    const sf::Font & font,
    unsigned int characterSize,
    sf::Vector2f size,
    sf::Vector2f position )
```

Init the text box.

Parameters

<i>font</i>	
<i>characterSize</i>	
<i>size</i>	
<i>position</i>	

6.51.3.7 isFinished()

```
const bool & TextBox::isFinished ( )
```

Check if the text box is finished.

Returns

true
false

6.51.3.8 isInputMode()

```
const bool & TextBox::isInputMode ( )
```

Check if the text box is in input mode.

Returns

true
false

6.51.3.9 isPressed()

```
bool TextBox::isPressed (
    sf::Vector2f mousePos )
```

Check if the mouse is pressed.

Parameters

<i>mousePos</i>	
-----------------	--

Returns

true

false

6.51.3.10 operator=() [1/2]

```
TextBox& TextBox::operator= (
    const TextBox & other ) [delete]
```

Assign a Text Box object by copy.

Parameters

<i>other</i>	
--------------	--

Returns

TextBox&

6.51.3.11 operator=() [2/2]

```
TextBox& TextBox::operator= (
    TextBox && other ) [default]
```

Assign a Text Box object by move.

Parameters

<i>other</i>	
--------------	--

Returns

TextBox&

6.51.3.12 setString()

```
void TextBox::setString (
    const sf::String & str )
```

Set the String.

Parameters

<i>str</i>	
------------	--

6.51.3.13 textEvent()

```
int TextBox::textEvent (
    sf::Event event )
```

Handle the text event.

Parameters

<i>event</i>	
--------------	--

Returns

int

6.51.4 Field Documentation

6.51.4.1 box

```
sf::RectangleShape TextBox::box [private]
```

6.51.4.2 descriptionText

```
sf::Text TextBox::descriptionText [private]
```

6.51.4.3 finished

```
bool TextBox::finished [private]
```


6.51.4.4 inputMode

```
bool TextBox::inputMode [private]
```

6.51.4.5 inputText

```
sf::String TextBox::inputText [private]
```

6.51.4.6 rect

```
sf::IntRect TextBox::rect [private]
```

6.51.4.7 text

```
sf::Text TextBox::text [private]
```

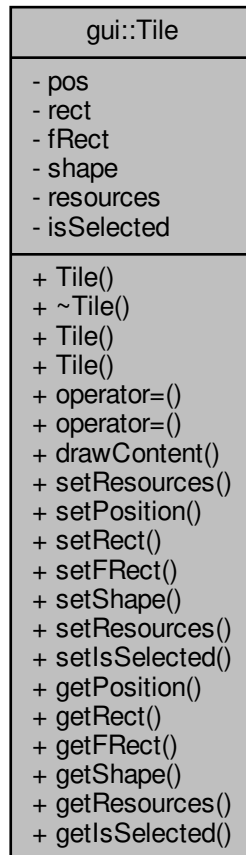
The documentation for this class was generated from the following files:

- Components/TextBox/[TextBox.hpp](#)
- Components/TextBox/[TextBox.cpp](#)

6.52 gui::Tile Class Reference

```
#include <Tile.hpp>
```

Collaboration diagram for gui::Tile:



Public Member Functions

- [Tile](#) ()=default
Construct a new [Tile](#) object.
- [~Tile](#) ()=default
Destroy the [Tile](#) object.
- [Tile](#) (const [Tile](#) &other)=delete
Construct a new [Tile](#) object by copy.
- [Tile](#) ([Tile](#) &&other)=default
Construct a new [Tile](#) object by move.
- [Tile](#) & [operator=](#) (const [Tile](#) &other)=delete
Assign a [Tile](#) object by copy.
- [Tile](#) & [operator=](#) ([Tile](#) &&other)=default
Assign a [Tile](#) object by move.
- void [drawContent](#) ([gui::ItemDrawer](#) &itemDrawer, [sf::RenderWindow](#) &>window)
Set the Resources.
- void [setResources](#) (const int &res, const int &nb)

- Set the Resources.*
- void [setPosition](#) (const sf::Vector2f &[pos](#))
- Set the Position.*
- void [setRect](#) (const sf::IntRect &[rect](#))
- Set the Rect.*
- void [setFRect](#) (const sf::FloatRect &[fRect](#))
- Set the Float Rect.*
- void [setShape](#) (const sf::RectangleShape &[shape](#))
- Set the Shape.*
- void [setResources](#) (const std::vector< [infos](#) > &[resources](#))
- Set the Resources.*
- void [setIsSelected](#) (const bool &[isSelected](#))
- Set boolean if tiles is selected.*
- const sf::Vector2f & [getPosition](#) ()
- Get the Position.*
- const sf::IntRect & [getRect](#) ()
- Get the Rect.*
- const sf::FloatRect & [getFRect](#) ()
- Get the Float Rect.*
- sf::RectangleShape & [getShape](#) ()
- Get the Shape.*
- std::vector< [infos](#) > & [getResources](#) ()
- Get the Resources.*
- const bool & [getIsSelected](#) ()
- Get the boolean selected.*

Private Attributes

- sf::Vector2f [pos](#)
- sf::IntRect [rect](#)
- sf::FloatRect [fRect](#)
- sf::RectangleShape [shape](#)
- std::vector< [infos](#) > [resources](#)
- bool [isSelected](#)

6.52.1 Constructor & Destructor Documentation

6.52.1.1 Tile() [1/3]

```
gui::Tile::Tile ( ) [default]
```

Construct a new [Tile](#) object.

6.52.1.2 ~Tile()

```
gui::Tile::~~Tile ( ) [default]
```

Destroy the [Tile](#) object.

6.52.1.3 Tile() [2/3]

```
gui::Tile::Tile (
    const Tile & other ) [delete]
```

Construct a new [Tile](#) object by copy.

Parameters

<i>other</i>	
--------------	--

6.52.1.4 Tile() [3/3]

```
gui::Tile::Tile (
    Tile && other ) [default]
```

Construct a new [Tile](#) object by move.

Parameters

<i>other</i>	
--------------	--

6.52.2 Member Function Documentation

6.52.2.1 drawContent()

```
void Tile::drawContent (
    gui::ItemDrawer & itemDrawer,
    sf::RenderWindow & window )
```

Set the Resources.

Parameters

<i>itemDrawer</i>	
<i>window</i>	

6.52.2.2 getFRect()

```
const sf::FloatRect & Tile::getFRect ( )
```

Get the Float Rect.

Returns

const sf::FloatRect&

6.52.2.3 getIsSelected()

```
const bool & Tile::getIsSelected ( )
```

Get the boolean selected.

Returns

true

false

6.52.2.4 getPosition()

```
const sf::Vector2f & Tile::getPosition ( )
```

Get the Position.

Returns

const sf::Vector2f&

6.52.2.5 getRect()

```
const sf::IntRect & Tile::getRect ( )
```

Get the Rect.

Returns

const sf::IntRect&

6.52.2.6 getResources()

```
std::vector< infos > & Tile::getResources ( )
```

Get the Resources.

Returns

std::vector<infos>&

6.52.2.7 getShape()

```
sf::RectangleShape & Tile::getShape ( )
```

Get the Shape.

Returns

sf::RectangleShape&

6.52.2.8 operator=() [1/2]

```
Tile& gui::Tile::operator= (
    const Tile & other ) [delete]
```

Assign a [Tile](#) object by copy.

Parameters

<i>itemDrawer</i>	
<i>window</i>	

6.52.2.9 operator=() [2/2]

```
Tile& gui::Tile::operator= (
    Tile && other ) [default]
```

Assign a [Tile](#) object by move.

Parameters

<i>itemDrawer</i>	
<i>window</i>	

6.52.2.10 setFRect()

```
void Tile::setFRect (
    const sf::FloatRect & fRect )
```

Set the Float Rect.

Parameters

<i>fRect</i>	
--------------	--

6.52.2.11 setIsSelected()

```
void Tile::setIsSelected (
    const bool & isSelected )
```

Set boolean if tiles is selected.

Parameters

<i>isSelected</i>	
-------------------	--

6.52.2.12 setPosition()

```
void Tile::setPosition (
    const sf::Vector2f & pos )
```

Set the Position.

Parameters

<i>pos</i>	
------------	--

6.52.2.13 setRect()

```
void Tile::setRect (
    const sf::IntRect & rect )
```

Set the Rect.

Parameters

<i>rect</i>	
-------------	--

6.52.2.14 setResources() [1/2]

```
void Tile::setResources (
    const int & res,
    const int & nb )
```

Set the Resources.

Parameters

<i>res</i>	
<i>nb</i>	

6.52.2.15 setResources() [2/2]

```
void Tile::setResources (
    const std::vector< infos > & resources )
```

Set the Resources.

Parameters

<i>resources</i>	
------------------	--

6.52.2.16 setShape()

```
void Tile::setShape (
    const sf::RectangleShape & shape )
```

Set the Shape.

Parameters

<i>shape</i>	
--------------	--

6.52.3 Field Documentation

6.52.3.1 fRect

```
sf::FloatRect Tile::fRect [private]
```

6.52.3.2 isSelected

```
bool Tile::isSelected [private]
```

6.52.3.3 pos

```
sf::Vector2f Tile::pos [private]
```

6.52.3.4 rect

```
sf::IntRect Tile::rect [private]
```

6.52.3.5 resources

```
std::vector< infos > Tile::resources [private]
```

6.52.3.6 shape

```
sf::RectangleShape Tile::shape [private]
```

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

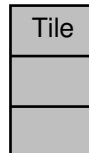
- [Map/Tile.hpp](#)
- [Map/Tile.cpp](#)

6.53 Tile Class Reference

[Tile](#) class.

```
#include <Tile.hpp>
```

Collaboration diagram for Tile:



6.53.1 Detailed Description

[Tile](#) class.

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

- Map/[Tile.hpp](#)

6.54 gui::Window Class Reference

[Window](#) class.

```
#include <Window.hpp>
```

Collaboration diagram for gui::Window:

gui::Window
<ul style="list-style-type: none"> - window - texture - sprite - event - mousePos - cursorSprite - cursorTexture - port - hostname - font - hostText - portText - clearColor
<ul style="list-style-type: none"> + Window() + ~Window() + Window() + Window() + operator=() + operator=() + createWindow() + setClearColor() + clearWindow() + displayWindow() + displayHostPortText() + destroyWindow() + isOpenWindow() + setBackground() + getBackground() + getMousePos() + getEvent() + drawObject() + setMousePos() + getWindow() + getCamera() + setCursor() + setCursorPos()

Public Member Functions

- [Window](#) ()=default
Default constructor.
- [~Window](#) () noexcept=default
Destroy the [Window](#) object.
- [Window](#) (const [Window](#) &w)=delete
Construct a new [Window](#) object by copy = delete.
- [Window](#) ([Window](#) &&w)=default

- Construct a new [Window](#) object by move = delete.

 - [Window](#) & [operator=](#) (const [Window](#) &w)=delete

Overload equal operator by copy = delete.
- [Window](#) & [operator=](#) ([Window](#) &&w)=default

Assign a [Window](#) object by move = default.
- void [createWindow](#) (const std::string &title, int width, int height, int [port](#), const std::string &hostname)

Creates a window in SFML.
- void [setClearColor](#) (sf::Color color)

Set the clear color in SFML.
- void [clearWindow](#) ()

Clear the window in SFML.
- void [displayWindow](#) ()

Display the window in SFML.
- void [displayHostPortText](#) ()

Display the text in SFML.
- void [destroyWindow](#) ()

Destroy the window in SFML.
- bool [isOpenWindow](#) ()

Check if the window is open in SFML.
- void [setBackground](#) (const std::string &path)

Set the background in SFML.
- sf::Sprite & [getBackground](#) ()

Get the background in SFML.
- const sf::Vector2f & [getMousePos](#) () const

Get the mouse position in SFML.
- sf::Event & [getEvent](#) ()

Get the event in SFML.
- template<typename T >
void [drawObject](#) (T &drawable)

Draw a drawable object in SFML.
- void [setMousePos](#) (sf::Vector2f mousePos)

Set the mouse position in SFML.
- sf::RenderWindow & [getWindow](#) ()

Get the window in SFML.
- [Camera](#) & [getCamera](#) ()

Get the camera in SFML.
- void [setCursor](#) (float size)

Set the cursor in SFML.
- void [setCursorPos](#) (sf::Vector2f pos)

Set the Cursor Pos.

Private Attributes

- sf::RenderWindow [window](#)
- sf::Texture [texture](#)
- sf::Sprite [sprite](#)
- sf::Event [event](#)
- sf::Vector2f [mousePos](#)
- sf::Sprite [cursorSprite](#)
- sf::Texture [cursorTexture](#)
- int [port](#)

- std::string [hostname](#)
- sf::Font [font](#)
- sf::Text [hostText](#)
- sf::Text [portText](#)
- sf::Color [clearColor](#)

6.54.1 Detailed Description

[Window](#) class.

6.54.2 Constructor & Destructor Documentation

6.54.2.1 Window() [1/3]

```
gui::Window::Window ( ) [default]
```

Default constructor.

6.54.2.2 ~Window()

```
gui::Window::~~Window ( ) [default], [noexcept]
```

Destroy the [Window](#) object.

6.54.2.3 Window() [2/3]

```
gui::Window::Window (
    const Window & w ) [delete]
```

Construct a new [Window](#) object by copy = delete.

6.54.2.4 Window() [3/3]

```
gui::Window::Window (
    Window && w ) [default]
```

Construct a new [Window](#) object by move = delete.

Parameters

<i>w</i>	
----------	--

6.54.3 Member Function Documentation

6.54.3.1 clearWindow()

```
void gui::Window::clearWindow ( )
```

Clear the window in SFML.

6.54.3.2 createWindow()

```
void gui::Window::createWindow (
    const std::string & title,
    int width,
    int height,
    int port,
    const std::string & hostname )
```

Creates a window in SFML.

Parameters

<i>title</i>	The title of the window.
<i>width</i>	The width of the window.
<i>height</i>	The height of the window.
<i>port</i>	The port of the server.
<i>hostname</i>	The hostname of the server.

Exceptions

<i>Errors::ErrorTexture</i>	
<i>Errors::ErrorFont</i>	

6.54.3.3 destroyWindow()

```
void gui::Window::destroyWindow ( )
```

Destroy the window in SFML.

6.54.3.4 displayHostPortText()

```
void gui::Window::displayHostPortText ( )
```

Display the text in SFML.

6.54.3.5 displayWindow()

```
void gui::Window::displayWindow ( )
```

Display the window in SFML.

6.54.3.6 drawObject()

```
template<typename T >  
void gui::Window::drawObject (   
    T & drawable ) [inline]
```

Draw a drawable object in SFML.

Parameters

<i>drawable</i>	The drawable object to draw.
-----------------	------------------------------

6.54.3.7 getBackground()

```
sf::Sprite & gui::Window::getBackground ( )
```

Get the background in SFML.

Returns

sf::Sprite The background.

6.54.3.8 getCamera()

```
Camera& gui::Window::getCamera ( )
```

Get the camera in SFML.

Returns

Camera The camera.

6.54.3.9 `getEvent()`

```
sf::Event & gui::Window::getEvent ( )
```

Get the event in SFML.

Returns

sf::Event The event.

6.54.3.10 `getMousePos()`

```
const sf::Vector2f & gui::Window::getMousePos ( ) const
```

Get the mouse position in SFML.

Returns

sf::Vector2f The mouse position.

6.54.3.11 `getWindow()`

```
sf::RenderWindow & gui::Window::getWindow ( )
```

Get the window in SFML.

Returns

sf::RenderWindow The window.

6.54.3.12 `isOpenWindow()`

```
bool gui::Window::isOpenWindow ( )
```

Check if the window is open in SFML.

Returns

True if the window is open, False otherwise.

6.54.3.13 `operator=()` [1/2]

```
Window& gui::Window::operator= (
    const Window & w ) [delete]
```

Overload equal operator by copy = delete.

Parameters

<i>w</i>	
----------	--

Returns

[Window](#)&

6.54.3.14 operator=() [2/2]

```
Window& gui::Window::operator= (  
    Window && w ) [default]
```

Assign a [Window](#) object by move = default.

Parameters

<i>w</i>	
----------	--

Returns

[Window](#)&

6.54.3.15 setBackground()

```
void gui::Window::setBackground (  
    const std::string & path )
```

Set the background in SFML.

Parameters

<i>path</i>	The path of the background.
-------------	-----------------------------

Exceptions

Errors::ErrorTexture	
--------------------------------------	--

6.54.3.16 setClearColor()

```
void gui::Window::setClearColor (  
    sf::Color color )
```

Set the clear color in SFML.

Parameters

<i>color</i>	The color to set.
--------------	-------------------

6.54.3.17 `setCursor()`

```
void gui::Window::setCursor (
    float size )
```

Set the cursor in SFML.

Parameters

<i>size</i>	The size of the cursor.
-------------	-------------------------

6.54.3.18 `setCursorPos()`

```
void gui::Window::setCursorPos (
    sf::Vector2f pos )
```

Set the Cursor Pos.

Parameters

<i>pos</i>	
------------	--

6.54.3.19 `setMousePos()`

```
void gui::Window::setMousePos (
    sf::Vector2f mousePos )
```

Set the mouse position in SFML.

Returns

sf::Vector2f The mouse position.

6.54.4 Field Documentation

6.54.4.1 clearColor

```
sf::Color gui::Window::clearColor [private]
```

The clear color in SFML

6.54.4.2 cursorSprite

```
sf::Sprite gui::Window::cursorSprite [private]
```

6.54.4.3 cursorTexture

```
sf::Texture gui::Window::cursorTexture [private]
```

6.54.4.4 event

```
sf::Event gui::Window::event [private]
```

6.54.4.5 font

```
sf::Font gui::Window::font [private]
```

6.54.4.6 hostname

```
std::string gui::Window::hostname [private]
```

6.54.4.7 hostText

```
sf::Text gui::Window::hostText [private]
```

6.54.4.8 mousePos

```
sf::Vector2f gui::Window::mousePos [private]
```

6.54.4.9 port

```
int gui::Window::port [private]
```

6.54.4.10 portText

```
sf::Text gui::Window::portText [private]
```

6.54.4.11 sprite

```
sf::Sprite gui::Window::sprite [private]
```

6.54.4.12 texture

```
sf::Texture gui::Window::texture [private]
```

6.54.4.13 window

```
sf::RenderWindow gui::Window::window [private]
```

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

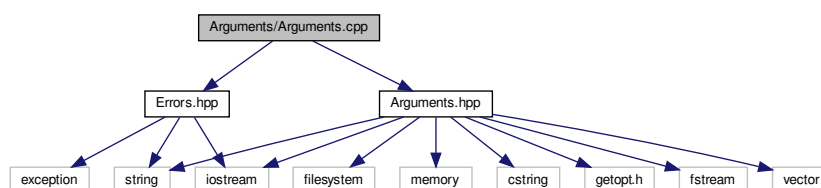
- Components/Window/[Window.hpp](#)
- Components/Window/[Window.cpp](#)

Chapter 7

File Documentation

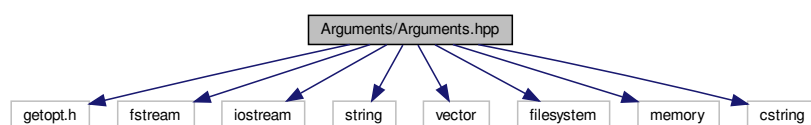
7.1 Arguments/Arguments.cpp File Reference

```
#include "Arguments.hpp"
#include "Errors.hpp"
Include dependency graph for Arguments.cpp:
```

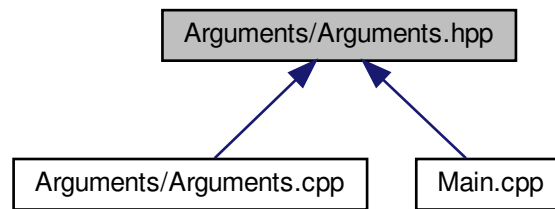


7.2 Arguments/Arguments.hpp File Reference

```
#include <getopt.h>
#include <fstream>
#include <iostream>
#include <string>
#include <vector>
#include <filesystem>
#include <memory>
#include <cstring>
Include dependency graph for Arguments.hpp:
```



This graph shows which files directly or indirectly include this file:



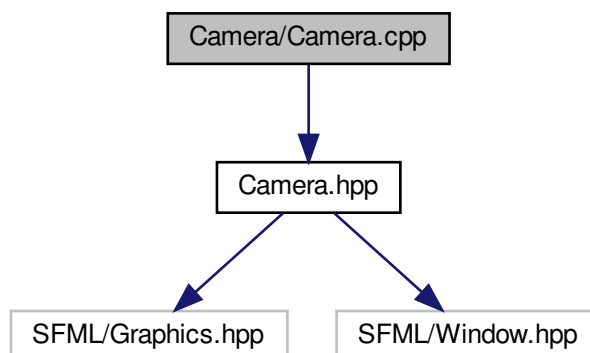
Data Structures

- class [Arguments](#)
Class for the arguments.

7.3 Camera/Camera.cpp File Reference

```
#include "Camera.hpp"
```

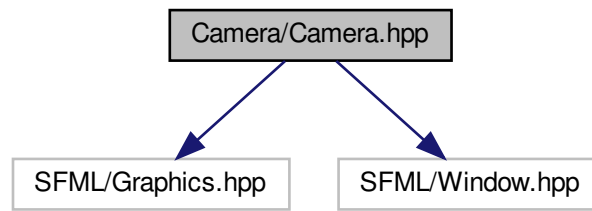
Include dependency graph for Camera.cpp:



7.4 Camera/Camera.hpp File Reference

```
#include <SFML/Graphics.hpp>  
#include <SFML/Window.hpp>
```

Include dependency graph for Camera.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class [gui::Camera](#)
Camera class.

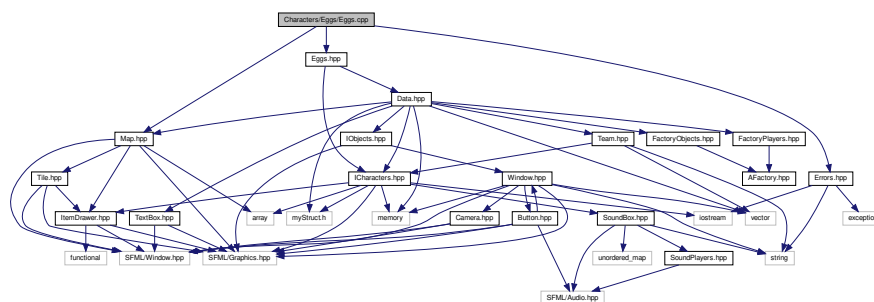
Namespaces

- [gui](#)

7.5 Characters/Eggs/Eggs.cpp File Reference

```
#include "Eggs.hpp"
#include "Map.hpp"
#include "Errors.hpp"
```

Include dependency graph for Eggs.cpp:



Functions

- static sf::IntRect [findRectEggs](#) ([TypeEggs](#) typeParams)

Variables

- static std::vector< sf::IntRect > [rectEggs](#)

7.5.1 Function Documentation

7.5.1.1 findRectEggs()

```
static sf::IntRect findRectEggs (
    TypeEggs typeParams ) [static]
```

7.5.2 Variable Documentation

7.5.2.1 rectEggs

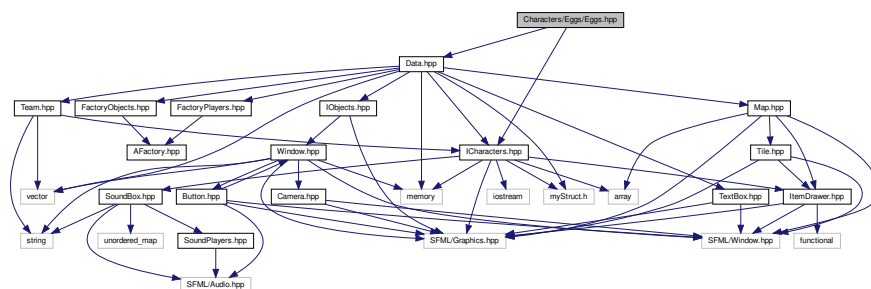
```
std::vector<sf::IntRect> rectEggs [static]
```

Initial value:

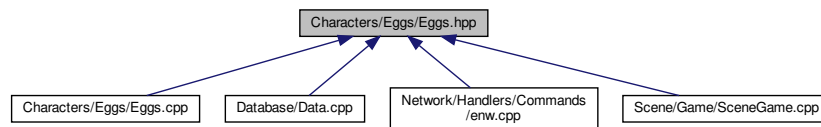
```
= {
    sf::IntRect(95, 6599, 105, 83),
    sf::IntRect(95, 6682, 105, 83),
    sf::IntRect(95, 6765, 105, 83),
    sf::IntRect(95, 6848, 105, 83),
    sf::IntRect(95, 6931, 105, 83),
    sf::IntRect(95, 7014, 105, 83),
    sf::IntRect(95, 7097, 105, 83)
}
```

7.6 Characters/Eggs/Eggs.hpp File Reference

```
#include "ICharacters.hpp"
#include "Data.hpp"
Include dependency graph for Eggs.hpp:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- class [Eggs](#)
Eggs class.

Enumerations

- enum class [TypeEggs](#) {
POURPRE, BLUE, YELLOW, ICE,
PURPLE, GREEN, WHITE }
TypesEggs enum class.

7.6.1 Enumeration Type Documentation

7.6.1.1 TypeEggs

```
enum TypeEggs [strong]
```

TypesEggs enum class.

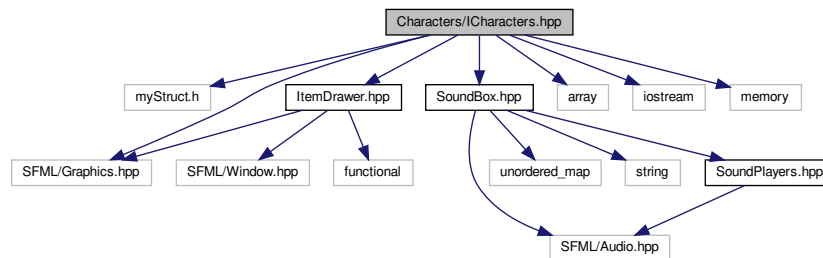
Enumerator

POURPRE	
BLUE	
YELLOW	
ICE	
PURPLE	
GREEN	
WHITE	

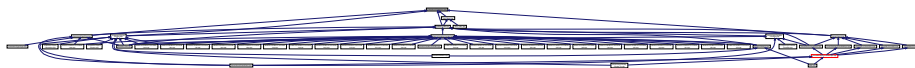
7.7 Characters/ICharacters.hpp File Reference

```
#include "myStruct.h"
#include "ItemDrawer.hpp"
```

```
#include "SoundBox.hpp"
#include <SFML/Graphics.hpp>
#include <array>
#include <iostream>
#include <memory>
Include dependency graph for ICharacters.hpp:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- class [ICharacters](#)
Interface for the characters.

Enumerations

- enum class [CharacterType](#) { [Player](#) , [Egg](#) , [Commentator](#) }

7.7.1 Enumeration Type Documentation

7.7.1.1 CharacterType

```
enum CharacterType [strong]
```

Enumerator

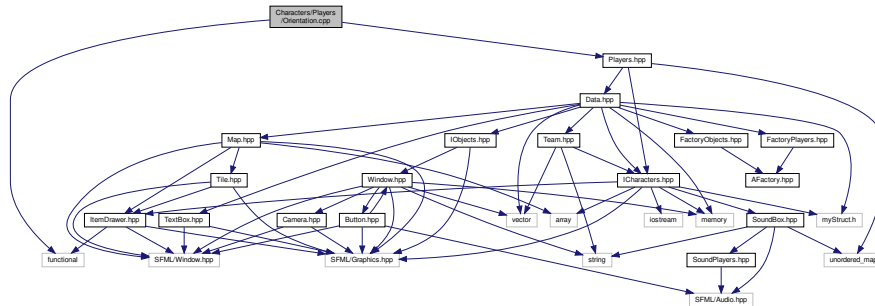
Player	
Egg	
Commentator	

7.8 Characters/Players/Orientation.cpp File Reference

```
#include "Players.hpp"
```

```
#include <functional>
```

Include dependency graph for Orientation.cpp:



Functions

- `int redRectFunc` (Orientation orientation)
- `int yellowRectFunc` (Orientation orientation)
- `int greenRectFunc` (Orientation orientation)
- `int blueRectFunc` (Orientation orientation)
- `int darkBlueRectFunc` (Orientation orientation)
- `int purpleRectFunc` (Orientation orientation)
- `int getRightRect` (const std::string &colorParams, Orientation orientation)
- `int getFacingRectByColor` (const std::string &colorParams)

7.8.1 Function Documentation

7.8.1.1 blueRectFunc()

```
int blueRectFunc (
    Orientation orientation )
```

7.8.1.2 darkBlueRectFunc()

```
int darkBlueRectFunc (
    Orientation orientation )
```

7.8.1.3 getFacingRectByColor()

```
int getFacingRectByColor (
    const std::string & colorParams )
```

7.8.1.4 getRightRect()

```
int getRightRect (
    const std::string & colorParams,
    Orientation orientation )
```

7.8.1.5 greenRectFunc()

```
int greenRectFunc (
    Orientation orientation )
```

7.8.1.6 purpleRectFunc()

```
int purpleRectFunc (
    Orientation orientation )
```

7.8.1.7 redRectFunc()

```
int redRectFunc (
    Orientation orientation )
```

7.8.1.8 yellowRectFunc()

```
int yellowRectFunc (
    Orientation orientation )
```

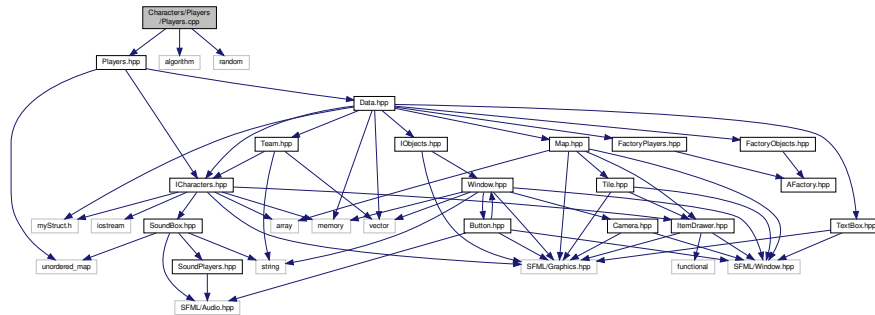
7.9 Characters/Players/Players.cpp File Reference

```
#include "Players.hpp"
```

```
#include <algorithm>
```

```
#include <random>
```

Include dependency graph for Players.cpp:



Functions

- int **getRightRect** (const std::string &color, Orientation orientation)

Variables

- constexpr char **PATH_GOLEM** [] = "./gui/Resources/golem_spritesheet.png"

7.9.1 Function Documentation

7.9.1.1 getRightRect()

```
int getRightRect (
    const std::string & color,
    Orientation orientation )
```

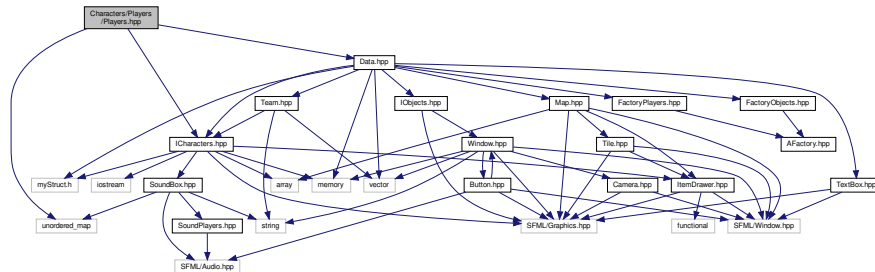
7.9.2 Variable Documentation

7.9.2.1 PATH_GOLEM

```
constexpr char PATH_GOLEM[] = "../gui/Resources/golem_spritesheet.png" [constexpr]
```

7.10 Characters/Players/Players.hpp File Reference

```
#include "ICharacters.hpp"
#include "Data.hpp"
#include <unordered_map>
Include dependency graph for Players.hpp:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- class [Players](#)
Players class.

Variables

- static `std::unordered_map< std::string, sf::Color >` [broadcastColor](#)

7.10.1 Variable Documentation

7.10.1.1 broadcastColor

```
std::unordered_map<std::string, sf::Color> broadcastColor [static]
```

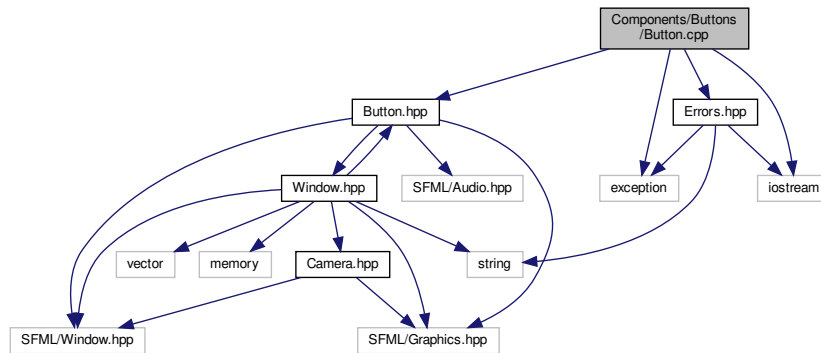
Initial value:

```
= {
    {"red", sf::Color(242, 14, 27, 155)},
    {"yellow", sf::Color(242, 228, 24, 155)},
    {"green", sf::Color(34, 239, 23, 155)},
    {"blue", sf::Color(32, 173, 240, 155)},
    {"darkBlue", sf::Color(17, 21, 238, 155)},
    {"purple", sf::Color(183, 21, 243, 155)},
}
```

7.11 Components/Buttons/Button.cpp File Reference

```
#include "Button.hpp"
#include <exception>
#include <iostream>
#include "Errors.hpp"
```

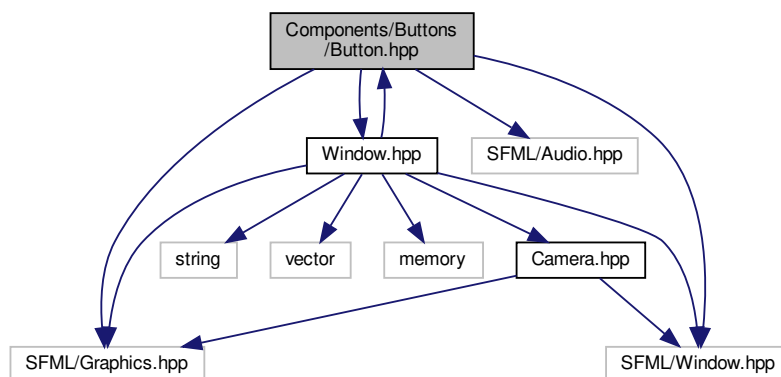
Include dependency graph for Button.cpp:



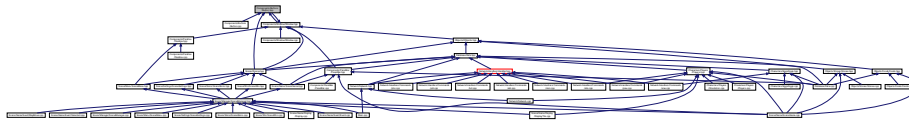
7.12 Components/Buttons/Button.hpp File Reference

```
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
#include <SFML/Audio.hpp>
#include "Window.hpp"
```

Include dependency graph for Button.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class [gui::Button](#)
Button class.

Namespaces

- [gui](#)

Enumerations

- enum [ButtonState](#) { [BUTTON_IDLE](#) = 0 , [BUTTON_HOVER](#) , [BUTTON_CLICKED](#) , [BUTTON_LOCKED](#) }

7.12.1 Enumeration Type Documentation

7.12.1.1 ButtonState

enum [ButtonState](#)

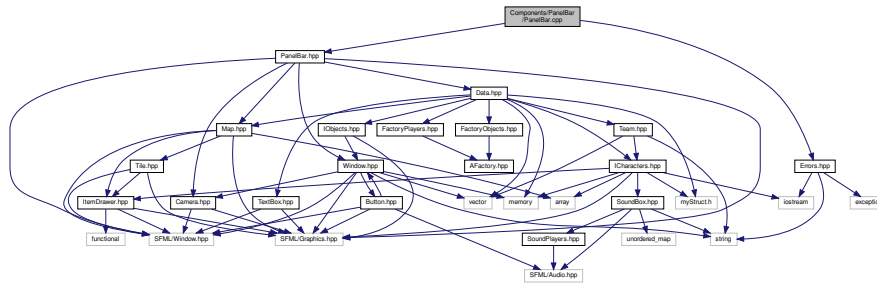
Enumerator

BUTTON_IDLE	
BUTTON_HOVER	
BUTTON_CLICKED	
BUTTON_LOCKED	

7.13 Components/PanelBar/PanelBar.cpp File Reference

```
#include "PanelBar.hpp"
#include "Errors.hpp"
```


Include dependency graph for PanelBar.cpp:



Functions

- int [getFacingRectByColor](#) (const std::string &color)

7.13.1 Function Documentation

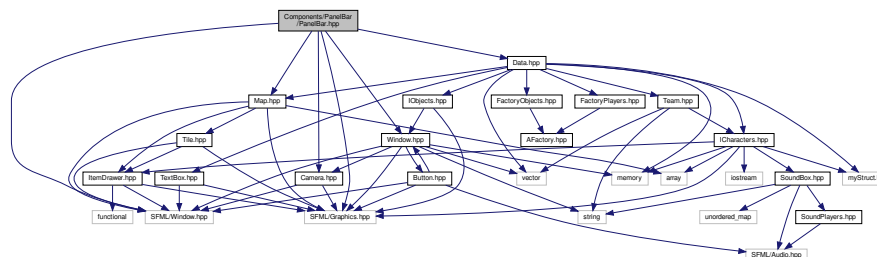
7.13.1.1 getFacingRectByColor()

```
int getFacingRectByColor (
    const std::string & color )
```

7.14 Components/PanelBar/PanelBar.hpp File Reference

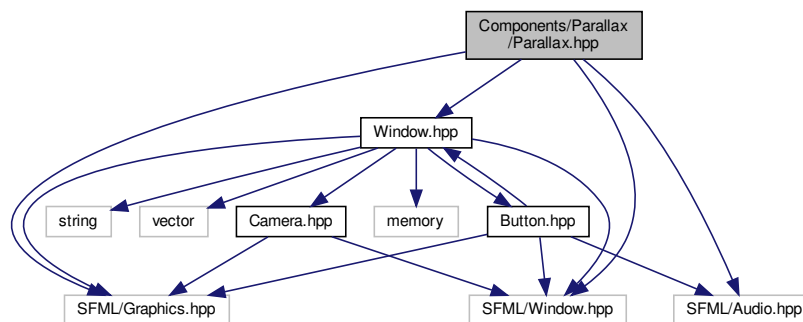
```
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
#include "Camera.hpp"
#include "Map.hpp"
#include "Window.hpp"
#include "Data.hpp"
```

Include dependency graph for PanelBar.hpp:




```
#include "Window.hpp"
```

Include dependency graph for Parallax.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct [Background](#)
Background struct.
- class [gui::Parallax](#)
Parallax class.

Namespaces

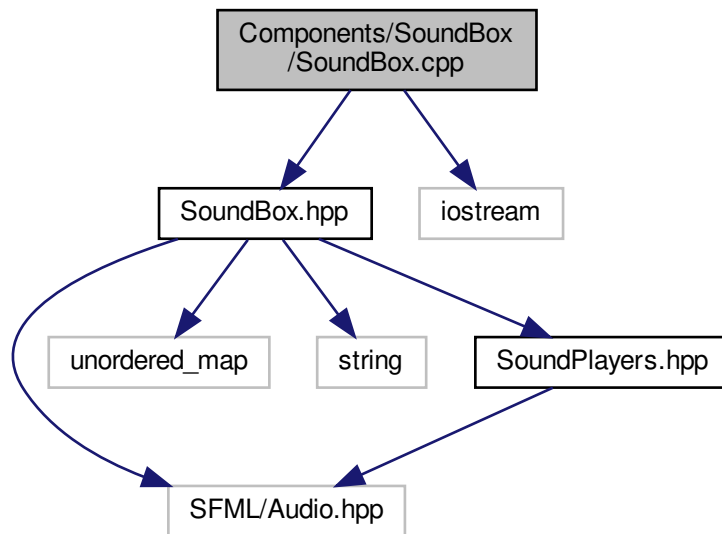
- [gui](#)

7.17 Components/SoundBox/SoundBox.cpp File Reference

```
#include "SoundBox.hpp"
```

```
#include <iostream>
```

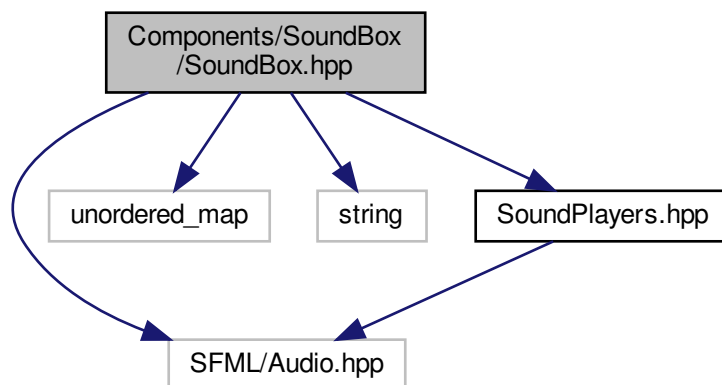
Include dependency graph for SoundBox.cpp:



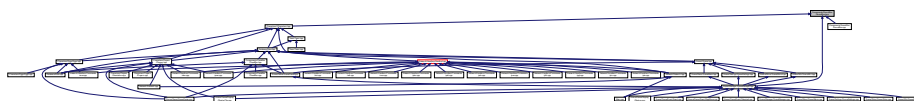
7.18 Components/SoundBox/SoundBox.hpp File Reference

```
#include <SFML/Audio.hpp>
#include <unordered_map>
#include <string>
#include "SoundPlayers.hpp"
```

Include dependency graph for SoundBox.hpp:



This graph shows which files directly or indirectly include this file:



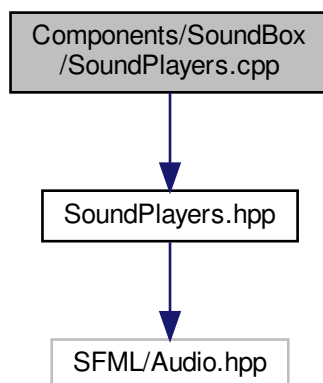
Data Structures

- struct [Song](#)
[Song](#) struct.
- class [SoundBox](#)
[SoundBox](#) class.

7.19 Components/SoundBox/SoundPlayers.cpp File Reference

```
#include "SoundPlayers.hpp"
```

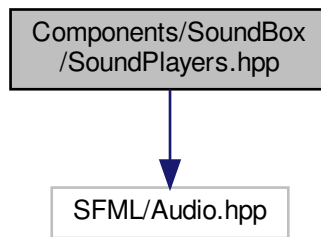
Include dependency graph for SoundPlayers.cpp:



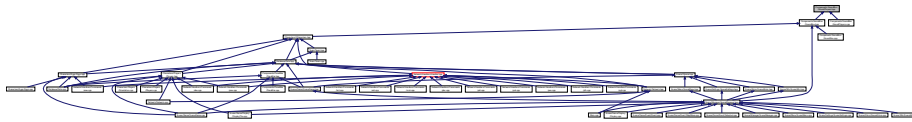
7.20 Components/SoundBox/SoundPlayers.hpp File Reference

```
#include "SFML/Audio.hpp"
```

Include dependency graph for SoundPlayers.hpp:



This graph shows which files directly or indirectly include this file:



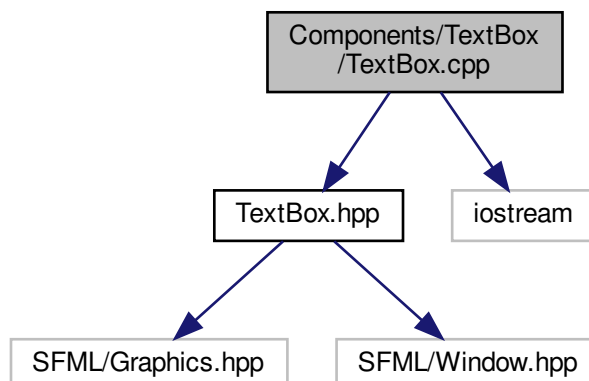
Data Structures

- class [SoundPlayer](#)
SoundPlayer class.

7.21 Components/TextBox/TextBox.cpp File Reference

```
#include "TextBox.hpp"
#include <iostream>
```

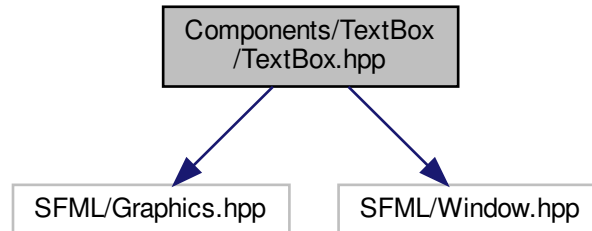
Include dependency graph for TextBox.cpp:



7.22 Components/TextBox/TextBox.hpp File Reference

```
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
```

Include dependency graph for TextBox.hpp:



This graph shows which files directly or indirectly include this file:



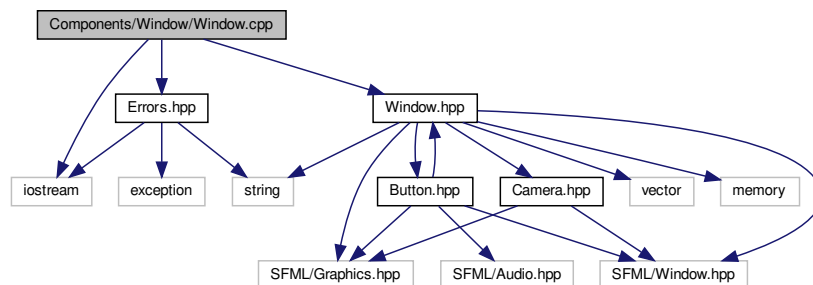
Data Structures

- class [TextBox](#)
TextBox class.

7.23 Components/Window/Window.cpp File Reference

```
#include "Window.hpp"
#include <iostream>
#include "Errors.hpp"
```

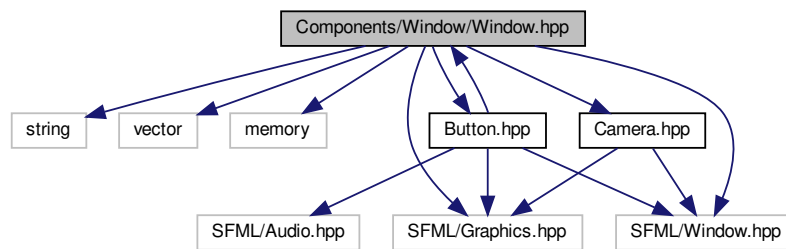
Include dependency graph for Window.cpp:



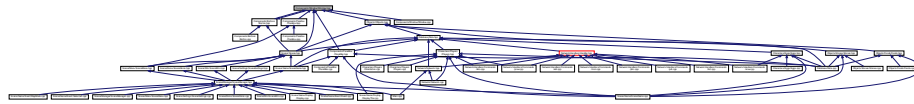
7.24 Components/Window/Window.hpp File Reference

```
#include <string>
#include <vector>
#include <memory>
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
#include "Button.hpp"
#include "Camera.hpp"
```

Include dependency graph for Window.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class `gui::Window`
Window class.

Namespaces

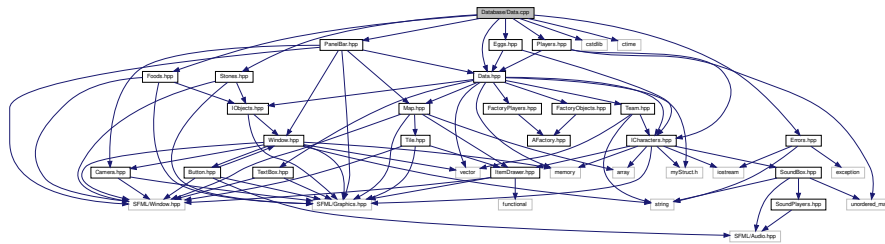
- `gui`

7.25 Database/Data.cpp File Reference

```
#include "Data.hpp"
#include "Foods.hpp"
#include "Stones.hpp"
#include "PanelBar.hpp"
#include "Players.hpp"
#include "Eggs.hpp"
#include <cstdlib>
```

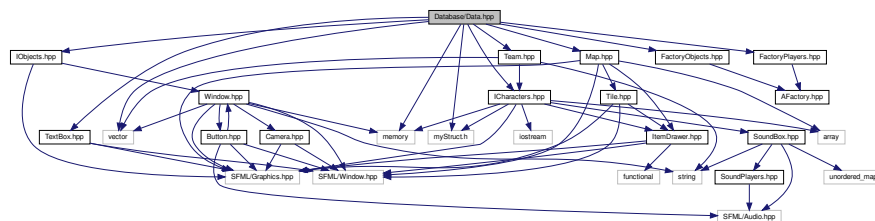


```
#include <ctime>
#include "Errors.hpp"
Include dependency graph for Data.cpp:
```

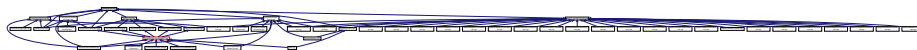


7.26 Database/Data.hpp File Reference

```
#include <vector>
#include <memory>
#include "IObjects.hpp"
#include "ICharacters.hpp"
#include "Map.hpp"
#include "FactoryObjects.hpp"
#include "FactoryPlayers.hpp"
#include "myStruct.h"
#include "Team.hpp"
#include "TextBox.hpp"
Include dependency graph for Data.hpp:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- class `gui::Data`
Data class.

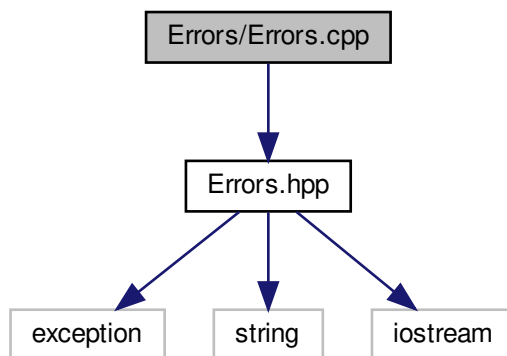
Namespaces

- gui

7.27 Errors/Errors.cpp File Reference

```
#include "Errors.hpp"
```

Include dependency graph for Errors.cpp:



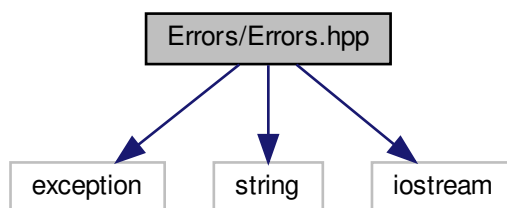
7.28 Errors/Errors.hpp File Reference

```
#include <exception>
```

```
#include <string>
```

```
#include <iostream>
```

Include dependency graph for Errors.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class [Errors](#)
- class [Errors::ErrorTeam](#)
A class for Team Error.
- class [Errors::ErrorTexture](#)
A class for Texture Error.
- class [Errors::ErrorUsage](#)
A class for Usage Error.
- class [Errors::ErrorFile](#)
A class for file Error.
- class [Errors::ErrorOpen](#)
A class for open Error.
- class [Errors::ErrorCharacter](#)
A class for Character Error.
- class [Errors::ErrorEnv](#)
A class for env Error.
- class [Errors::ErrorFont](#)
A class for font Error.
- class [Errors::ErrorSocket](#)
A class for socket Error.
- class [Errors::ErrorBroadcast](#)
A class for broadcast Error.
- class [Errors::ErrorElevate](#)
A class for Elevate Error.
- class [Errors::ErrorColor](#)
- class [Errors::ErrorInventory](#)
A class for Inventory Error.
- class [Errors::ErrorButton](#)
A class for Button Error.

7.29 Factory/AFactory.hpp File Reference

This graph shows which files directly or indirectly include this file:



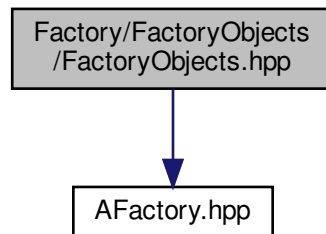
Data Structures

- class [AFactory](#)
AFactory class.

7.30 Factory/FactoryObjects/FactoryObjects.hpp File Reference

```
#include "AFactory.hpp"
```

Include dependency graph for FactoryObjects.hpp:



This graph shows which files directly or indirectly include this file:



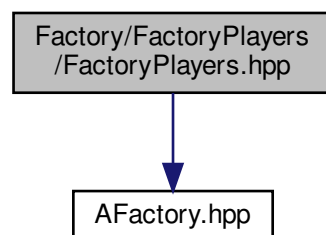
Data Structures

- class [FactoryObjects](#)

7.31 Factory/FactoryPlayers/FactoryPlayers.hpp File Reference

```
#include "AFactory.hpp"
```

Include dependency graph for FactoryPlayers.hpp:



This graph shows which files directly or indirectly include this file:



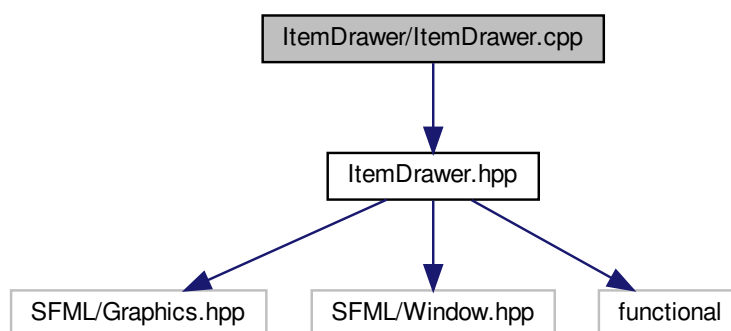
Data Structures

- class [FactoryPlayers](#)
FactoryPlayers class.

7.32 ItemDrawer/ItemDrawer.cpp File Reference

```
#include "ItemDrawer.hpp"
```

Include dependency graph for ItemDrawer.cpp:



Namespaces

- [gui](#)

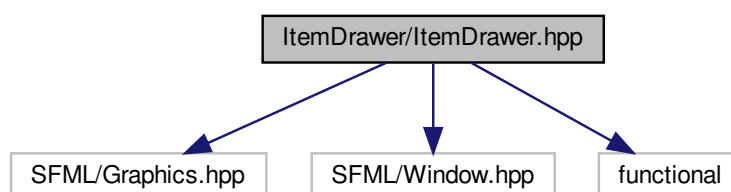
7.33 ItemDrawer/ItemDrawer.hpp File Reference

```
#include <SFML/Graphics.hpp>
```

```
#include <SFML/Window.hpp>
```

```
#include <functional>
```

Include dependency graph for ItemDrawer.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class [gui::ItemDrawer](#)
ItemDrawer class.

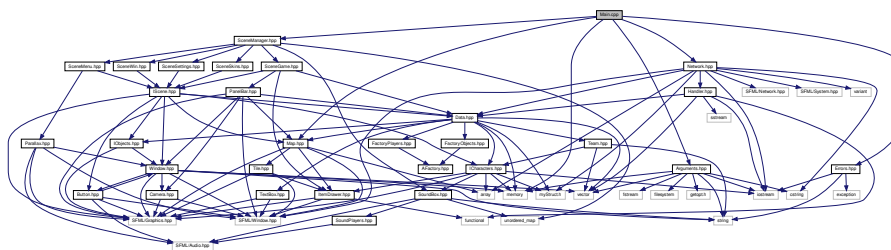
Namespaces

- [gui](#)

7.34 Main.cpp File Reference

```
#include "Map.hpp"
#include "SceneManager.hpp"
#include "Arguments.hpp"
#include "Network.hpp"
#include <memory>
#include "Errors.hpp"
```

Include dependency graph for Main.cpp:



Functions

- int [launch](#) ([gui::Network](#) &client, [Arguments](#) &args)
- int [main](#) (const int argc, char *const argv[])

7.34.1 Function Documentation

7.34.1.1 launch()

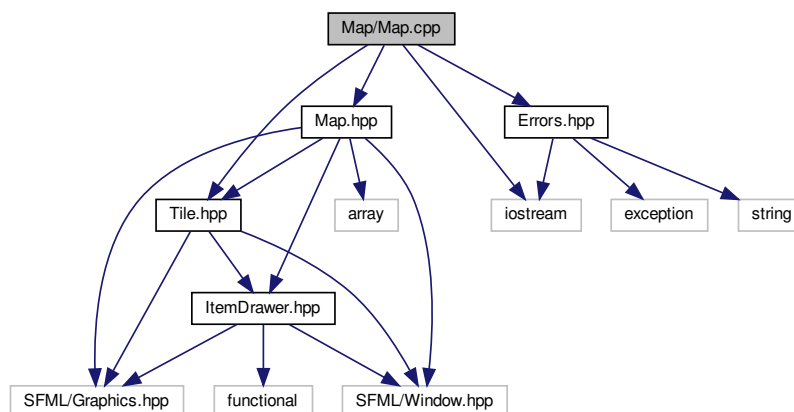
```
int launch (
    gui::Network & client,
    Arguments & args )
```

7.34.1.2 main()

```
int main (
    const int argc,
    char *const argv[] )
```

7.35 Map/Map.cpp File Reference

```
#include "Map.hpp"
#include "Tile.hpp"
#include <iostream>
#include "Errors.hpp"
Include dependency graph for Map.cpp:
```



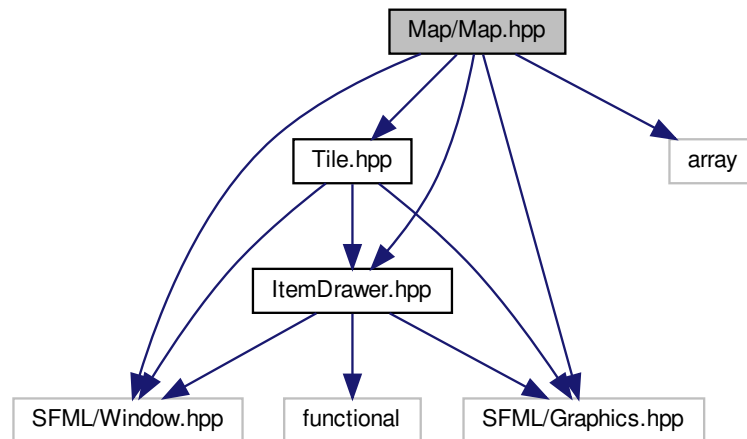
Namespaces

- [gui](#)

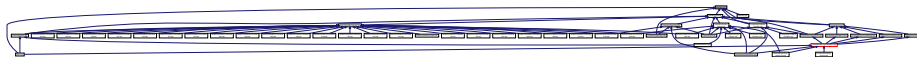
7.36 Map/Map.hpp File Reference

```
#include "ItemDrawer.hpp"
#include "Tile.hpp"
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
#include <array>
```

Include dependency graph for Map.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class [gui::Map](#)
Map Class.

Namespaces

- [gui](#)

Variables

- constexpr int [RECT_GRASS](#) = 64
- constexpr char [PATH_BACK](#) [] = "../gui/Resources/tiles.png"

7.36.1 Variable Documentation

7.36.1.1 PATH_BACK

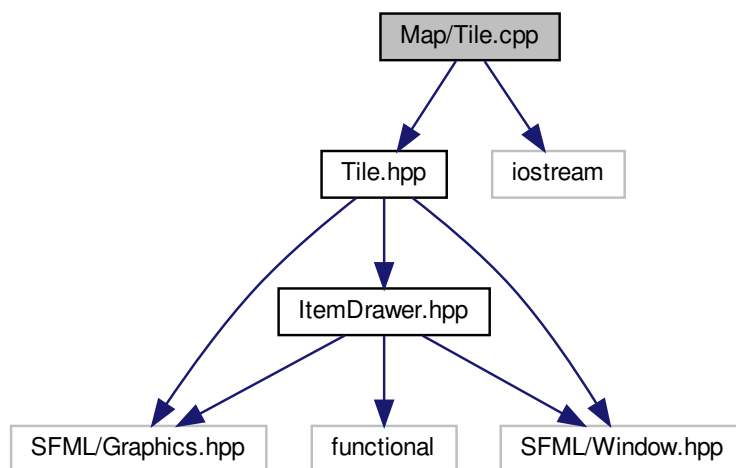
```
constexpr char PATH_BACK[] = "../gui/Resources/tiles.png" [constexpr]
```

7.36.1.2 RECT_GRASS

```
constexpr int RECT_GRASS = 64 [constexpr]
```

7.37 Map/Tile.cpp File Reference

```
#include "Tile.hpp"  
#include <iostream>  
Include dependency graph for Tile.cpp:
```



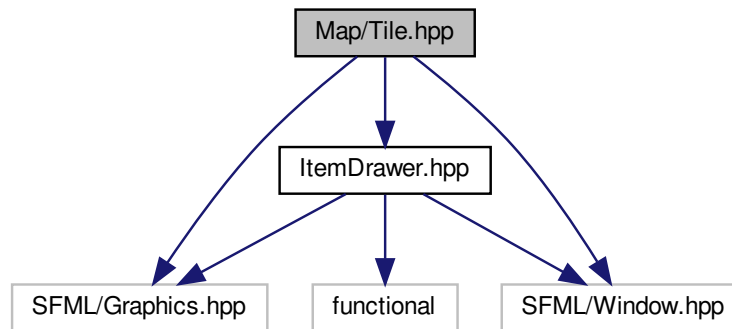
Namespaces

- [gui](#)

7.38 Map/Tile.hpp File Reference

```
#include "ItemDrawer.hpp"
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
```

Include dependency graph for Tile.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct [infos](#)
- class [gui::Tile](#)

Namespaces

- [gui](#)

Variables

- constexpr float [TILE_SIZE](#) = 64

7.38.1 Variable Documentation

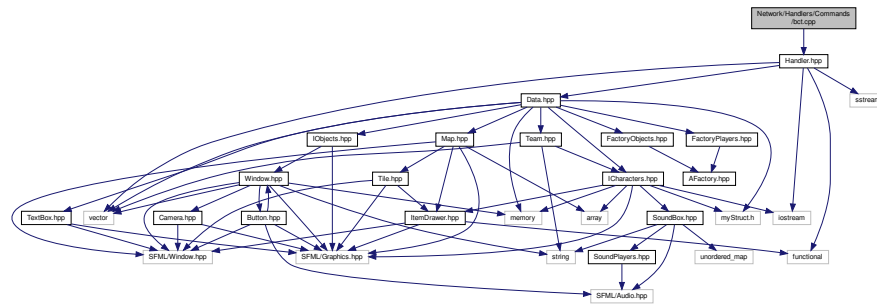
7.38.1.1 TILE_SIZE

```
constexpr float TILE_SIZE = 64 [constexpr]
```

7.39 Network/Handlers/Commands/bct.cpp File Reference

```
#include "Handler.hpp"
```

Include dependency graph for bct.cpp:



Functions

- `std::size_t get_char_occurrence (std::string &str, char c)`

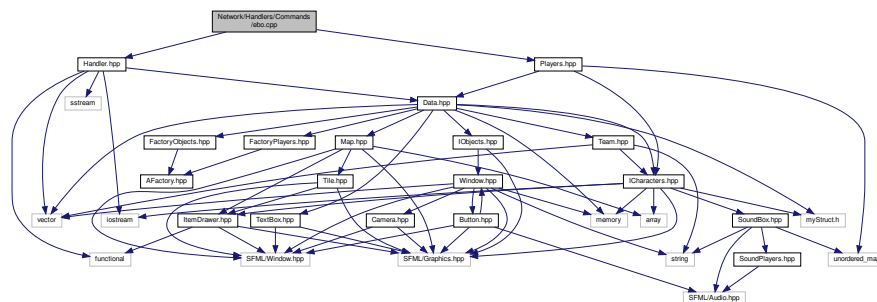
7.39.1 Function Documentation

7.39.1.1 `get_char_occurrence()`

```
std::size_t get_char_occurrence (
    std::string & str,
    char c )
```

7.40 Network/Handlers/Commands/ebo.cpp File Reference

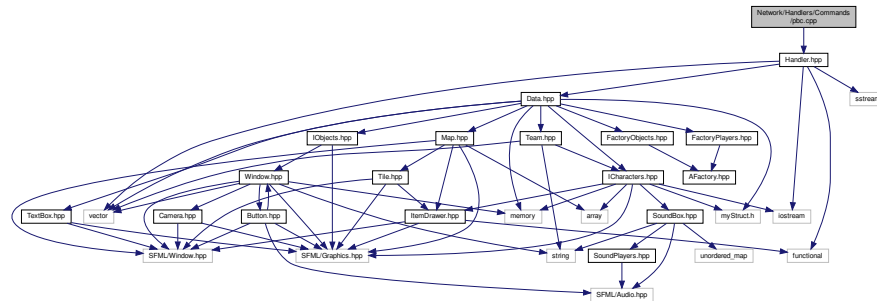
```
#include "Handler.hpp"
#include "Players.hpp"
Include dependency graph for ebo.cpp:
```



7.44 Network/Handlers/Commands/pbc.cpp File Reference

```
#include "Handler.hpp"
```

Include dependency graph for pbc.cpp:

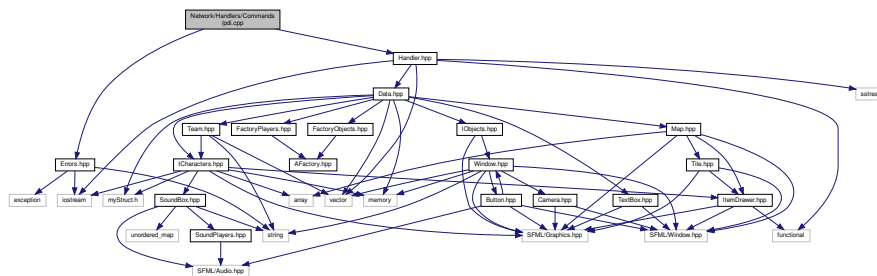


7.45 Network/Handlers/Commands/pdi.cpp File Reference

```
#include "Handler.hpp"
```

```
#include "Errors.hpp"
```

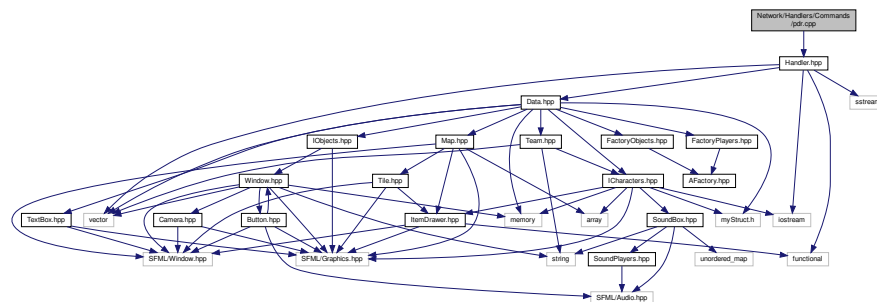
Include dependency graph for pdi.cpp:



7.46 Network/Handlers/Commands/pdr.cpp File Reference

```
#include "Handler.hpp"
```

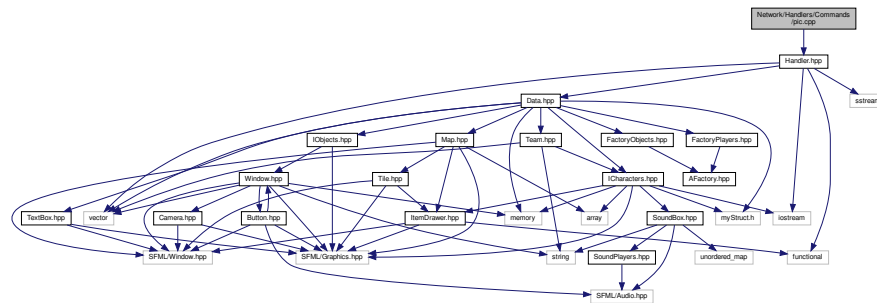
Include dependency graph for pdr.cpp:



7.50 Network/Handlers/Commands/pic.cpp File Reference

```
#include "Handler.hpp"
```

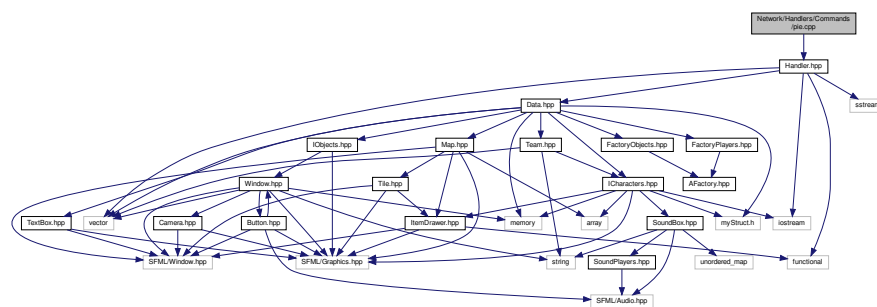
Include dependency graph for pic.cpp:



7.51 Network/Handlers/Commands/pie.cpp File Reference

```
#include "Handler.hpp"
```

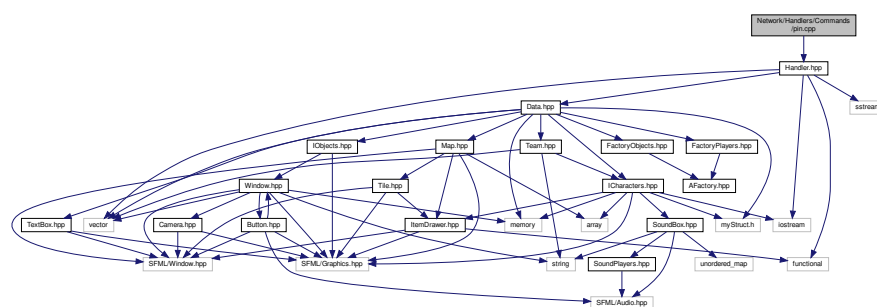
Include dependency graph for pie.cpp:



7.52 Network/Handlers/Commands/pin.cpp File Reference

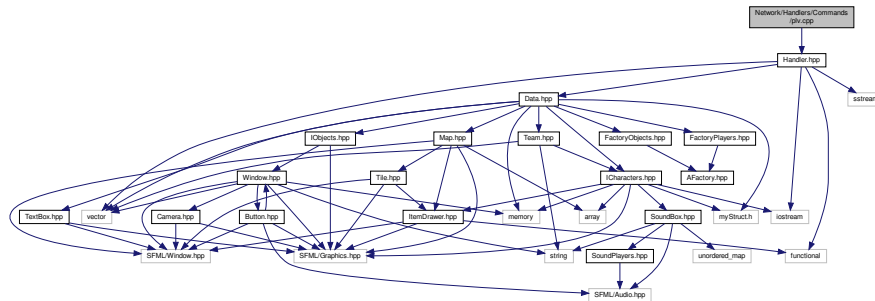
```
#include "Handler.hpp"
```

Include dependency graph for pin.cpp:



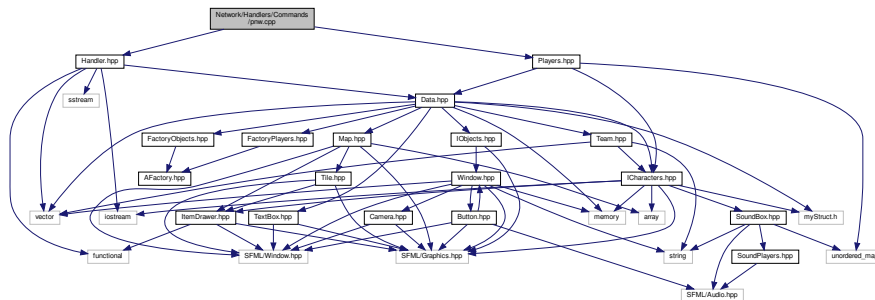
7.53 Network/Handlers/Commands/plv.cpp File Reference

```
#include "Handler.hpp"
Include dependency graph for plv.cpp:
```



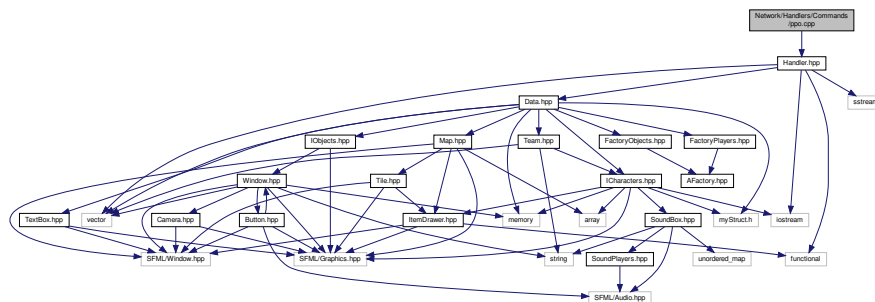
7.54 Network/Handlers/Commands/pnw.cpp File Reference

```
#include "Handler.hpp"
#include "Players.hpp"
Include dependency graph for pnw.cpp:
```



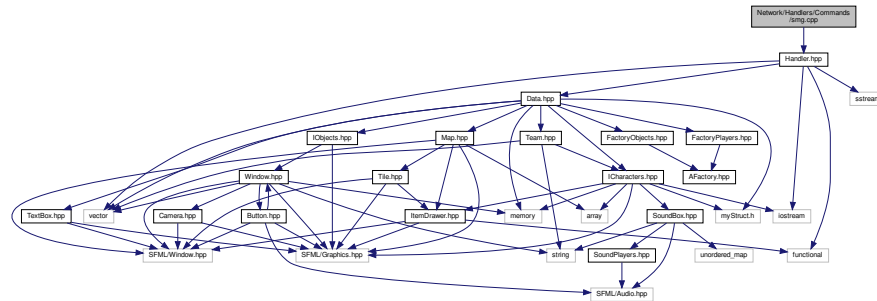
7.55 Network/Handlers/Commands/ppo.cpp File Reference

```
#include "Handler.hpp"
Include dependency graph for ppo.cpp:
```



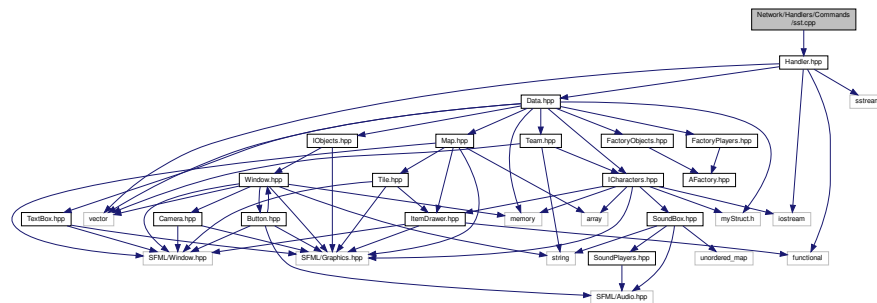
7.59 Network/Handlers/Commands/smg.cpp File Reference

```
#include "Handler.hpp"
Include dependency graph for smg.cpp:
```



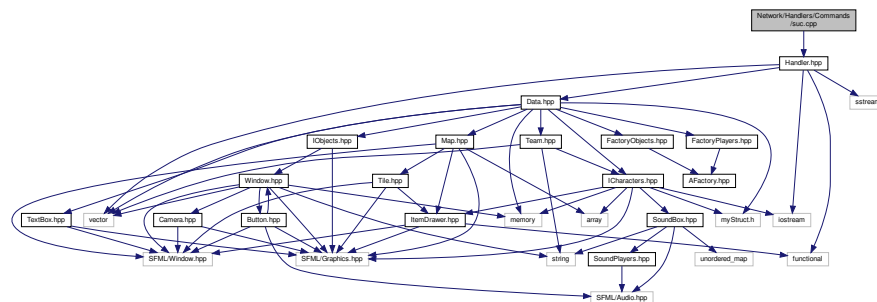
7.60 Network/Handlers/Commands/sst.cpp File Reference

```
#include "Handler.hpp"
Include dependency graph for sst.cpp:
```



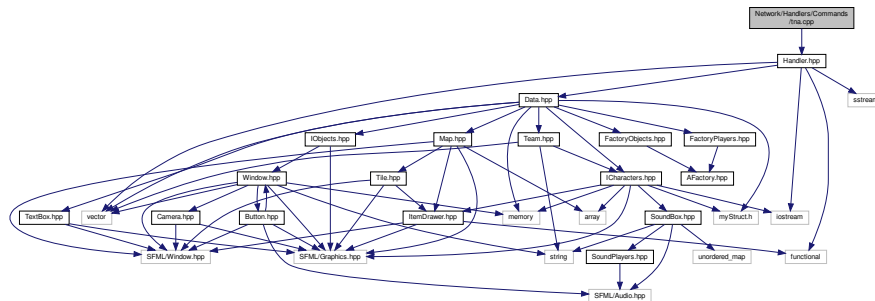
7.61 Network/Handlers/Commands/suc.cpp File Reference

```
#include "Handler.hpp"
Include dependency graph for suc.cpp:
```



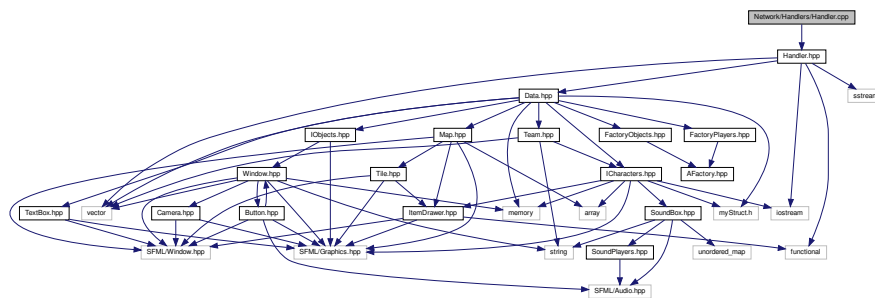
7.62 Network/Handlers/Commands/tna.cpp File Reference

```
#include "Handler.hpp"
Include dependency graph for tna.cpp:
```



7.63 Network/Handlers/Handler.cpp File Reference

```
#include "Handler.hpp"
Include dependency graph for Handler.cpp:
```



Functions

- `std::istream & operator>> (std::istream &is, Orientation &orientation)`
operator<< overload for Orientation

7.63.1 Function Documentation

7.63.1.1 operator>>()

```
std::istream& operator>> (
    std::istream & is,
    Orientation & orientation )
```

operator<< overload for Orientation

Parameters

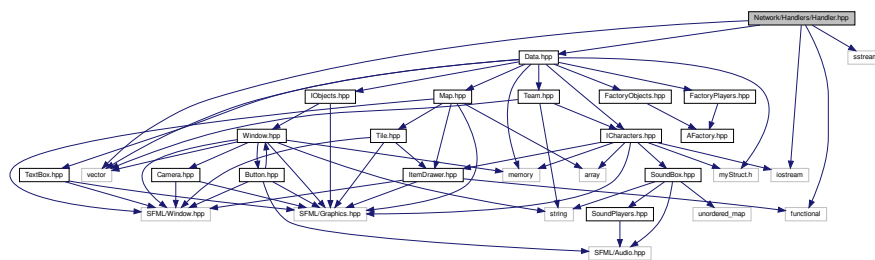
<i>is</i>	
<i>orientation</i>	

Returns

std::istream&

7.64 Network/Handlers/Handler.hpp File Reference

```
#include <vector>
#include <functional>
#include <sstream>
#include <iostream>
#include "Data.hpp"
Include dependency graph for Handler.hpp:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- struct [Command](#)
Command struct.
- class [gui::Handler](#)
Handler class.

Namespaces

- [gui](#)

Functions

- std::istream & [operator>>](#) (std::istream &is, Orientation &orientation)
operator<< overload for Orientation

7.64.1 Function Documentation

7.64.1.1 operator>>()

```
std::istream& operator>> (
    std::istream & is,
    Orientation & orientation )
```

operator<< overload for Orientation

Parameters

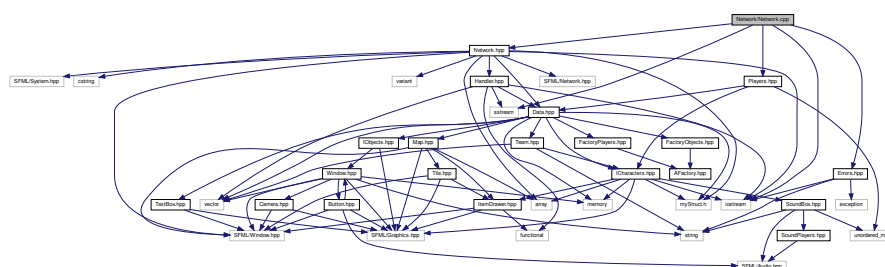
<i>is</i>	
<i>orientation</i>	

Returns

std::istream&

7.65 Network/Network.cpp File Reference

```
#include "Network.hpp"
#include <iostream>
#include <sstream>
#include "Players.hpp"
#include "Errors.hpp"
Include dependency graph for Network.cpp:
```

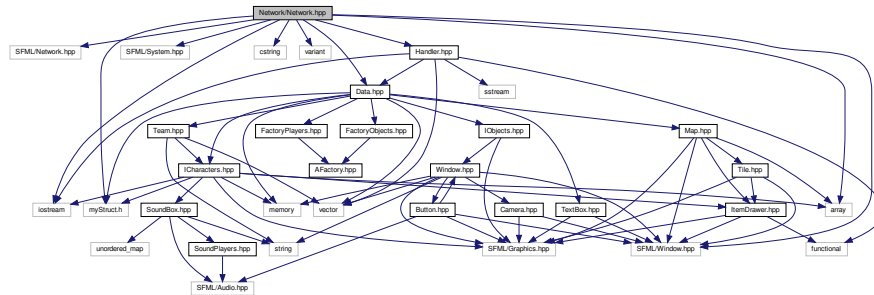


7.66 Network/Network.hpp File Reference

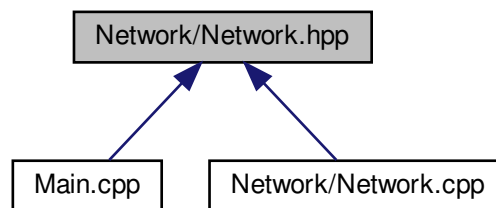
```
#include <SFML/Network.hpp>
#include <SFML/Window.hpp>
#include <SFML/System.hpp>
#include <iostream>
#include <cstring>
```

```
#include <variant>
#include "Data.hpp"
#include <array>
#include "myStruct.h"
#include "Handler.hpp"
```

Include dependency graph for Network.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class `gui::Network`
Network class.

Namespaces

- `gui`

Variables

- constexpr `std::size_t maxBufferSize = 4092`
max buffer size @const std::size_t maxBufferSize

7.66.1 Variable Documentation

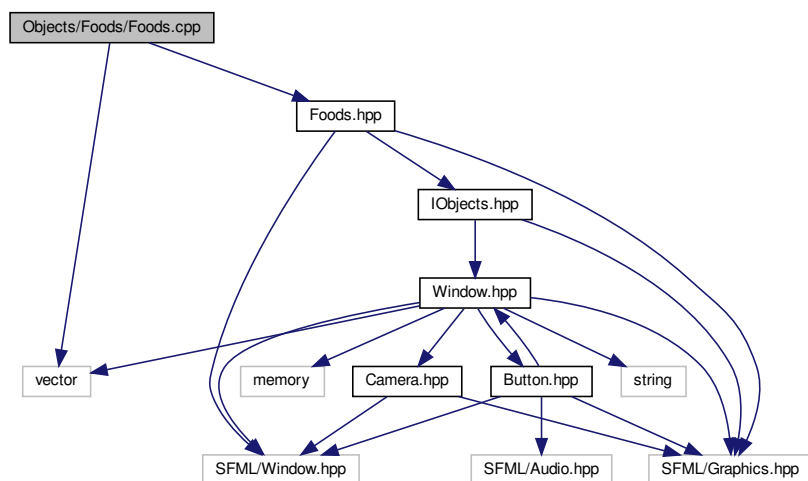
7.66.1.1 maxBufferSize

```
constexpr std::size_t maxBufferSize = 4092 [constexpr]
```

max buffer size @const std::size_t maxBufferSize

7.67 Objects/Foods/Foods.cpp File Reference

```
#include "Foods.hpp"
#include <vector>
Include dependency graph for Foods.cpp:
```



Functions

- `sf::IntRect findRect ()`

Variables

- `static std::vector< sf::IntRect > rects`

7.67.1 Function Documentation

7.67.1.1 findRect()

```
sf::IntRect findRect ( )
```

7.67.2 Variable Documentation

7.67.2.1 rects

```
std::vector<sf::IntRect> rects [static]
```

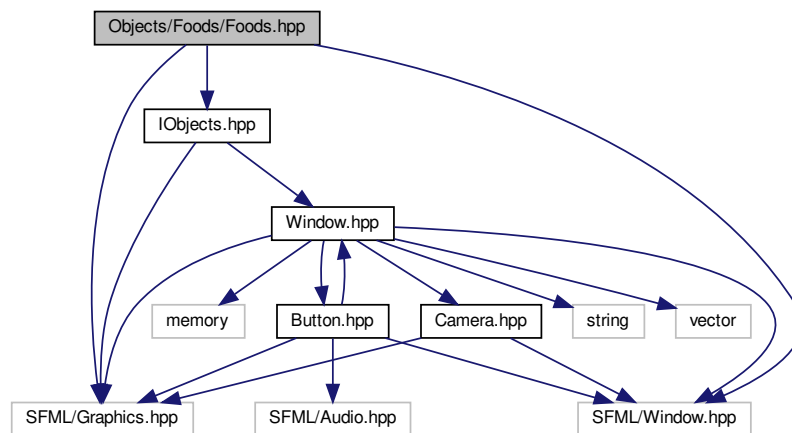
Initial value:

```
= {
    sf::IntRect(800, 1560, 75, 95),
    sf::IntRect(720, 1555, 75, 95),
```

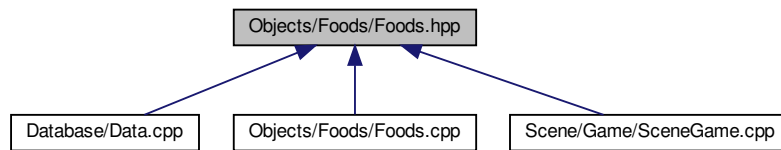
```
}
```

7.68 Objects/Foods/Foods.hpp File Reference

```
#include "IObjects.hpp"
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
Include dependency graph for Foods.hpp:
```



This graph shows which files directly or indirectly include this file:



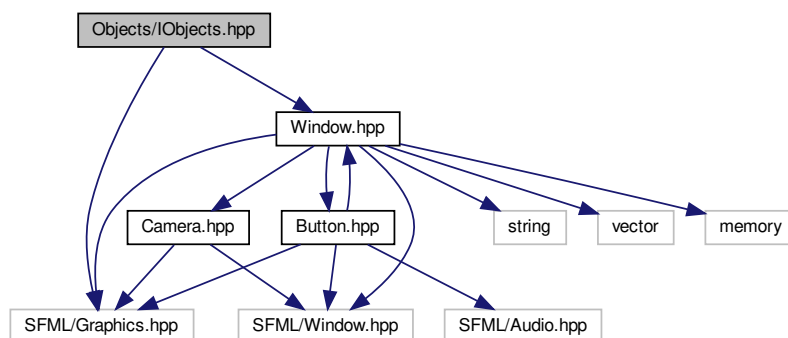
Data Structures

- class [Foods](#)
Foods class.

7.69 Objects/IObjects.hpp File Reference

```
#include "Window.hpp"
#include <SFML/Graphics.hpp>
```

Include dependency graph for IObjects.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class [IObjects](#)
IObjects class.

Enumerations

- enum class `TypeObject` { `STONES` , `FOOD` }

Variables

- constexpr char `PATH_SPRITE` [] = `"./gui/Resources/spritesheet.png"`

7.69.1 Enumeration Type Documentation

7.69.1.1 TypeObject

```
enum TypeObject [strong]
```

Enumerator

STONES	
FOOD	

7.69.2 Variable Documentation

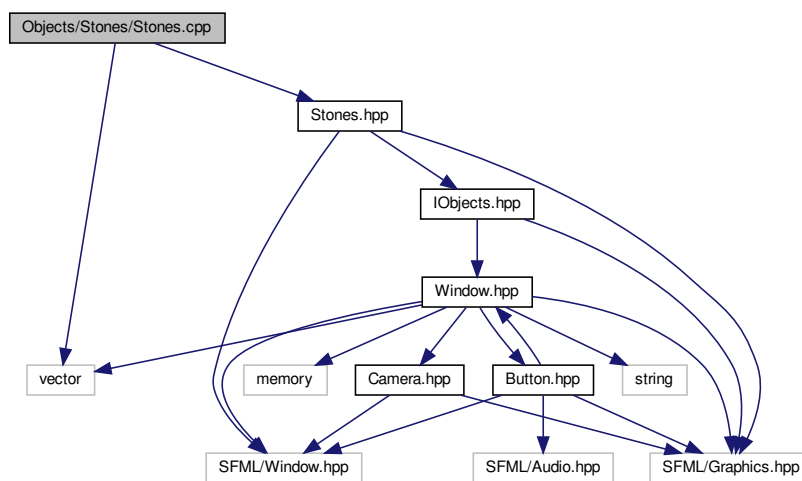
7.69.2.1 PATH_SPRITE

```
constexpr char PATH_SPRITE[] = "./gui/Resources/spritesheet.png" [constexpr]
```

7.70 Objects/Stones/Stones.cpp File Reference

```
#include "Stones.hpp"  
#include <vector>
```

Include dependency graph for Stones.cpp:



Functions

- static sf::IntRect `findRect` (`TypeStone` type)

Variables

- static std::vector< sf::IntRect > `rects`

7.70.1 Function Documentation

7.70.1.1 findRect()

```
static sf::IntRect findRect (
    TypeStone type ) [static]
```

7.70.2 Variable Documentation

7.70.2.1 rects

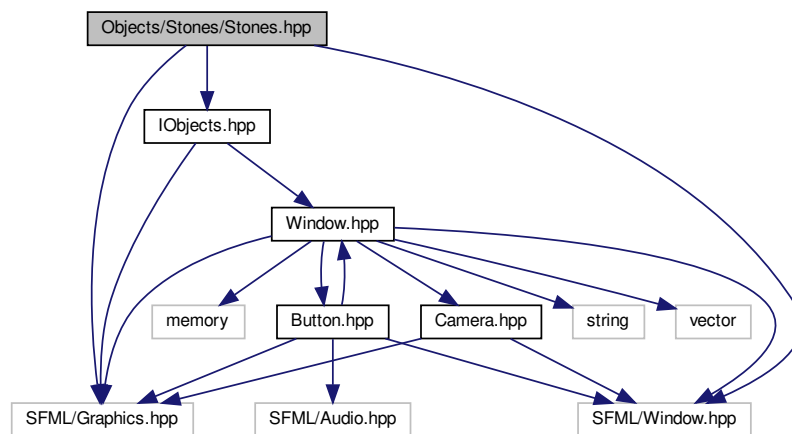
```
std::vector<sf::IntRect> rects [static]
```

Initial value:

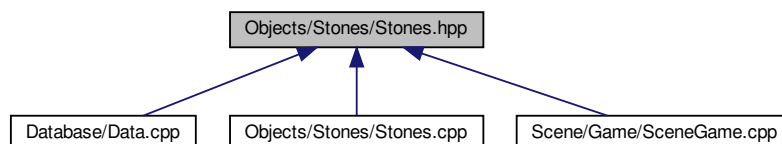
```
= {
    sf::IntRect(794, 1665, 75, 83),
    sf::IntRect(958, 1670, 75, 85),
    sf::IntRect(1200, 1670, 65, 83),
    sf::IntRect(1050, 1675, 77, 70),
    sf::IntRect(1128, 1675, 65, 75),
    sf::IntRect(875, 1660, 80, 89)
}
```

7.71 Objects/Stones/Stones.hpp File Reference

```
#include "IObjects.hpp"
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
Include dependency graph for Stones.hpp:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- class [Stones](#)
Stones class.

Enumerations

- enum class `TypeStone` {
 `LINEMATE` , `DERAUMERE` , `SIBUR` , `MENDIANE` ,
 `PHIRAS` , `THYSTAME` }

7.71.1 Enumeration Type Documentation

7.71.1.1 TypeStone

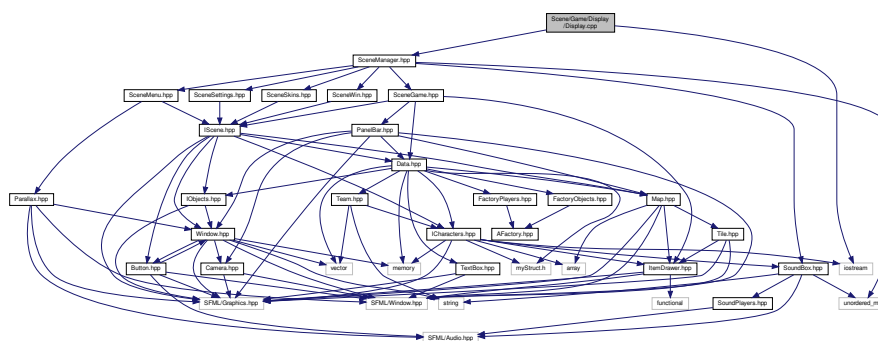
```
enum TypeStone [strong]
```

Enumerator

LINEMATE	
DERAUMERE	
SIBUR	
MENDIANE	
PHIRAS	
THYSTAME	

7.72 Scene/Game/Display/Display.cpp File Reference

```
#include "SceneManager.hpp"
#include <iostream>
Include dependency graph for Display.cpp:
```



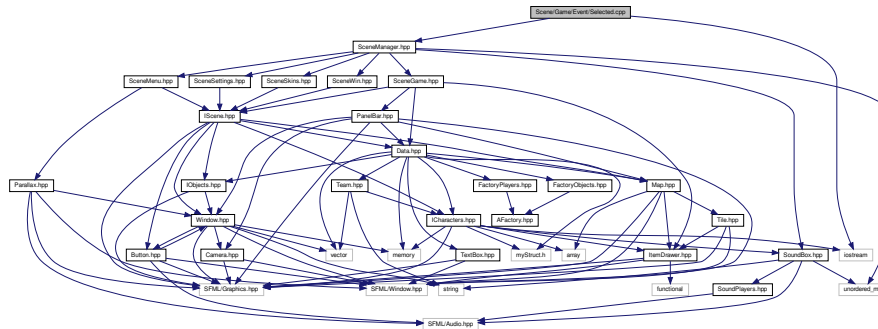
7.73 Scene/Game/Display/DisplayTile.cpp File Reference

```
#include "SceneManager.hpp"
#include "Players.hpp"
```


7.76 Scene/Game/Event/Selected.cpp File Reference

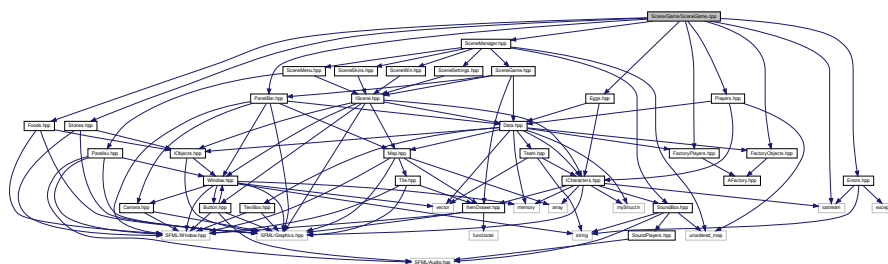
```
#include "SceneManager.hpp"
#include <iostream>
```

Include dependency graph for Selected.cpp:



7.77 Scene/Game/SceneGame.cpp File Reference

```
#include "SceneManager.hpp"
#include <iostream>
#include "FactoryObjects.hpp"
#include "FactoryPlayers.hpp"
#include "Foods.hpp"
#include "Stones.hpp"
#include "PanelBar.hpp"
#include "Players.hpp"
#include "Eggs.hpp"
#include "Errors.hpp"
Include dependency graph for SceneGame.cpp:
```



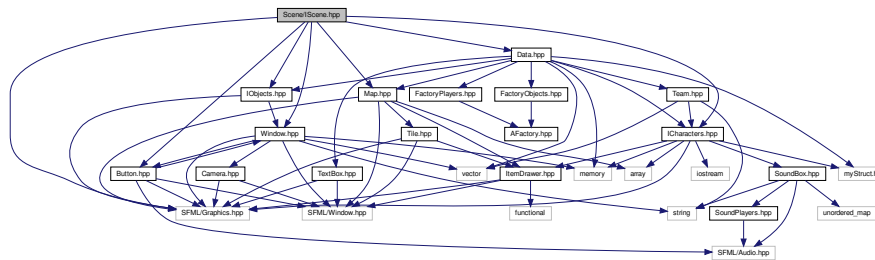
Variables

- constexpr char **PATH_GOLEM** [] = "./gui/Resources/golem_spritesheet.png"
- constexpr char **PATH_PANEL** [] = "./gui/Resources/inventory.png"

7.77.1 Variable Documentation

7.79 Scene/IScene.hpp File Reference

```
#include <SFML/Graphics.hpp>
#include "Window.hpp"
#include "Button.hpp"
#include "Map.hpp"
#include "IObjects.hpp"
#include "ICharacters.hpp"
#include "Data.hpp"
Include dependency graph for IScene.hpp:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- class **IScene**

Namespaces

- gui

7.80 Scene/Manager/SceneManager.cpp File Reference

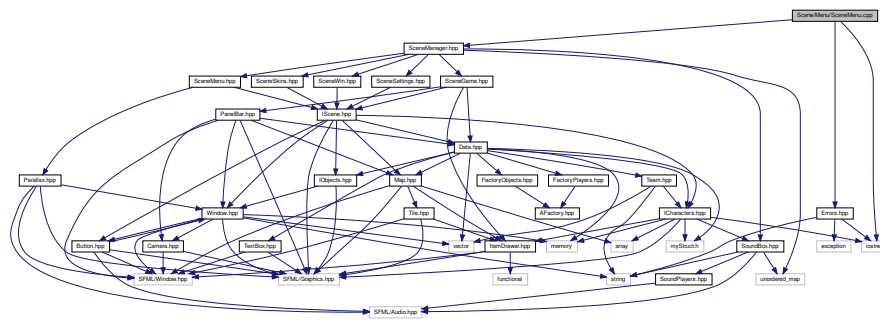
```
#include "SceneManager.hpp"
#include <iostream>
```


Namespaces

- [gui](#)

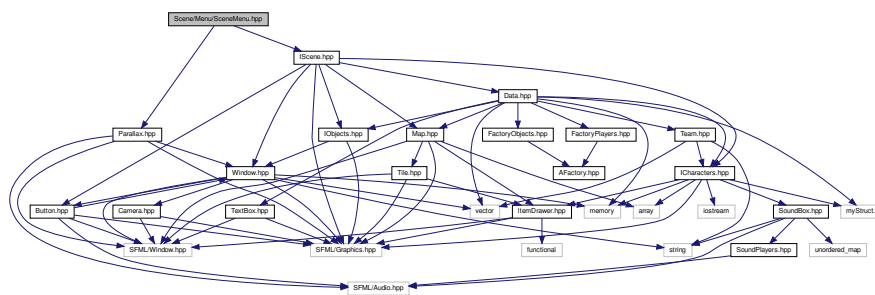
7.82 Scene/Menu/SceneManager.cpp File Reference

```
#include "SceneManager.hpp"
#include <iostream>
#include "Errors.hpp"
Include dependency graph for SceneManager.cpp:
```

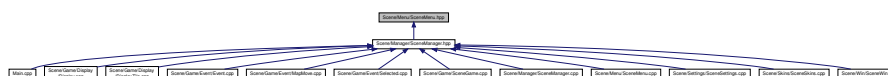


7.83 Scene/Menu/SceneManager.hpp File Reference

```
#include "IScene.hpp"
#include "Parallax.hpp"
Include dependency graph for SceneManager.hpp:
```



This graph shows which files directly or indirectly include this file:

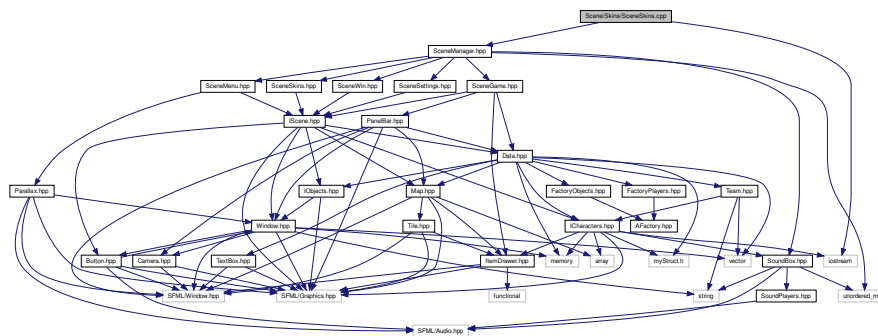


Data Structures

- class [SceneSettings](#)
Settings scene class.

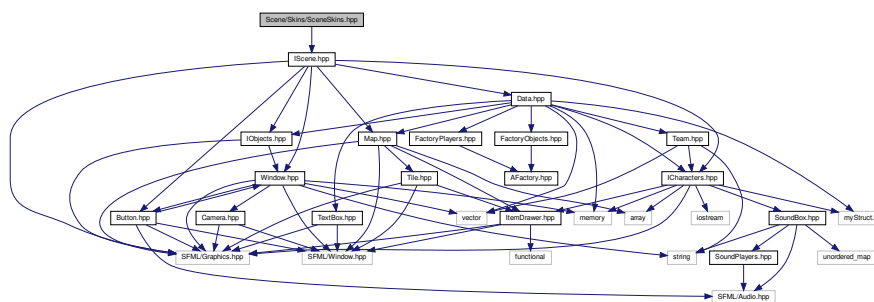
7.86 Scene/Skins/SceneSkins.cpp File Reference

```
#include <iostream>
#include "SceneManager.hpp"
Include dependency graph for SceneSkins.cpp:
```

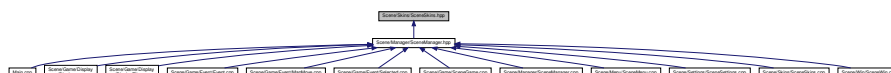


7.87 Scene/Skins/SceneSkins.hpp File Reference

```
#include "IScene.hpp"
Include dependency graph for SceneSkins.hpp:
```



This graph shows which files directly or indirectly include this file:



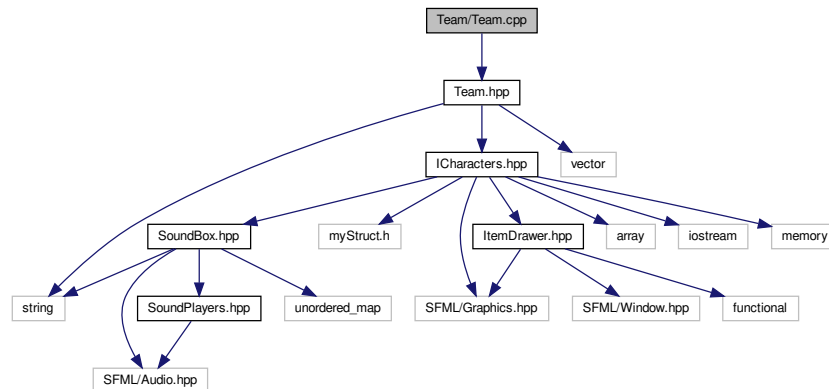
Data Structures

- class [SceneWin](#)

7.90 Team/Team.cpp File Reference

```
#include "Team.hpp"
```

Include dependency graph for Team.cpp:



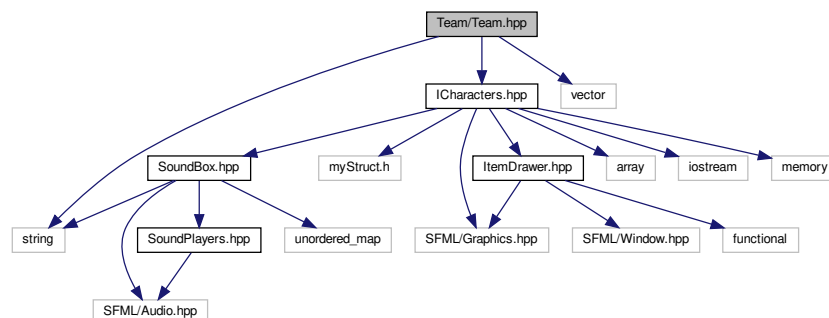
7.91 Team/Team.hpp File Reference

```
#include <string>
```

```
#include <vector>
```

```
#include "ICharacters.hpp"
```

Include dependency graph for Team.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class [gui::Team](#)
Team class.

Namespaces

- [gui](#)

Index

- ~AFactory
 - AFactory, [15](#)
- ~Arguments
 - Arguments, [18](#)
- ~Button
 - gui::Button, [26](#)
- ~Camera
 - gui::Camera, [33](#)
- ~Data
 - gui::Data, [40](#)
- ~Eggs
 - Eggs, [53](#)
- ~ErrorBroadcast
 - Errors::ErrorBroadcast, [66](#)
- ~ErrorButton
 - Errors::ErrorButton, [70](#)
- ~ErrorCharacter
 - Errors::ErrorCharacter, [74](#)
- ~ErrorColor
 - Errors::ErrorColor, [78](#)
- ~ErrorElevate
 - Errors::ErrorElevate, [82](#)
- ~ErrorEnv
 - Errors::ErrorEnv, [86](#)
- ~ErrorFile
 - Errors::ErrorFile, [90](#)
- ~ErrorFont
 - Errors::ErrorFont, [94](#)
- ~ErrorInventory
 - Errors::ErrorInventory, [98](#)
- ~ErrorOpen
 - Errors::ErrorOpen, [102](#)
- ~ErrorSocket
 - Errors::ErrorSocket, [110](#)
- ~ErrorTeam
 - Errors::ErrorTeam, [115](#)
- ~ErrorTexture
 - Errors::ErrorTexture, [119](#)
- ~ErrorUsage
 - Errors::ErrorUsage, [123](#)
- ~Errors
 - Errors, [106](#)
- ~FactoryObjects
 - FactoryObjects, [126](#)
- ~FactoryPlayers
 - FactoryPlayers, [130](#)
- ~Foods
 - Foods, [135](#)
- ~Handler
 - gui::Handler, [141](#)
- ~ICharacters
 - ICharacters, [154](#)
- ~IOObjects
 - IOObjects, [165](#)
- ~IScene
 - IScene, [167](#)
- ~ItemDrawer
 - gui::ItemDrawer, [170](#)
- ~Map
 - gui::Map, [176](#)
- ~Network
 - gui::Network, [182](#)
- ~PanelBar
 - gui::PanelBar, [189](#)
- ~Parallax
 - gui::Parallax, [197](#)
- ~Players
 - Players, [204](#)
- ~SceneGame
 - SceneGame, [220](#)
- ~SceneManager
 - gui::SceneManager, [233](#)
- ~SceneMenu
 - SceneMenu, [243](#)
- ~SceneSettings
 - SceneSettings, [250](#)
- ~SceneSkins
 - SceneSkins, [256](#)
- ~SceneWin
 - SceneWin, [263](#)
- ~SoundBox
 - SoundBox, [270](#)
- ~SoundPlayer
 - SoundPlayer, [276](#)
- ~Stones
 - Stones, [282](#)
- ~Team
 - gui::Team, [287](#)
- ~TextBox
 - TextBox, [291](#)
- ~Tile
 - gui::Tile, [299](#)
- ~Window
 - gui::Window, [309](#)
- activate
 - TextBox, [292](#)
- addCharacter
 - gui::Team, [287](#)

- AFactory, 13
 - ~AFactory, 15
 - AFactory, 14, 15
 - createObject, 15
 - operator=, 16
- alive
 - Eggs, 62
 - Players, 214
- animate
 - Eggs, 53
 - ICharacters, 155
 - Players, 205
- applyStateButton
 - gui::Button, 26
- Arguments, 17
 - ~Arguments, 18
 - Arguments, 18, 19
 - checkArgs, 19
 - checkRessources, 19
 - displayHelp, 20
 - getHostname, 20
 - getPort, 20
 - hostname, 21
 - operator=, 20
 - port, 21
 - processArguments, 21
 - resources, 21
- Arguments/Arguments.cpp, 317
- Arguments/Arguments.hpp, 317
- Background, 22
 - position, 22
 - speed, 22
 - sprite, 22
 - texture, 23
- backTile
 - gui::Map, 179
- bct.cpp
 - get_char_occurrence, 347
- bctCommand
 - gui::Handler, 142
- BLUE
 - Eggs.hpp, 321
- blueRectFunc
 - Orientation.cpp, 323
- box
 - TextBox, 296
- broadcastClock
 - Players, 214
- broadcastColor
 - Players.hpp, 326
- broadcasting
 - Eggs, 62
 - Players, 214
- buffer
 - gui::Network, 184
 - SceneSkins, 260
 - SceneWin, 266
 - SoundPlayer, 279
- bufferButton
 - gui::Button, 30
- Button
 - gui::Button, 25, 26
- Button.hpp
 - BUTTON_CLICKED, 328
 - BUTTON_HOVER, 328
 - BUTTON_IDLE, 328
 - BUTTON_LOCKED, 328
 - ButtonState, 328
- BUTTON_CLICKED
 - Button.hpp, 328
- BUTTON_HOVER
 - Button.hpp, 328
- BUTTON_IDLE
 - Button.hpp, 328
- BUTTON_LOCKED
 - Button.hpp, 328
- buttons
 - gui::SceneManager, 239
- buttonsHover
 - SceneGame, 221
- buttonsPressed
 - SceneGame, 221
- ButtonState
 - Button.hpp, 328
- buttonState
 - gui::Button, 30
- Camera
 - gui::Camera, 32, 33
- camera
 - gui::PanelBar, 193
 - SceneGame, 228
 - SceneMenu, 246
 - SceneSettings, 253
 - SceneSkins, 260
 - SceneWin, 266
- Camera/Camera.cpp, 318
- Camera/Camera.hpp, 318
- characters
 - gui::Data, 47
 - gui::Team, 289
- Characters/Eggs/Eggs.cpp, 319
- Characters/Eggs/Eggs.hpp, 320
- Characters/ICharacters.hpp, 321
- Characters/Players/Orientation.cpp, 323
- Characters/Players/Players.cpp, 325
- Characters/Players/Players.hpp, 326
- CharacterType
 - ICharacters.hpp, 322
- checkArgs
 - Arguments, 19
- checkEvents
 - IScene, 167
 - SceneGame, 222
 - SceneMenu, 244
 - SceneSettings, 251
 - SceneSkins, 257

- SceneWin, 264
- checkName
 - SceneGame, 222
- checkRessources
 - Arguments, 19
- circle
 - Players, 214
- clear
 - TextBox, 293
- clearColor
 - gui::Window, 314
- clearWindow
 - gui::Window, 310
- clock
 - Eggs, 62
 - gui::Parallax, 198
 - Players, 214
- closeEvent
 - SceneGame, 222
- color
 - Players, 215
- Command, 37
 - handler, 37
 - keyword, 37
- commands
 - gui::Handler, 150
- Commentator
 - ICharacters.hpp, 322
- Components/Buttons/Button.cpp, 327
- Components/Buttons/Button.hpp, 327
- Components/PanelBar/PanelBar.cpp, 328
- Components/PanelBar/PanelBar.hpp, 329
- Components/Parallax/Parallax.cpp, 330
- Components/Parallax/Parallax.hpp, 330
- Components/SoundBox/SoundBox.cpp, 331
- Components/SoundBox/SoundBox.hpp, 332
- Components/SoundBox/SoundPlayers.cpp, 333
- Components/SoundBox/SoundPlayers.hpp, 333
- Components/TextBox/TextBox.cpp, 334
- Components/TextBox/TextBox.hpp, 335
- Components/Window/Window.cpp, 335
- Components/Window/Window.hpp, 336
- connected
 - gui::Network, 185
- connectToServer
 - gui::Network, 182
- connexionSprite
 - SceneMenu, 246
- createButtons
 - gui::SceneManager, 234
- createCharacter
 - IScene, 167
 - SceneGame, 223
 - SceneMenu, 244
 - SceneSettings, 251
 - SceneSkins, 258
 - SceneWin, 264
- createObject
 - AFactory, 15
 - IScene, 168
 - SceneGame, 223
 - SceneMenu, 244
 - SceneSettings, 251
 - SceneSkins, 258
 - SceneWin, 264
- createObjects
 - FactoryObjects, 127
- createPlayers
 - FactoryPlayers, 131
- createSounds
 - gui::SceneManager, 234
- createText
 - gui::PanelBar, 189
- createWindow
 - gui::Window, 310
- currentPlayerIndex
 - SceneGame, 228
- currentState
 - gui::SceneManager, 239
- cursorSprite
 - gui::Window, 315
- cursorTexture
 - gui::Window, 315
- darkBlueRectFunc
 - Orientation.cpp, 323
- Data
 - gui::Data, 40
- Database/Data.cpp, 336
- Database/Data.hpp, 337
- DERAUMERE
 - Stones.hpp, 365
- descriptionText
 - TextBox, 296
- destroyWindow
 - gui::Window, 310
- directionnalButtons
 - SceneSkins, 258
- display
 - Eggs, 54
 - Foods, 136
 - gui::Map, 177
 - gui::PanelBar, 190
 - gui::Parallax, 197
 - gui::SceneManager, 234
 - ICharacters, 155
 - IOjects, 165
 - IScene, 168
 - Players, 205
 - SceneGame, 223
 - SceneMenu, 245
 - SceneSettings, 252
 - SceneSkins, 258
 - SceneWin, 265
 - Stones, 283
- displayButton
 - gui::Button, 27

- displayCharacter
 - SceneGame, 224
- displayHelp
 - Arguments, 20
- displayHostPortText
 - gui::Window, 310
- displayInventory
 - SceneGame, 224
- displayInventoryText
 - gui::PanelBar, 190
- displayPanel
 - SceneGame, 224
- displayRessourcesText
 - gui::PanelBar, 190
- displaySelected
 - SceneGame, 225
- displayTileContent
 - SceneGame, 225
- displayWindow
 - gui::Window, 311
- doAction
 - Eggs, 54
 - ICharacters, 155
 - Players, 206
- draw
 - TextBox, 293
- drawContent
 - gui::Tile, 300
- drawItem
 - gui::ItemDrawer, 171
- drawLevel
 - gui::ItemDrawer, 171
- drawMeteor
 - gui::ItemDrawer, 172
- drawObject
 - gui::Window, 311
- eboCommand
 - gui::Handler, 142
- ediCommand
 - gui::Handler, 142
- Egg
 - ICharacters.hpp, 322
- Eggs, 49
 - ~Eggs, 53
 - alive, 62
 - animate, 53
 - broadcasting, 62
 - clock, 62
 - display, 54
 - doAction, 54
 - Eggs, 52, 53
 - getColor, 54
 - getId, 55
 - getInventory, 55
 - getLevel, 55
 - getOrientation, 56
 - getPos, 56
 - getRectLeft, 56
 - getTeamName, 56
 - getType, 57
 - id, 62
 - initial, 62
 - isAlive, 57
 - isBroadcasting, 57
 - isElevating, 57
 - isLaying, 58
 - laying, 62
 - level, 63
 - operator=, 58
 - orientation, 63
 - pos, 63
 - randomColor, 59
 - rect, 63
 - setAlive, 59
 - setBroadcasting, 59
 - setElevating, 60
 - setInventory, 60
 - setLaying, 60
 - setLevel, 61
 - setPos, 61
 - setResources, 61
 - soundRadius, 63
 - teamName, 63
 - type, 63
- Eggs.cpp
 - findRectEggs, 320
 - rectEggs, 320
- Eggs.hpp
 - BLUE, 321
 - GREEN, 321
 - ICE, 321
 - POURPRE, 321
 - PURPLE, 321
 - TypeEggs, 321
 - WHITE, 321
 - YELLOW, 321
- elevateClock
 - Players, 215
- elevating
 - Players, 215
- enwCommand
 - gui::Handler, 143
- ErrorBroadcast
 - Errors::ErrorBroadcast, 66
- ErrorButton
 - Errors::ErrorButton, 70
- ErrorCharacter
 - Errors::ErrorCharacter, 74
- ErrorColor
 - Errors::ErrorColor, 77, 78
- ErrorElevate
 - Errors::ErrorElevate, 82
- ErrorEnv
 - Errors::ErrorEnv, 86
- ErrorFile
 - Errors::ErrorFile, 90

ErrorFont
 Errors::ErrorFont, 94
 ErrorInventory
 Errors::ErrorInventory, 98
 ErrorOpen
 Errors::ErrorOpen, 102
 Errors, 104
 ~Errors, 106
 Errors, 106, 107
 operator=, 107
 Errors/Errors.cpp, 338
 Errors/Errors.hpp, 338
 Errors::ErrorBroadcast, 64
 ~ErrorBroadcast, 66
 ErrorBroadcast, 66
 name, 68
 operator=, 67
 what, 67
 Errors::ErrorButton, 68
 ~ErrorButton, 70
 ErrorButton, 70
 name, 72
 operator=, 71
 what, 71
 Errors::ErrorCharacter, 72
 ~ErrorCharacter, 74
 ErrorCharacter, 74
 name, 76
 operator=, 75
 what, 75
 Errors::ErrorColor, 76
 ~ErrorColor, 78
 ErrorColor, 77, 78
 name, 80
 operator=, 79
 what, 79
 Errors::ErrorElevate, 80
 ~ErrorElevate, 82
 ErrorElevate, 82
 name, 84
 operator=, 83
 what, 83
 Errors::ErrorEnv, 84
 ~ErrorEnv, 86
 ErrorEnv, 86
 name, 88
 operator=, 87
 what, 87
 Errors::ErrorFile, 88
 ~ErrorFile, 90
 ErrorFile, 90
 name, 92
 operator=, 91
 what, 91
 Errors::ErrorFont, 92
 ~ErrorFont, 94
 ErrorFont, 94
 name, 96
 operator=, 95
 what, 95
 Errors::ErrorInventory, 96
 ~ErrorInventory, 98
 ErrorInventory, 98
 name, 100
 operator=, 99
 what, 99
 Errors::ErrorOpen, 100
 ~ErrorOpen, 102
 ErrorOpen, 102
 name, 104
 operator=, 103
 what, 103
 Errors::ErrorSocket, 108
 ~ErrorSocket, 110
 ErrorSocket, 110
 name, 112
 operator=, 111
 what, 111
 Errors::ErrorTeam, 113
 ~ErrorTeam, 115
 ErrorTeam, 115
 name, 117
 operator=, 116
 what, 116
 Errors::ErrorTexture, 117
 ~ErrorTexture, 119
 ErrorTexture, 119
 name, 121
 operator=, 120
 what, 120
 Errors::ErrorUsage, 121
 ~ErrorUsage, 123
 ErrorUsage, 123
 name, 125
 operator=, 124
 what, 124
 ErrorSocket
 Errors::ErrorSocket, 110
 ErrorT, 112
 ErrorTeam
 Errors::ErrorTeam, 115
 ErrorTexture
 Errors::ErrorTexture, 119
 ErrorUsage
 Errors::ErrorUsage, 123
 event
 gui::Window, 315
 facingRectColor
 gui::PanelBar, 193
 Factory/AFactory.hpp, 339
 Factory/FactoryObjects/FactoryObjects.hpp, 340
 Factory/FactoryPlayers/FactoryPlayers.hpp, 340
 factoryCharacters
 gui::Data, 47
 FactoryObjects, 125
 ~FactoryObjects, 126

- createObjects, 127
- FactoryObjects, 126, 127
- operator=, 127
- factoryObjects
 - gui::Data, 47
- FactoryPlayers, 128
 - ~FactoryPlayers, 130
 - createPlayers, 131
 - FactoryPlayers, 130
 - operator=, 131
- findRect
 - Foods.cpp, 359
 - Stones.cpp, 363
- findRectEggs
 - Eggs.cpp, 320
- finished
 - TextBox, 296
- font
 - gui::PanelBar, 194
 - gui::Window, 315
 - SceneWin, 266
- FOOD
 - IObjects.hpp, 362
- foodClock
 - Players, 215
- Foods, 133
 - ~Foods, 135
 - display, 136
 - Foods, 135, 136
 - getType, 136
 - operator=, 137
 - pos, 137
 - rect, 137
 - type, 138
- Foods.cpp
 - findRect, 359
 - rects, 360
- fRect
 - gui::Tile, 305
- get_char_occurrence
 - bct.cpp, 347
- getBackground
 - gui::Window, 311
- getBuffer
 - SoundPlayer, 277
- getButton
 - gui::SceneManager, 234
- getButtons
 - gui::SceneManager, 235
- getButtonState
 - gui::Button, 27
- getCamera
 - gui::PanelBar, 192
 - gui::Window, 311
 - SceneGame, 225
- getCenter
 - gui::Camera, 33
- getCharacterById
 - gui::Data, 41
- getCharacters
 - gui::Data, 41
 - gui::Team, 288
- getColor
 - Eggs, 54
 - ICharacters, 155
 - Players, 206
- getEvent
 - gui::Window, 311
- getFacingRectByColor
 - Orientation.cpp, 323
 - PanelBar.cpp, 329
- getFactoryCharacters
 - gui::Data, 41
- getFRect
 - gui::Tile, 301
- getFrequency
 - gui::Data, 42
- getHostname
 - Arguments, 20
- getId
 - Eggs, 55
 - ICharacters, 156
 - Players, 206
- getInventory
 - Eggs, 55
 - ICharacters, 156
 - Players, 206
- getIsPlaying
 - SoundPlayer, 277
- getIsSelected
 - gui::Tile, 301
- getLevel
 - Eggs, 55
 - ICharacters, 156
 - Players, 207
- getMap
 - gui::Data, 42
- getMapFrame
 - gui::Data, 42
- getMapFrameSprite
 - gui::SceneManager, 235
- getMousePos
 - gui::Window, 312
- getName
 - gui::Button, 27
 - gui::Team, 288
- getOrientation
 - Eggs, 56
 - ICharacters, 156
 - Players, 207
- getPort
 - Arguments, 20
- getPos
 - Eggs, 56
 - gui::Button, 27
 - ICharacters, 157

- Players, 207
- getPosition
 - gui::Tile, 301
 - TextBox, 293
- getRect
 - gui::Tile, 301
- getRectButton
 - gui::Button, 27
- getRectLeft
 - Eggs, 56
 - ICharacters, 157
 - Players, 207
- getResources
 - gui::Tile, 301
- getRessourcesText
 - gui::PanelBar, 192
- getRightRect
 - Orientation.cpp, 324
 - Players.cpp, 325
- getShape
 - gui::Tile, 302
- getSizeButton
 - gui::Button, 27
- getSound
 - SoundPlayer, 277
- getSoundBox
 - gui::SceneManager, 235
- getSoundStatus
 - SoundBox, 271
- getSprite
 - gui::SceneManager, 235
- getString
 - TextBox, 293
- getTeamByName
 - gui::Data, 42
- getTeamName
 - Eggs, 56
 - ICharacters, 157
 - Players, 208
- getTeamNameById
 - gui::Data, 43
- getTeams
 - gui::Data, 43
- getTextBox
 - gui::Data, 43
- getTiles
 - gui::Map, 177
- getTileSprite
 - gui::SceneManager, 235
- getTilesRect
 - gui::Data, 44
- getType
 - Eggs, 57
 - Foods, 136
 - ICharacters, 157
 - IObjects, 165
 - Players, 208
 - Stones, 283
- getView
 - gui::Camera, 34
- getVolSprite
 - gui::SceneManager, 236
- getVolume
 - gui::SceneManager, 236
 - SoundBox, 271
- getWindow
 - gui::Window, 312
- getWinningTeam
 - gui::Data, 44
- getZoom
 - gui::Camera, 34
- GREEN
 - Eggs.hpp, 321
- greenRectFunc
 - Orientation.cpp, 324
- gui, 11
- gui::Button, 23
 - ~Button, 26
 - applyStateButton, 26
 - bufferButton, 30
 - Button, 25, 26
 - buttonState, 30
 - displayButton, 27
 - getButtonState, 27
 - getName, 27
 - getPos, 27
 - getRectButton, 27
 - getSizeButton, 27
 - hoverHandler, 27
 - isButtonPressed, 28
 - lockStateButton, 28
 - nameButton, 30
 - operator=, 28
 - playSoundButton, 29
 - posButton, 30
 - rectButton, 30
 - setButtonState, 29
 - setPos, 29
 - setShapeButton, 29
 - shape, 30
 - shapeButton, 30
 - shapePos, 30
 - soundButton, 31
 - stopSoundButton, 29
- gui::Camera, 31
 - ~Camera, 33
 - Camera, 32, 33
 - getCenter, 33
 - getView, 34
 - getZoom, 34
 - operator=, 34
 - setCenter, 35
 - setSize, 35
 - setView, 35
 - setZoom, 36
 - view, 36

- zoomFactor, 36
- gui::Data, 38
 - ~Data, 40
 - characters, 47
 - Data, 40
 - factoryCharacters, 47
 - factoryObjects, 47
 - getCharacterById, 41
 - getCharacters, 41
 - getFactoryCharacters, 41
 - getFrequency, 42
 - getMap, 42
 - getMapFrame, 42
 - getTeamByName, 42
 - getTeamNameById, 43
 - getTeams, 43
 - getTextBox, 43
 - getTilesRect, 44
 - getWinningTeam, 44
 - isPlayer, 44
 - isTeam, 45
 - map, 47
 - mapFrame, 47
 - operator=, 45
 - setCharacters, 45
 - setFrequency, 46
 - setMapFrame, 46
 - setTilesRect, 46
 - setWinningTeam, 46
 - teams, 47
 - textBox, 48
 - tilesRect, 48
 - timeUnit, 48
 - winningTeam, 48
- gui::Handler, 138
 - ~Handler, 141
 - bctCommand, 142
 - commands, 150
 - eboCommand, 142
 - ediCommand, 142
 - enwCommand, 143
 - handle, 143
 - Handler, 141
 - mszCommand, 143
 - operator=, 144
 - pbcCommand, 144
 - pdiCommand, 144
 - pdrCommand, 145
 - pexCommand, 145
 - pfkCommand, 145
 - pgtCommand, 146
 - picCommand, 146
 - pieCommand, 146
 - pinCommand, 147
 - plvCommand, 147
 - pnwCommand, 147
 - ppoCommand, 148
 - sbpCommand, 148
 - segCommand, 148
 - sgtCommand, 148
 - smgCommand, 149
 - sstCommand, 149
 - sucCommand, 149
 - tnaCommand, 150
- gui::ItemDrawer, 169
 - ~ItemDrawer, 170
 - drawItem, 171
 - drawLevel, 171
 - drawMeteor, 172
 - ItemDrawer, 170
 - itemRect, 173
 - itemSprite, 174
 - levelRect, 174
 - meteorRect, 174
 - operator=, 172
 - setSprite, 173
- gui::Map, 175
 - ~Map, 176
 - backTile, 179
 - display, 177
 - getTiles, 177
 - height, 179
 - initMap, 177
 - Map, 176
 - operator=, 178
 - setTileSkin, 178
 - texture, 179
 - tiles, 179
 - width, 179
- gui::Network, 180
 - ~Network, 182
 - buffer, 184
 - connected, 185
 - connectToServer, 182
 - handler, 185
 - isConnected, 182
 - Network, 181, 182
 - operator=, 183
 - parseCommand, 183
 - processData, 183
 - receivedSize, 185
 - run, 184
 - sendCommand, 184
 - serverIP, 185
 - serverPort, 185
 - socket, 185
- gui::PanelBar, 186
 - ~PanelBar, 189
 - camera, 193
 - createText, 189
 - display, 190
 - displayInventoryText, 190
 - displayResourcesText, 190
 - facingRectColor, 193
 - font, 194
 - getCamera, 192

- getRessourcesText, 192
- operator=, 192
- PanelBar, 187, 189
- playerLevel, 194
- pos, 194
- resourcesInvText, 194
- resourcesText, 194
- setInventoryText, 193
- setRessourcesText, 193
- gui::Parallax, 195
 - ~Parallax, 197
 - clock, 198
 - display, 197
 - load, 197
 - move, 197
 - operator=, 197, 198
 - Parallax, 196
 - parallax, 198
 - reset, 198
- gui::SceneManager, 230
 - ~SceneManager, 233
 - buttons, 239
 - createButtons, 234
 - createSounds, 234
 - currentState, 239
 - display, 234
 - getButton, 234
 - getButtons, 235
 - getMapFrameSprite, 235
 - getSoundBox, 235
 - getSprite, 235
 - getTileSprite, 235
 - getVolSprite, 236
 - getVolume, 236
 - loadScene, 236
 - mapFrameSprite, 239
 - mapFrameTexture, 239
 - operator=, 236, 237
 - SceneManager, 233
 - setMapFrameSprite, 237
 - setState, 237
 - setTileSprite, 238
 - setVolume, 238
 - soundbox, 239
 - sprite, 239
 - texture, 239
 - tileSprite, 239
 - tileTexture, 240
 - update, 238
 - volSprite, 240
- gui::Team, 285
 - ~Team, 287
 - addCharacter, 287
 - characters, 289
 - getCharacters, 288
 - getName, 288
 - name, 289
 - operator=, 288
 - setName, 289
 - Team, 286, 287
- gui::Tile, 297
 - ~Tile, 299
 - drawContent, 300
 - fRect, 305
 - getFRect, 301
 - getIsSelected, 301
 - getPosition, 301
 - getRect, 301
 - getResources, 301
 - getShape, 302
 - isSelected, 305
 - operator=, 302
 - pos, 305
 - rect, 305
 - resources, 305
 - setFRect, 303
 - setIsSelected, 303
 - setPosition, 303
 - setRect, 303
 - setResources, 304
 - setShape, 304
 - shape, 305
 - Tile, 299, 300
- gui::Window, 306
 - ~Window, 309
 - clearColor, 314
 - clearWindow, 310
 - createWindow, 310
 - cursorSprite, 315
 - cursorTexture, 315
 - destroyWindow, 310
 - displayHostPortText, 310
 - displayWindow, 311
 - drawObject, 311
 - event, 315
 - font, 315
 - getBackground, 311
 - getCamera, 311
 - getEvent, 311
 - getMousePos, 312
 - getWindow, 312
 - hostname, 315
 - hostText, 315
 - isOpenWindow, 312
 - mousePos, 315
 - operator=, 312, 313
 - port, 315
 - portText, 316
 - setBackground, 313
 - setClearColor, 313
 - setCursor, 314
 - setCursorPos, 314
 - setMousePos, 314
 - sprite, 316
 - texture, 316
 - Window, 309

- window, 316
- handle
 - gui::Handler, 143
- Handler
 - gui::Handler, 141
- handler
 - Command, 37
 - gui::Network, 185
- Handler.cpp
 - operator>>, 355
- Handler.hpp
 - operator>>, 357
- haswonText
 - SceneWin, 267
- height
 - gui::Map, 179
- hostname
 - Arguments, 21
 - gui::Window, 315
- hostText
 - gui::Window, 315
- hoverHandler
 - gui::Button, 27
- ICE
 - Eggs.hpp, 321
- ICharacters, 151
 - ~ICharacters, 154
 - animate, 155
 - display, 155
 - doAction, 155
 - getColor, 155
 - getId, 156
 - getInventory, 156
 - getLevel, 156
 - getOrientation, 156
 - getPos, 157
 - getRectLeft, 157
 - getTeamName, 157
 - getType, 157
 - isAlive, 158
 - isBroadcasting, 158
 - isElevating, 158
 - isLaying, 159
 - setAlive, 159
 - setBroadcasting, 159
 - setElevating, 160
 - setInventory, 160
 - setLaying, 160
 - setLevel, 161
 - setPos, 161
 - setResources, 161
- ICharacters.hpp
 - CharacterType, 322
 - Commentator, 322
 - Egg, 322
 - Player, 322
- id
 - Eggs, 62
 - Players, 215
- Infos, 162
- infos, 162
 - obj, 163
 - pos, 163
 - spriteld, 163
- init
 - SoundPlayer, 278
- initial
 - Eggs, 62
 - Players, 215
- initMap
 - gui::Map, 177
- initSong
 - SoundBox, 272
- initTextBox
 - TextBox, 293
- inputMode
 - TextBox, 296
- inputText
 - TextBox, 297
- inventory
 - Players, 215
- IOObjects, 163
 - ~IOObjects, 165
 - display, 165
 - getType, 165
- IOObjects.hpp
 - FOOD, 362
 - PATH_SPRITE, 362
 - STONES, 362
 - TypeObject, 362
- isAlive
 - Eggs, 57
 - ICharacters, 158
 - Players, 208
- isBroadcasting
 - Eggs, 57
 - ICharacters, 158
 - Players, 208
- isButtonPressed
 - gui::Button, 28
- IScene, 166
 - ~IScene, 167
 - checkEvents, 167
 - createCharacter, 167
 - createObject, 168
 - display, 168
 - loadScene, 168
- isConnected
 - gui::Network, 182
- isElevating
 - Eggs, 57
 - ICharacters, 158
 - Players, 209
- isFinished
 - TextBox, 294

- isInputMode
 - TextBox, 294
- isLaying
 - Eggs, 58
 - ICharacters, 159
 - Players, 209
- isOpenWindow
 - gui::Window, 312
- isPlayer
 - gui::Data, 44
- isPlaying
 - SoundPlayer, 279
- isPressed
 - TextBox, 294
- isSelected
 - gui::Tile, 305
- isTeam
 - gui::Data, 45
- ItemDrawer
 - gui::ItemDrawer, 170
- itemDrawer
 - SceneGame, 228
- ItemDrawer/ItemDrawer.cpp, 341
- ItemDrawer/ItemDrawer.hpp, 341
- itemRect
 - gui::ItemDrawer, 173
- itemSprite
 - gui::ItemDrawer, 174
- keyword
 - Command, 37
- launch
 - Main.cpp, 342
- laying
 - Eggs, 62
 - Players, 215
- level
 - Eggs, 63
 - Players, 216
- levelRect
 - gui::ItemDrawer, 174
- LINEMATE
 - Stones.hpp, 365
- load
 - gui::Parallax, 197
- loadScene
 - gui::SceneManager, 236
 - IScene, 168
 - SceneGame, 225
 - SceneMenu, 245
 - SceneSettings, 252
 - SceneSkins, 259
 - SceneWin, 265
- lockStateButton
 - gui::Button, 28
- logo
 - SceneMenu, 247
- main
 - Main.cpp, 343
- Main.cpp, 342
 - launch, 342
 - main, 343
- Map
 - gui::Map, 176
- map
 - gui::Data, 47
- Map.hpp
 - PATH_BACK, 345
 - RECT_GRASS, 345
- Map/Map.cpp, 343
- Map/Map.hpp, 344
- Map/Tile.cpp, 345
- Map/Tile.hpp, 346
- mapFrame
 - gui::Data, 47
 - SceneGame, 228
- mapFrameSprite
 - gui::SceneManager, 239
- mapFrameTexture
 - gui::SceneManager, 239
- mapMove
 - SceneGame, 226
- maxBufferSize
 - Network.hpp, 359
- MENDIANE
 - Stones.hpp, 365
- meteorRect
 - gui::ItemDrawer, 174
- mousePos
 - gui::Window, 315
- move
 - gui::Parallax, 197
- mszCommand
 - gui::Handler, 143
- name
 - Errors::ErrorBroadcast, 68
 - Errors::ErrorButton, 72
 - Errors::ErrorCharacter, 76
 - Errors::ErrorColor, 80
 - Errors::ErrorElevate, 84
 - Errors::ErrorEnv, 88
 - Errors::ErrorFile, 92
 - Errors::ErrorFont, 96
 - Errors::ErrorInventory, 100
 - Errors::ErrorOpen, 104
 - Errors::ErrorSocket, 112
 - Errors::ErrorTeam, 117
 - Errors::ErrorTexture, 121
 - Errors::ErrorUsage, 125
 - gui::Team, 289
 - Song, 268
- nameButton
 - gui::Button, 30
- Network
 - gui::Network, 181, 182

- Network.hpp
 - maxBufferSize, 359
- Network/Handlers/Commands/bct.cpp, 347
- Network/Handlers/Commands/ebo.cpp, 347
- Network/Handlers/Commands/edi.cpp, 348
- Network/Handlers/Commands/enw.cpp, 348
- Network/Handlers/Commands/msz.cpp, 348
- Network/Handlers/Commands/pbc.cpp, 349
- Network/Handlers/Commands/pdi.cpp, 349
- Network/Handlers/Commands/pdr.cpp, 349
- Network/Handlers/Commands/pex.cpp, 350
- Network/Handlers/Commands/pfk.cpp, 350
- Network/Handlers/Commands/pgt.cpp, 350
- Network/Handlers/Commands/pic.cpp, 351
- Network/Handlers/Commands/pie.cpp, 351
- Network/Handlers/Commands/pin.cpp, 351
- Network/Handlers/Commands/plv.cpp, 352
- Network/Handlers/Commands/pnw.cpp, 352
- Network/Handlers/Commands/ppo.cpp, 352
- Network/Handlers/Commands/sbp.cpp, 353
- Network/Handlers/Commands/seg.cpp, 353
- Network/Handlers/Commands/sgt.cpp, 353
- Network/Handlers/Commands/msg.cpp, 354
- Network/Handlers/Commands/sst.cpp, 354
- Network/Handlers/Commands/suc.cpp, 354
- Network/Handlers/Commands/tna.cpp, 355
- Network/Handlers/Handler.cpp, 355
- Network/Handlers/Handler.hpp, 356
- Network/Network.cpp, 357
- Network/Network.hpp, 357
- obj
 - infos, 163
- Objects/Foods/Foods.cpp, 359
- Objects/Foods/Foods.hpp, 360
- Objects/IObjects.hpp, 361
- Objects/Stones/Stones.cpp, 362
- Objects/Stones/Stones.hpp, 364
- operator>>
 - Handler.cpp, 355
 - Handler.hpp, 357
- operator=
 - AFactory, 16
 - Arguments, 20
 - Eggs, 58
 - Errors, 107
 - Errors::ErrorBroadcast, 67
 - Errors::ErrorButton, 71
 - Errors::ErrorCharacter, 75
 - Errors::ErrorColor, 79
 - Errors::ErrorElevate, 83
 - Errors::ErrorEnv, 87
 - Errors::ErrorFile, 91
 - Errors::ErrorFont, 95
 - Errors::ErrorInventory, 99
 - Errors::ErrorOpen, 103
 - Errors::ErrorSocket, 111
 - Errors::ErrorTeam, 116
 - Errors::ErrorTexture, 120
 - Errors::ErrorUsage, 124
 - FactoryObjects, 127
 - FactoryPlayers, 131
 - Foods, 137
 - gui::Button, 28
 - gui::Camera, 34
 - gui::Data, 45
 - gui::Handler, 144
 - gui::ItemDrawer, 172
 - gui::Map, 178
 - gui::Network, 183
 - gui::PanelBar, 192
 - gui::Parallax, 197, 198
 - gui::SceneManager, 236, 237
 - gui::Team, 288
 - gui::Tile, 302
 - gui::Window, 312, 313
 - Players, 209, 210
 - SceneGame, 226
 - SceneMenu, 246
 - SceneSettings, 252, 253
 - SceneSkins, 259
 - SceneWin, 265, 266
 - SoundBox, 272
 - SoundPlayer, 278
 - Stones, 283, 284
 - TextBox, 295
- orientation
 - Eggs, 63
 - Players, 216
- Orientation.cpp
 - blueRectFunc, 323
 - darkBlueRectFunc, 323
 - getFacingRectByColor, 323
 - getRightRect, 324
 - greenRectFunc, 324
 - purpleRectFunc, 324
 - redRectFunc, 324
 - yellowRectFunc, 324
- PanelBar
 - gui::PanelBar, 187, 189
- PanelBar.cpp
 - getFacingRectByColor, 329
- panelBars
 - SceneGame, 228
- panelBarVisible
 - SceneGame, 228
- Parallax
 - gui::Parallax, 196
- parallax
 - gui::Parallax, 198
 - SceneMenu, 247
- parseCommand
 - gui::Network, 183
- PATH_BACK
 - Map.hpp, 345
- PATH_GOLEM
 - Players.cpp, 325

- SceneGame.cpp, 367
- PATH_PANEL
 - SceneGame.cpp, 368
- PATH_SPRITE
 - IObjects.hpp, 362
- pbCommand
 - gui::Handler, 144
- pdiCommand
 - gui::Handler, 144
- pdrCommand
 - gui::Handler, 145
- pexCommand
 - gui::Handler, 145
- pfkCommand
 - gui::Handler, 145
- pgtCommand
 - gui::Handler, 146
- PHIRAS
 - Stones.hpp, 365
- picCommand
 - gui::Handler, 146
- pieCommand
 - gui::Handler, 146
- pinCommand
 - gui::Handler, 147
- play
 - SoundBox, 274
- Player
 - ICharacters.hpp, 322
- playerLevel
 - gui::PanelBar, 194
- Players, 199
 - ~Players, 204
 - alive, 214
 - animate, 205
 - broadcastClock, 214
 - broadcasting, 214
 - circle, 214
 - clock, 214
 - color, 215
 - display, 205
 - doAction, 206
 - elevateClock, 215
 - elevating, 215
 - foodClock, 215
 - getColor, 206
 - getId, 206
 - getInventory, 206
 - getLevel, 207
 - getOrientation, 207
 - getPos, 207
 - getRectLeft, 207
 - getTeamName, 208
 - getType, 208
 - id, 215
 - initial, 215
 - inventory, 215
 - isAlive, 208
 - isBroadcasting, 208
 - isElevating, 209
 - isLaying, 209
 - laying, 215
 - level, 216
 - operator=, 209, 210
 - orientation, 216
 - Players, 203–205
 - pos, 216
 - randomColor, 210
 - rect, 216
 - setAlive, 210
 - setBroadcasting, 212
 - setElevating, 212
 - setInventory, 212
 - setLaying, 213
 - setLevel, 213
 - setPos, 213
 - setResources, 214
 - soundRadius, 216
 - teamName, 216
- Players.cpp
 - getRightRect, 325
 - PATH_GOLEM, 325
- Players.hpp
 - broadcastColor, 326
- playSoundButton
 - gui::Button, 29
- plvCommand
 - gui::Handler, 147
- pnwCommand
 - gui::Handler, 147
- port
 - Arguments, 21
 - gui::Window, 315
- portText
 - gui::Window, 316
- pos
 - Eggs, 63
 - Foods, 137
 - gui::PanelBar, 194
 - gui::Tile, 305
 - infos, 163
 - Players, 216
 - Stones, 284
- posButton
 - gui::Button, 30
- position
 - Background, 22
- POURPRE
 - Eggs.hpp, 321
- ppoCommand
 - gui::Handler, 148
- previousSelected
 - SceneGame, 227
- processArguments
 - Arguments, 21
- processData

- gui::Network, [183](#)
- PURPLE
 - Eggs.hpp, [321](#)
- purpleRectFunc
 - Orientation.cpp, [324](#)
- randomColor
 - Eggs, [59](#)
 - Players, [210](#)
- receivedSize
 - gui::Network, [185](#)
- rect
 - Eggs, [63](#)
 - Foods, [137](#)
 - gui::Tile, [305](#)
 - Players, [216](#)
 - Stones, [284](#)
 - TextBox, [297](#)
- RECT_GRASS
 - Map.hpp, [345](#)
- rectButton
 - gui::Button, [30](#)
- rectEggs
 - Eggs.cpp, [320](#)
- rects
 - Foods.cpp, [360](#)
 - Stones.cpp, [363](#)
- redRectFunc
 - Orientation.cpp, [324](#)
- removeCharacter
 - SceneGame, [227](#)
- reset
 - gui::Parallax, [198](#)
- resources
 - Arguments, [21](#)
 - gui::Tile, [305](#)
- resourcesInvText
 - gui::PanelBar, [194](#)
- resourcesText
 - gui::PanelBar, [194](#)
- run
 - gui::Network, [184](#)
- sbpCommand
 - gui::Handler, [148](#)
- Scene/Game/Display/Display.cpp, [365](#)
- Scene/Game/Display/DisplayTile.cpp, [365](#)
- Scene/Game/Event/Event.cpp, [366](#)
- Scene/Game/Event/MapMove.cpp, [366](#)
- Scene/Game/Event/Selected.cpp, [367](#)
- Scene/Game/SceneGame.cpp, [367](#)
- Scene/Game/SceneGame.hpp, [368](#)
- Scene/IScene.hpp, [369](#)
- Scene/Manager/SceneManager.cpp, [369](#)
- Scene/Manager/SceneManager.hpp, [370](#)
- Scene/Menu/SceneMenu.cpp, [371](#)
- Scene/Menu/SceneMenu.hpp, [371](#)
- Scene/Settings/SceneSettings.cpp, [372](#)
- Scene/Settings/SceneSettings.hpp, [372](#)
- Scene/Skins/SceneSkins.cpp, [373](#)
- Scene/Skins/SceneSkins.hpp, [373](#)
- Scene/Win/SceneWin.cpp, [374](#)
- Scene/Win/SceneWin.hpp, [374](#)
- SceneGame, [217](#)
 - ~SceneGame, [220](#)
 - buttonsHover, [221](#)
 - buttonsPressed, [221](#)
 - camera, [228](#)
 - checkEvents, [222](#)
 - checkName, [222](#)
 - closeEvent, [222](#)
 - createCharacter, [223](#)
 - createObject, [223](#)
 - currentPlayerIndex, [228](#)
 - display, [223](#)
 - displayCharacter, [224](#)
 - displayInventory, [224](#)
 - displayPanel, [224](#)
 - displaySelected, [225](#)
 - displayTileContent, [225](#)
 - getCamera, [225](#)
 - itemDrawer, [228](#)
 - loadScene, [225](#)
 - mapFrame, [228](#)
 - mapMove, [226](#)
 - operator=, [226](#)
 - panelBars, [228](#)
 - panelBarVisible, [228](#)
 - previousSelected, [227](#)
 - removeCharacter, [227](#)
 - SceneGame, [220](#), [221](#)
 - selected, [227](#)
 - showInventory, [229](#)
 - sprite, [229](#)
 - spriteCharacters, [229](#)
 - spritePanel, [229](#)
 - texture, [229](#)
 - textureCharacters, [229](#)
 - texturePanel, [229](#)
- SceneGame.cpp
 - PATH_GOLEM, [367](#)
 - PATH_PANEL, [368](#)
- SceneManager
 - gui::SceneManager, [233](#)
- SceneMenu, [240](#)
 - ~SceneMenu, [243](#)
 - camera, [246](#)
 - checkEvents, [244](#)
 - connexionSprite, [246](#)
 - createCharacter, [244](#)
 - createObject, [244](#)
 - display, [245](#)
 - loadScene, [245](#)
 - logo, [247](#)
 - operator=, [246](#)
 - parallax, [247](#)
 - SceneMenu, [243](#), [244](#)

- texture, [247](#)
- textureLogo, [247](#)
- SceneSettings, [248](#)
 - ~SceneSettings, [250](#)
 - camera, [253](#)
 - checkEvents, [251](#)
 - createCharacter, [251](#)
 - createObject, [251](#)
 - display, [252](#)
 - loadScene, [252](#)
 - operator=, [252](#), [253](#)
 - SceneSettings, [250](#), [251](#)
 - volumeChanged, [253](#)
- SceneSkins, [254](#)
 - ~SceneSkins, [256](#)
 - buffer, [260](#)
 - camera, [260](#)
 - checkEvents, [257](#)
 - createCharacter, [258](#)
 - createObject, [258](#)
 - directionnalButtons, [258](#)
 - display, [258](#)
 - loadScene, [259](#)
 - operator=, [259](#)
 - SceneSkins, [256](#), [257](#)
 - sound, [260](#)
- SceneWin, [261](#)
 - ~SceneWin, [263](#)
 - buffer, [266](#)
 - camera, [266](#)
 - checkEvents, [264](#)
 - createCharacter, [264](#)
 - createObject, [264](#)
 - display, [265](#)
 - font, [266](#)
 - haswonText, [267](#)
 - loadScene, [265](#)
 - operator=, [265](#), [266](#)
 - SceneWin, [263](#), [264](#)
 - sound, [267](#)
 - winText, [267](#)
- segCommand
 - gui::Handler, [148](#)
- selected
 - SceneGame, [227](#)
- sendCommand
 - gui::Network, [184](#)
- serverIP
 - gui::Network, [185](#)
- serverPort
 - gui::Network, [185](#)
- setAlive
 - Eggs, [59](#)
 - ICharacters, [159](#)
 - Players, [210](#)
- setBackground
 - gui::Window, [313](#)
- setBroadcasting
 - Eggs, [59](#)
 - ICharacters, [159](#)
 - Players, [212](#)
- setBuffer
 - SoundPlayer, [278](#)
- setButtonState
 - gui::Button, [29](#)
- setCenter
 - gui::Camera, [35](#)
- setCharacters
 - gui::Data, [45](#)
- setClearColor
 - gui::Window, [313](#)
- setCursor
 - gui::Window, [314](#)
- setCursorPos
 - gui::Window, [314](#)
- setElevating
 - Eggs, [60](#)
 - ICharacters, [160](#)
 - Players, [212](#)
- setFRect
 - gui::Tile, [303](#)
- setFrequency
 - gui::Data, [46](#)
- setInventory
 - Eggs, [60](#)
 - ICharacters, [160](#)
 - Players, [212](#)
- setInventoryText
 - gui::PanelBar, [193](#)
- setIsPlaying
 - SoundPlayer, [279](#)
- setIsSelected
 - gui::Tile, [303](#)
- setLaying
 - Eggs, [60](#)
 - ICharacters, [160](#)
 - Players, [213](#)
- setLevel
 - Eggs, [61](#)
 - ICharacters, [161](#)
 - Players, [213](#)
- setMapFrame
 - gui::Data, [46](#)
- setMapFrameSprite
 - gui::SceneManager, [237](#)
- setMousePos
 - gui::Window, [314](#)
- setName
 - gui::Team, [289](#)
- setPos
 - Eggs, [61](#)
 - gui::Button, [29](#)
 - ICharacters, [161](#)
 - Players, [213](#)
- setPosition
 - gui::Tile, [303](#)

- setRect
 - gui::Tile, 303
- setResources
 - Eggs, 61
 - gui::Tile, 304
 - ICharacters, 161
 - Players, 214
- setRessourcesText
 - gui::PanelBar, 193
- setShape
 - gui::Tile, 304
- setShapeButton
 - gui::Button, 29
- setSize
 - gui::Camera, 35
- setSound
 - SoundPlayer, 279
- setSprite
 - gui::ItemDrawer, 173
- setState
 - gui::SceneManager, 237
- setString
 - TextBox, 295
- setTileSkin
 - gui::Map, 178
- setTileSprite
 - gui::SceneManager, 238
- setTilesRect
 - gui::Data, 46
- setView
 - gui::Camera, 35
- setVolume
 - gui::SceneManager, 238
 - SoundBox, 274
- setWinningTeam
 - gui::Data, 46
- setZoom
 - gui::Camera, 36
- sgtCommand
 - gui::Handler, 148
- shape
 - gui::Button, 30
 - gui::Tile, 305
- shapeButton
 - gui::Button, 30
- shapePos
 - gui::Button, 30
- showInventory
 - SceneGame, 229
- SIBUR
 - Stones.hpp, 365
- smgCommand
 - gui::Handler, 149
- socket
 - gui::Network, 185
- Song, 267
 - name, 268
 - soundPlayer, 269
- sound
 - SceneSkins, 260
 - SceneWin, 267
 - SoundPlayer, 279
- SoundBox, 269
 - ~SoundBox, 270
 - getSoundStatus, 271
 - getVolume, 271
 - initSong, 272
 - operator=, 272
 - play, 274
 - setVolume, 274
 - SoundBox, 270, 271
 - sounds, 275
 - stop, 274
 - volume, 275
- soundbox
 - gui::SceneManager, 239
- soundButton
 - gui::Button, 31
- SoundPlayer, 275
 - ~SoundPlayer, 276
 - buffer, 279
 - getBuffer, 277
 - getIsPlaying, 277
 - getSound, 277
 - init, 278
 - isPlaying, 279
 - operator=, 278
 - setBuffer, 278
 - setIsPlaying, 279
 - setSound, 279
 - sound, 279
 - SoundPlayer, 276, 277
- soundPlayer
 - Song, 269
- soundRadius
 - Eggs, 63
 - Players, 216
- sounds
 - SoundBox, 275
- speed
 - Background, 22
- sprite
 - Background, 22
 - gui::SceneManager, 239
 - gui::Window, 316
 - SceneGame, 229
- spriteCharacters
 - SceneGame, 229
- spriteld
 - infos, 163
- spritePanel
 - SceneGame, 229
- sstCommand
 - gui::Handler, 149
- STONES
 - IObjects.hpp, 362

- Stones, 280
 - ~Stones, 282
 - display, 283
 - getType, 283
 - operator=, 283, 284
 - pos, 284
 - rect, 284
 - Stones, 282, 283
 - type, 284
 - typeObject, 285
- Stones.cpp
 - findRect, 363
 - rects, 363
- Stones.hpp
 - DERAUMERE, 365
 - LINEMATE, 365
 - MENDIANE, 365
 - PHIRAS, 365
 - SIBUR, 365
 - THYSTAME, 365
 - TypeStone, 365
- stop
 - SoundBox, 274
- stopSoundButton
 - gui::Button, 29
- sucCommand
 - gui::Handler, 149
- Team
 - gui::Team, 286, 287
- Team/Team.cpp, 375
- Team/Team.hpp, 375
- teamName
 - Eggs, 63
 - Players, 216
- teams
 - gui::Data, 47
- text
 - TextBox, 297
- TextBox, 290
 - ~TextBox, 291
 - activate, 292
 - box, 296
 - clear, 293
 - descriptionText, 296
 - draw, 293
 - finished, 296
 - getPosition, 293
 - getString, 293
 - initTextBox, 293
 - inputMode, 296
 - inputText, 297
 - isFinished, 294
 - isInputMode, 294
 - isPressed, 294
 - operator=, 295
 - rect, 297
 - setString, 295
 - text, 297
 - TextBox, 291, 292
 - textEvent, 296
- textBox
 - gui::Data, 48
- textEvent
 - TextBox, 296
- texture
 - Background, 23
 - gui::Map, 179
 - gui::SceneManager, 239
 - gui::Window, 316
 - SceneGame, 229
 - SceneMenu, 247
- textureCharacters
 - SceneGame, 229
- textureLogo
 - SceneMenu, 247
- texturePanel
 - SceneGame, 229
- THYSTAME
 - Stones.hpp, 365
- Tile, 306
 - gui::Tile, 299, 300
- Tile.hpp
 - TILE_SIZE, 346
- TILE_SIZE
 - Tile.hpp, 346
- tiles
 - gui::Map, 179
- tileSprite
 - gui::SceneManager, 239
- tilesRect
 - gui::Data, 48
- tileTexture
 - gui::SceneManager, 240
- timeUnit
 - gui::Data, 48
- tnaCommand
 - gui::Handler, 150
- type
 - Eggs, 63
 - Foods, 138
 - Stones, 284
- TypeEggs
 - Eggs.hpp, 321
- TypeObject
 - IOObjects.hpp, 362
- typeObject
 - Stones, 285
- TypeStone
 - Stones.hpp, 365
- update
 - gui::SceneManager, 238
- view
 - gui::Camera, 36
- volSprite
 - gui::SceneManager, 240

- volume
 - SoundBox, [275](#)
- volumeChanged
 - SceneSettings, [253](#)
- what
 - Errors::ErrorBroadcast, [67](#)
 - Errors::ErrorButton, [71](#)
 - Errors::ErrorCharacter, [75](#)
 - Errors::ErrorColor, [79](#)
 - Errors::ErrorElevate, [83](#)
 - Errors::ErrorEnv, [87](#)
 - Errors::ErrorFile, [91](#)
 - Errors::ErrorFont, [95](#)
 - Errors::ErrorInventory, [99](#)
 - Errors::ErrorOpen, [103](#)
 - Errors::ErrorSocket, [111](#)
 - Errors::ErrorTeam, [116](#)
 - Errors::ErrorTexture, [120](#)
 - Errors::ErrorUsage, [124](#)
- WHITE
 - Eggs.hpp, [321](#)
- width
 - gui::Map, [179](#)
- Window
 - gui::Window, [309](#)
- window
 - gui::Window, [316](#)
- winningTeam
 - gui::Data, [48](#)
- winText
 - SceneWin, [267](#)
- YELLOW
 - Eggs.hpp, [321](#)
- yellowRectFunc
 - Orientation.cpp, [324](#)
- zoomFactor
 - gui::Camera, [36](#)